

# ANSYS Mechanical APDL Structural Analysis Guide

---



ANSYS, Inc.  
Southpointe  
275 Technology Drive  
Canonsburg, PA 15317  
ansysinfo@ansys.com  
<http://www.ansys.com>  
(T) 724-746-3304  
(F) 724-514-9494

Release 15.0  
November 2013

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

---

## Copyright and Trademark Information

© 2013 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

<b>1. Overview of Structural Analyses</b> .....	1
1.1. Types of Structural Analysis .....	1
1.2. Elements Used in Structural Analyses .....	2
1.3. Material Model Interface .....	2
1.4. Damping .....	2
1.4.1. Alpha and Beta Damping (Rayleigh Damping) .....	5
1.4.2. Material-Dependent Alpha and Beta Damping (Rayleigh Damping) .....	6
1.4.3. Constant Global Damping Ratio .....	6
1.4.4. Constant Structural Damping Coefficient .....	6
1.4.5. Mode-Dependent Damping Ratio .....	7
1.4.6. Viscoelastic Material Damping .....	7
1.4.7. Element Damping .....	7
1.5. Solution Method .....	7
<b>2. Structural Static Analysis</b> .....	9
2.1. Linear vs. Nonlinear Static Analyses .....	9
2.2. Performing a Static Analysis .....	9
2.2.1. Build the Model .....	10
2.2.1.1. Points to Remember .....	10
2.2.2. Set Solution Controls .....	10
2.2.2.1. Access the Solution Controls Dialog Box .....	11
2.2.2.2. Using the Basic Tab .....	11
2.2.2.3. The Transient Tab .....	12
2.2.2.4. Using the Sol'n Options Tab .....	12
2.2.2.5. Using the Nonlinear Tab .....	12
2.2.2.6. Using the Advanced NL Tab .....	13
2.2.3. Set Additional Solution Options .....	13
2.2.3.1. Stress Stiffening Effects .....	13
2.2.3.2. Newton-Raphson Option .....	14
2.2.3.3. Prestress Effects Calculation .....	14
2.2.3.4. Mass Matrix Formulation .....	14
2.2.3.5. Reference Temperature .....	14
2.2.3.6. Mode Number .....	15
2.2.3.7. Creep Criteria .....	15
2.2.3.8. Printed Output .....	15
2.2.3.9. Extrapolation of Results .....	15
2.2.4. Apply the Loads .....	15
2.2.4.1. Load Types .....	15
2.2.4.1.1. Displacements (UX, UY, UZ, ROTX, ROTY, ROTZ) .....	15
2.2.4.1.2. Velocities (VELX, VELY, VELZ, OMGX, OMGY, OMGZ) .....	15
2.2.4.1.3. Forces (FX, FY, FZ) and Moments (MX, MY, MZ) .....	16
2.2.4.1.4. Pressures (PRES) .....	16
2.2.4.1.5. Temperatures (TEMP) .....	16
2.2.4.1.6. Fluences (FLUE) .....	16
2.2.4.1.7. Gravity, Spinning, Etc. .....	16
2.2.4.2. Apply Loads to the Model .....	16
2.2.4.2.1. Applying Loads Using TABLE Type Array Parameters .....	17
2.2.4.3. Calculating Inertia Relief .....	17
2.2.4.3.1. Inertia Relief Output .....	18
2.2.4.3.2. Using a Macro to Perform Inertia Relief Calculations .....	18
2.2.5. Solve the Analysis .....	18

2.2.6. Review the Results .....	19
2.2.6.1. Postprocessors .....	19
2.2.6.2. Points to Remember .....	19
2.2.6.3. Reviewing Results Data .....	19
2.2.6.4. Typical Postprocessing Operations .....	20
2.3. A Sample Static Analysis (GUI Method) .....	22
2.3.1. Problem Description .....	22
2.3.2. Problem Specifications .....	22
2.3.3. Problem Sketch .....	22
2.3.3.1. Set the Analysis Title .....	23
2.3.3.2. Set the System of Units .....	23
2.3.3.3. Define Parameters .....	23
2.3.3.4. Define the Element Types .....	24
2.3.3.5. Define Material Properties .....	24
2.3.3.6. Create Hexagonal Area as Cross-Section .....	24
2.3.3.7. Create Keypoints Along a Path .....	25
2.3.3.8. Create Lines Along a Path .....	25
2.3.3.9. Create Line from Shank to Handle .....	26
2.3.3.10. Cut Hex Section .....	26
2.3.3.11. Set Meshing Density .....	27
2.3.3.12. Set Element Type for Area Mesh .....	27
2.3.3.13. Generate Area Mesh .....	27
2.3.3.14. Drag the 2-D Mesh to Produce 3-D Elements .....	27
2.3.3.15. Select BOTAREA Component and Delete 2-D Elements .....	28
2.3.3.16. Apply Displacement Boundary Condition at End of Wrench .....	28
2.3.3.17. Display Boundary Conditions .....	29
2.3.3.18. Apply Pressure on Handle .....	29
2.3.3.19. Write the First Load Step .....	31
2.3.3.20. Define Downward Pressure .....	31
2.3.3.21. Write Second Load Step .....	32
2.3.3.22. Solve from Load Step Files .....	32
2.3.3.23. Read First Load Step and Review Results .....	32
2.3.3.24. Read the Next Load Step and Review Results .....	33
2.3.3.25. Zoom in on Cross-Section .....	34
2.3.3.26. Exit ANSYS .....	34
2.4. A Sample Static Analysis (Command or Batch Method) .....	34
2.5. Where to Find Other Examples .....	37
<b>3. Modal Analysis .....</b>	<b>39</b>
3.1. Uses for Modal Analysis .....	39
3.2. Understanding the Modal Analysis Process .....	39
3.3. Building the Model for a Modal Analysis .....	40
3.4. Applying Loads and Obtaining the Solution .....	40
3.4.1. Enter the Solution Processor .....	40
3.4.2. Define Analysis Type and Options .....	40
3.4.2.1. Option: New Analysis ( <b>ANTYPE</b> ) .....	41
3.4.2.2. Option: Analysis Type: Modal ( <b>ANTYPE</b> ) .....	41
3.4.2.3. Option: Mode-Extraction Method ( <b>MODOPT</b> ) .....	41
3.4.2.4. Option: Number of Modes to Extract ( <b>MODOPT</b> ) .....	43
3.4.2.5. Option: Number of Modes to Expand ( <b>MXPAND</b> ) .....	43
3.4.2.6. Option: Results File Output ( <b>OUTRES</b> ) .....	44
3.4.2.7. Option: Mass Matrix Formulation ( <b>LUMPM</b> ) .....	44
3.4.2.8. Option: Prestress Effects Calculation ( <b>PSTRES</b> ) .....	44

3.4.2.9. Option: Residual Vector Calculation ( <b>RESVEC</b> ) .....	44
3.4.2.10. Additional Modal Analysis Options .....	44
3.4.3. Apply Loads .....	44
3.4.3.1. Applying Loads Using Commands .....	45
3.4.3.2. Applying Loads Using the GUI .....	45
3.4.3.3. Listing Loads .....	46
3.4.4. Specify Load Step Options .....	46
3.4.5. Solve .....	46
3.4.5.1. Output .....	47
3.4.6. Participation Factor Table Output .....	47
3.4.7. Exit the Solution Processor .....	47
3.5. Reviewing the Results .....	47
3.5.1. Points to Remember .....	47
3.5.2. Reviewing Results Data .....	48
3.5.3. Option: Listing All Frequencies .....	48
3.5.4. Option: Display Deformed Shape .....	48
3.5.5. Option: Line Element Results .....	48
3.5.6. Option: Contour Displays .....	48
3.5.7. Option: Tabular Listings .....	49
3.5.8. Other Capabilities .....	49
3.6. Applying Prestress Effects in a Modal Analysis .....	49
3.6.1. Performing a Prestressed Modal Analysis from a Linear Base Analysis .....	49
3.6.2. Performing a Prestressed Modal Analysis from a Large-Deflection Base Analysis .....	50
3.7. Modal Analysis Examples .....	50
3.7.1. An Example Modal Analysis (GUI Method) .....	51
3.7.1.1. Problem Description .....	51
3.7.1.2. Problem Specifications .....	51
3.7.1.3. Problem Sketch .....	51
3.7.2. An Example Modal Analysis (Command or Batch Method) .....	51
3.7.3. Brake Squeal (Prestressed Modal) Analysis .....	53
3.7.3.1. Full Nonlinear Perturbed Modal Analysis .....	54
3.7.3.2. Partial Nonlinear Perturbed Modal Analysis .....	55
3.7.3.3. Linear Non-prestressed Modal Analysis .....	57
3.7.4. Reuse of <code>Jobname.MODESYM</code> in QRDAMP eigensolver .....	58
3.7.5. Calculate the Complex Mode Contribution Coefficients (CMCC) .....	59
3.7.6. Where to Find Other Modal Analysis Examples .....	60
3.8. Comparing Mode-Extraction Methods .....	61
3.8.1. Block Lanczos Method .....	62
3.8.2. PCG Lanczos Method .....	62
3.8.3. Supernode (SNODE) Method .....	63
3.8.4. Subspace Method .....	64
3.8.5. Unsymmetric Method .....	64
3.8.6. Damped Method .....	64
3.8.6.1. Damped Method--Real and Imaginary Parts of the Eigenvalue .....	64
3.8.6.2. Damped Method-Real and Imaginary Parts of the Eigenvector .....	65
3.8.7. QR Damped Method .....	65
3.8.8. Storage of Complex Results .....	65
3.9. Modal Analysis Tools for Subsequent Mode Superposition Analysis .....	66
3.9.1. Using the Residual-Vector Method to Improve Accuracy .....	66
3.9.1.1. Understanding the Residual Vector Method .....	66
3.9.1.2. Using the Residual Vector Method .....	67
3.9.2. Reusing Eigenmodes .....	68

3.9.2.1. Spectrum Analysis ( <b>ANTYPE,SPECTRUM</b> ) .....	68
3.9.2.2. Modal Transient Analysis/Harmonic Analysis .....	68
3.9.2.3. QR Damp Complex Modes Extraction .....	68
3.9.3. Generating and Using Multiple Loads in Mode-Superposition Analyses .....	68
3.9.3.1. Understanding the Multiple Loads Method .....	68
3.9.3.2. Using the Multiple Loads Method .....	69
3.9.4. Restarting a Modal Analysis .....	69
3.9.5. Enforced Motion Method for Mode-Superposition Transient and Harmonic Analyses .....	70
3.9.5.1. Understanding the Enforced Motion Method .....	70
3.9.5.2. Using the Enforced Motion Method .....	70
3.9.5.3. Sample Input for Enforced Motion Mode-Superposition Analysis .....	70
3.9.6. Using Mode Selection .....	71
3.9.6.1. Mode Selection Based on a User Defined Array .....	72
3.9.6.2. Mode Selection Based on Modal Effective Mass .....	73
3.9.6.3. Mode Selection Based on the Mode Coefficients .....	74
3.9.6.4. Mode Selection Based on DDAM Procedure .....	75
<b>4. Harmonic Analysis</b> .....	<b>77</b>
4.1. Uses for Harmonic Analysis .....	77
4.2. Commands Used in a Harmonic Analysis .....	78
4.3. Two Solution Methods .....	78
4.3.1. The Full Method .....	78
4.3.2. The Mode-Superposition Method .....	79
4.3.3. Restrictions Common to Both Methods .....	79
4.4. Performing a Harmonic Analysis .....	79
4.4.1. Full Harmonic Analysis .....	79
4.4.2. Build the Model .....	79
4.4.2.1. Modeling Hints .....	80
4.4.3. Apply Loads and Obtain the Solution .....	80
4.4.3.1. Enter the ANSYS Solution Processor .....	80
4.4.3.2. Define the Analysis Type and Options .....	80
4.4.3.3. Apply Loads on the Model .....	82
4.4.3.3.1. Applying Loads Using Commands .....	84
4.4.3.3.2. Applying Loads and Listing Loads Using the GUI .....	85
4.4.3.4. Specify Load Step Options .....	85
4.4.3.4.1. General Options .....	86
4.4.3.4.2. Dynamics Options .....	86
4.4.3.4.3. Output Controls .....	87
4.4.3.5. Save a Backup Copy of the Database to a Named File .....	87
4.4.3.6. Start Solution Calculations .....	87
4.4.3.7. Repeat for Additional Load Steps .....	88
4.4.3.8. Leave SOLUTION .....	88
4.4.4. Review the Results .....	88
4.4.4.1. Postprocessors .....	88
4.4.4.2. Points to Remember .....	88
4.4.4.3. Using POST26 .....	88
4.4.4.4. Using POST1 .....	89
4.5. Sample Harmonic Analysis (GUI Method) .....	90
4.5.1. Problem Description .....	90
4.5.2. Problem Specifications .....	90
4.5.3. Problem Diagram .....	91
4.5.3.1. Set the Analysis Title .....	91
4.5.3.2. Define the Element Types .....	91

4.5.3.3. Define the Real Constants .....	91
4.5.3.4. Create the Nodes .....	92
4.5.3.5. Create the Spring Elements .....	92
4.5.3.6. Create the Mass Elements .....	93
4.5.3.7. Specify the Analysis Type, MDOF, and Load Step Specifications .....	93
4.5.3.8. Define Loads and Boundary Conditions .....	94
4.5.3.9. Solve the Model .....	94
4.5.3.10. Review the Results .....	94
4.5.3.11. Exit ANSYS .....	96
4.6. Example Harmonic Analysis (Command or Batch Method) .....	96
4.7. Where to Find Other Examples .....	96
4.8. Mode-Superposition Harmonic Analysis .....	97
4.8.1. Obtain the Modal Solution .....	97
4.8.2. Obtain the Mode-Superposition Harmonic Solution .....	98
4.8.3. Expand the Mode-Superposition Solution .....	100
4.8.3.1. Points to Remember .....	101
4.8.3.2. Expanding the Modes .....	101
4.8.4. Review the Results of the Expanded Solution .....	103
4.8.5. Sample Input .....	103
4.9. Additional Harmonic Analysis Details .....	104
4.9.1. Prestressed Harmonic Analysis .....	104
4.9.1.1. Prestressed Full Harmonic Analysis .....	104
4.9.1.2. Prestressed Mode-Superposition Harmonic Analysis .....	105
<b>5. Transient Dynamic Analysis .....</b>	<b>107</b>
5.1. Preparing for a Transient Dynamic Analysis .....	107
5.2. Two Solution Methods .....	108
5.2.1. Full Method .....	108
5.2.2. Mode-Superposition Method .....	109
5.3. Performing a Full Transient Dynamic Analysis .....	109
5.3.1. Build the Model .....	110
5.3.1.1. Points to Remember .....	110
5.3.2. Establish Initial Conditions .....	110
5.3.3. Set Solution Controls .....	113
5.3.3.1. Access the Solution Controls Dialog Box .....	113
5.3.3.2. Using the Basic Tab .....	113
5.3.3.3. Using the Transient Tab .....	114
5.3.3.4. Using the Remaining Solution Controls Tabs .....	115
5.3.3.4.1. Set Additional Solution Options .....	115
5.3.3.4.1.1. Prestress Effects .....	116
5.3.3.4.1.2. Damping Option .....	116
5.3.3.4.1.3. Mass Matrix Formulation .....	116
5.3.4. Apply the Loads .....	117
5.3.5. Save the Load Configuration for the Current Load Step .....	117
5.3.6. Repeat Steps 3-6 for Each Load Step .....	117
5.3.7. Save a Backup Copy of the Database .....	118
5.3.8. Start the Transient Solution .....	118
5.3.9. Exit the Solution Processor .....	118
5.3.10. Review the Results .....	118
5.3.10.1. Postprocessors .....	118
5.3.10.2. Points to Remember .....	118
5.3.10.3. Using POST26 .....	118
5.3.10.4. Other Capabilities .....	119

5.3.10.5. Using POST1 .....	119
5.3.11. Sample Input for a Full Transient Dynamic Analysis .....	119
5.4. Performing a Mode-Superposition Transient Dynamic Analysis .....	120
5.4.1. Build the Model .....	121
5.4.2. Obtain the Modal Solution .....	121
5.4.3. Obtain the Mode-Superposition Transient Solution .....	122
5.4.3.1. Obtaining the Solution .....	122
5.4.4. Expand the Mode-Superposition Solution .....	126
5.4.4.1. Points to Remember .....	126
5.4.4.2. Expanding the Solution .....	126
5.4.4.3. Reviewing the Results of the Expanded Solution .....	128
5.4.5. Review the Results .....	128
5.4.6. Sample Input for a Mode-Superposition Transient Dynamic Analysis .....	128
5.5. Performing a Prestressed Transient Dynamic Analysis .....	129
5.5.1. Prestressed Full Transient Dynamic Analysis .....	129
5.5.2. Prestressed Mode-Superposition Transient Dynamic Analysis .....	130
5.6. Transient Dynamic Analysis Options .....	130
5.6.1. Guidelines for Integration Time Step .....	130
5.6.2. Automatic Time Stepping .....	132
5.7. Where to Find Other Examples .....	133
<b>6. Spectrum Analysis .....</b>	<b>135</b>
6.1. Understanding Spectrum Analysis .....	135
6.1.1. Response Spectrum .....	135
6.1.1.1. Single-Point Response Spectrum (SPRS) .....	136
6.1.1.2. Multi-Point Response Spectrum (MPRS) .....	136
6.1.2. Dynamic Design Analysis Method (DDAM) .....	136
6.1.3. Power Spectral Density .....	136
6.1.4. Deterministic vs. Probabilistic Analyses .....	137
6.2. Single-Point Response Spectrum (SPRS) Analysis Process .....	137
6.2.1. Step 1: Build the Model .....	137
6.2.1.1. Hints and Recommendations .....	137
6.2.2. Step 2: Obtain the Modal Solution .....	137
6.2.3. Step 3: Obtain the Spectrum Solution .....	138
6.2.4. Step 4: Review the Results .....	143
6.2.5. Running Multiple Spectrum Analyses .....	145
6.3. Example Spectrum Analysis (GUI Method) .....	146
6.3.1. Problem Description .....	146
6.3.2. Problem Specifications .....	146
6.3.3. Problem Sketch .....	147
6.3.4. Procedure .....	147
6.3.4.1. Set the Analysis Title .....	147
6.3.4.2. Define the Element Type .....	147
6.3.4.3. Define the Cross-Section Area .....	148
6.3.4.4. Define Material Properties .....	148
6.3.4.5. Define Keypoints and Line .....	148
6.3.4.6. Set Global Element Density and Mesh Line .....	149
6.3.4.7. Set Boundary Conditions .....	149
6.3.4.8. Specify Analysis Type and Options .....	150
6.3.4.9. Solve the Modal Analysis .....	150
6.3.4.10. Set Up the Spectrum Analysis .....	150
6.3.4.11. Define Spectrum Value vs. Frequency Table .....	150
6.3.4.12. Select Mode Combination Method .....	151



6.3.4.13. Solve Spectrum Analysis .....	151
6.3.4.14. Postprocessing: Print Out Nodal, Element, and Reaction Solutions .....	151
6.3.4.15. Exit Mechanical APDL .....	151
6.4. Example Spectrum Analysis (Command or Batch Method) .....	152
6.4.1. Single Point Response Spectrum Analysis on a Beam Structure .....	152
6.4.2. Single Point Response Spectrum Analysis on a Piping Structure with Excitation along X, Y, and Z Directions .....	153
6.4.3. Single Point Response Spectrum Analysis on a Piping Structure with Excitation along X, Y, and Z Directions Separately by Reusing the Existing Mode File .....	155
6.5. Where to Find Other Examples .....	158
6.6. Performing a Random Vibration (PSD) Analysis .....	159
6.6.1. Obtain the PSD Solution .....	159
6.6.2. Combine the Modes .....	163
6.6.3. Review the Results .....	164
6.6.3.1. Reviewing the Results in POST1 .....	165
6.6.3.1.1. Read the Desired Set of Results into the Database .....	165
6.6.3.1.2. Display the Results .....	166
6.6.3.2. Calculating Response PSDs in POST26 .....	166
6.6.3.3. Calculating Covariance in POST26 .....	167
6.6.4. Sample Input .....	167
6.7. DDAM Spectrum Analysis Process .....	168
6.7.1. Step 3: Obtain the Spectrum Solution .....	169
6.7.2. Step 4: Review the Results .....	172
6.7.3. Sample Input .....	172
6.8. Performing a Multi-Point Response Spectrum (MPRS) Analysis .....	173
6.8.1. Step 4: Obtain the Spectrum Solution .....	173
6.8.2. Step 5: Combine the Modes .....	176
6.8.3. Step 6: Review the Results .....	176
6.9. Example Multi-Point Response Spectrum (MPRS) Analysis (Command or Batch Method) .....	177
6.9.1. Problem Description .....	177
6.9.2. Problem Specifications .....	177
6.9.3. Problem Sketch .....	177
6.9.4. Command Listing .....	177
<b>7. Buckling Analysis .....</b>	<b>179</b>
7.1. Types of Buckling Analyses .....	179
7.1.1. Nonlinear Buckling Analysis .....	179
7.1.2. Eigenvalue Buckling Analysis .....	179
7.2. Commands Used in a Buckling Analysis .....	180
7.3. Performing a Nonlinear Buckling Analysis .....	180
7.3.1. Applying Load Increments .....	180
7.3.2. Automatic Time Stepping .....	180
7.3.3. Unconverged Solution .....	181
7.3.4. Hints and Tips for Performing a Nonlinear Buckling Analysis .....	181
7.4. Performing a Post-Buckling Analysis .....	181
7.5. Procedure for Eigenvalue Buckling Analysis .....	182
7.5.1. Build the Model .....	182
7.5.1.1. Points to Remember .....	182
7.5.2. Obtain the Static Solution .....	182
7.5.3. Obtain the Eigenvalue Buckling Solution .....	183
7.5.4. Review the Results .....	186
7.6. Sample Buckling Analysis (GUI Method) .....	186
7.6.1. Problem Description .....	186

7.6.2. Problem Specifications .....	186
7.6.3. Problem Sketch .....	187
7.6.3.1. Set the Analysis Title .....	187
7.6.3.2. Define the Element Type .....	187
7.6.3.3. Define the Real Constants and Material Properties .....	188
7.6.3.4. Define Nodes and Elements .....	188
7.6.3.5. Define the Boundary Conditions .....	189
7.6.3.6. Solve the Static Analysis .....	190
7.6.3.7. Solve the Buckling Analysis .....	190
7.6.3.8. Review the Results .....	191
7.6.3.9. Exit ANSYS .....	191
7.7. Sample Buckling Analysis (Command or Batch Method) .....	191
7.8. Where to Find Other Examples .....	192
<b>8. Nonlinear Structural Analysis .....</b>	<b>193</b>
8.1. Causes of Nonlinear Behavior .....	194
8.1.1. Changing Status (Including Contact) .....	194
8.1.2. Geometric Nonlinearities .....	194
8.1.3. Material Nonlinearities .....	195
8.2. Understanding Nonlinear Analyses .....	195
8.2.1. Conservative vs. Nonconservative Behavior; Path Dependency .....	197
8.2.2. Substeps .....	198
8.2.3. Load Direction in a Large-Deflection Analysis .....	198
8.2.4. Rotations in a Large-Deflection Analysis .....	199
8.2.5. Nonlinear Transient Analyses .....	200
8.3. Using Geometric Nonlinearities .....	200
8.3.1. Stress-Strain .....	200
8.3.1.1. Large Deflections with Small Strain .....	200
8.3.2. Stress Stiffening .....	200
8.4. Modeling Material Nonlinearities .....	201
8.4.1. Nonlinear Materials .....	201
8.4.1.1. Plasticity .....	202
8.4.1.2. Hyperelasticity Material Model .....	202
8.4.1.2.1. Mooney-Rivlin Hyperelastic Option (TB,HYPER,,,,MOONEY) .....	203
8.4.1.2.2. Ogden Hyperelastic Option (TB,HYPER,,,,OGDEN) .....	203
8.4.1.2.3. Neo-Hookean Hyperelastic Option (TB,HYPER,,,,NEO) .....	204
8.4.1.2.4. Polynomial Form Hyperelastic Option (TB,HYPER,,,,POLY) .....	204
8.4.1.2.5. Arruda-Boyce Hyperelastic Option (TB,HYPER,,,,BOYCE) .....	204
8.4.1.2.6. Gent Hyperelastic Option (TB,HYPER,,,,GENT) .....	205
8.4.1.2.7. Yeoh Hyperelastic Option (TB,HYPER,,,,YEOH) .....	205
8.4.1.2.8. Blatz-Ko Foam Hyperelastic Option (TB,HYPER,,,,BLATZ) .....	205
8.4.1.2.9. Ogden Compressible Foam Hyperelastic Option (TB,HYPER,,,,FOAM) .....	205
8.4.1.2.10. Response Function Hyperelastic Option (TB,HYPER,,,,RESPONSE) .....	206
8.4.1.2.11. User-Defined Hyperelastic Option (TB,HYPER,,,,USER) .....	208
8.4.1.3. Bergstrom-Boyce Hyperviscoelastic Material Model .....	208
8.4.1.4. Mullins Effect Material Model .....	209
8.4.1.5. Anisotropic Hyperelasticity Material Model .....	210
8.4.1.6. Creep Material Model .....	210
8.4.1.6.1. Implicit Creep Procedure .....	211
8.4.1.6.2. Explicit Creep Procedure .....	213
8.4.1.7. Shape Memory Alloy Material Model .....	213
8.4.1.8. Viscoplasticity .....	213
8.4.1.9. Viscoelasticity .....	215

8.4.1.10. Swelling .....	216
8.4.1.11. User-Defined Material Model .....	216
8.4.2. Material Model Combination Examples .....	216
8.4.2.1. RATE and CHAB and BISO Example .....	217
8.4.2.2. RATE and CHAB and MISO Example .....	218
8.4.2.3. RATE and CHAB and PLAS (Multilinear Isotropic Hardening) Example .....	218
8.4.2.4. RATE and CHAB and NLISO Example .....	219
8.4.2.5. BISO and CHAB Example .....	219
8.4.2.6. MISO and CHAB Example .....	219
8.4.2.7. PLAS (Multilinear Isotropic Hardening) and CHAB Example .....	220
8.4.2.8. NLISO and CHAB Example .....	220
8.4.2.9. PLAS (Multilinear Isotropic Hardening) and EDP Example .....	220
8.4.2.10. MISO and EDP Example .....	221
8.4.2.11. GURSON and BISO Example .....	221
8.4.2.12. GURSON and MISO Example .....	222
8.4.2.13. GURSON and PLAS (MISO) Example .....	222
8.4.2.14. NLISO and GURSON Example .....	223
8.4.2.15. RATE and BISO Example .....	223
8.4.2.16. MISO and RATE Example .....	224
8.4.2.17. RATE and PLAS (Multilinear Isotropic Hardening) Example .....	224
8.4.2.18. RATE and NLISO Example .....	224
8.4.2.19. BISO and CREEP Example .....	225
8.4.2.20. MISO and CREEP Example .....	225
8.4.2.21. PLAS (Multilinear Isotropic Hardening) and CREEP Example .....	225
8.4.2.22. NLISO and CREEP Example .....	226
8.4.2.23. BKIN and CREEP Example .....	226
8.4.2.24. HILL and BISO Example .....	226
8.4.2.25. HILL and MISO Example .....	227
8.4.2.26. HILL and PLAS (Multilinear Isotropic Hardening) Example .....	227
8.4.2.27. HILL and NLISO Example .....	228
8.4.2.28. HILL and BKIN Example .....	228
8.4.2.29. HILL and MKIN Example .....	228
8.4.2.30. HILL and KINH Example .....	229
8.4.2.31. HILL, and PLAS (Kinematic Hardening) Example .....	229
8.4.2.32. HILL and CHAB Example .....	230
8.4.2.33. HILL and BISO and CHAB Example .....	230
8.4.2.34. HILL and MISO and CHAB Example .....	230
8.4.2.35. HILL and PLAS (Multilinear Isotropic Hardening) and CHAB Example .....	231
8.4.2.36. HILL and NLISO and CHAB Example .....	231
8.4.2.37. HILL and RATE and BISO Example .....	232
8.4.2.38. HILL and RATE and MISO Example .....	233
8.4.2.39. HILL and RATE and NLISO Example .....	233
8.4.2.40. HILL and CREEP Example .....	234
8.4.2.41. HILL, CREEP and BISO Example .....	234
8.4.2.42. HILL and CREEP and MISO Example .....	235
8.4.2.43. HILL, CREEP and PLAS (Multilinear Isotropic Hardening) Example .....	236
8.4.2.44. HILL and CREEP and NLISO Example .....	236
8.4.2.45. HILL and CREEP and BKIN Example .....	237
8.4.2.46. HYPER and VISCO (Hyperelasticity and Viscoelasticity (Implicit)) Example .....	237
8.4.2.47. AHYPER and PRONY (Anisotropic Hyperelasticity and Viscoelasticity (Implicit)) Example .....	237
8.4.2.48. EDP and CREEP and PLAS (MISO) Example .....	238

8.4.2.49. CAP and CREEP and PLAS (MISO) Example .....	239
8.4.2.50. CHAB and CREEP Example .....	240
8.4.2.51. CHAB and CREEP and NLISO Example .....	240
8.4.2.52. CHAB and CREEP and HILL Example .....	240
8.4.2.53. CHAB and CREEP and HILL and MISO Example .....	241
8.5. Running a Nonlinear Analysis .....	242
8.6. Performing a Nonlinear Static Analysis .....	242
8.6.1. Build the Model .....	242
8.6.2. Set Solution Controls .....	243
8.6.2.1. Using the Basic Tab: Special Considerations .....	243
8.6.2.2. Advanced Analysis Options You Can Set on the Solution Controls Dialog Box .....	244
8.6.2.2.1. Equation Solver .....	244
8.6.2.3. Advanced Load Step Options You Can Set on the Solution Controls Dialog Box .....	244
8.6.2.3.1. Automatic Time Stepping .....	244
8.6.2.3.2. Convergence Criteria .....	245
8.6.2.3.3. Maximum Number of Equilibrium Iterations .....	246
8.6.2.3.4. Predictor-Corrector Option .....	246
8.6.2.3.5. VT Accelerator .....	246
8.6.2.3.6. Line Search Option .....	247
8.6.2.3.7. Cutback Criteria .....	247
8.6.3. Set Additional Solution Options .....	247
8.6.3.1. Advanced Analysis Options You Cannot Set via the Solution Controls Dialog Box .....	247
8.6.3.1.1. Stress Stiffness .....	247
8.6.3.1.2. Newton-Raphson Option .....	247
8.6.3.2. Advanced Load Step Options .....	249
8.6.3.2.1. Creep Criteria .....	249
8.6.3.2.2. Time Step Open Control .....	249
8.6.3.2.3. Solution Monitoring .....	250
8.6.3.2.4. Birth and Death .....	250
8.6.3.2.5. Output Control .....	251
8.6.4. Apply the Loads .....	251
8.6.5. Solve the Analysis .....	251
8.6.6. Review the Results .....	251
8.6.6.1. Points to Remember .....	251
8.6.6.2. Reviewing Results in POST1 .....	252
8.6.6.3. Reviewing Results in POST26 .....	253
8.6.7. Terminating a Running Job; Restarting .....	254
8.7. Performing a Nonlinear Transient Analysis .....	254
8.7.1. Build the Model .....	254
8.7.2. Apply Loads and Obtain the Solution .....	254
8.7.3. Review the Results .....	256
8.8. Example Input for a Nonlinear Transient Analysis .....	256
8.9. Restarts .....	257
8.10. Using Nonlinear (Changing-Status) Elements .....	257
8.10.1. Element Birth and Death .....	258
8.11. Unstable Structures .....	258
8.11.1. Using Nonlinear Stabilization .....	258
8.11.1.1. Input for Stabilization .....	259
8.11.1.1.1. Controlling the Stabilization Force .....	259
8.11.1.1.2. Applying a Constant or Reduced Stabilization Force .....	260
8.11.1.1.3. Using the Options for the First Substep .....	261
8.11.1.1.4. Setting the Limit Coefficient for Checking Stabilization Forces .....	261

8.11.1.2. Checking Results After Applying Stabilization .....	262
8.11.1.3. Tips for Using Stabilization .....	262
8.11.2. Using the Arc-Length Method .....	263
8.11.2.1. Checking Arc-Length Results .....	264
8.11.3. Nonlinear Stabilization vs. the Arc-Length Method .....	264
8.12. Guidelines for Nonlinear Analysis .....	265
8.12.1. Setting Up a Nonlinear Analysis .....	265
8.12.1.1. Understand Your Program and Structure Behavior .....	266
8.12.1.2. Simplify Your Model .....	266
8.12.1.3. Use an Adequate Mesh Density .....	266
8.12.1.4. Apply Loading Gradually .....	267
8.12.2. Overcoming Convergence Problems .....	267
8.12.2.1. Overview of Convergence Problems .....	267
8.12.2.2. Performing Nonlinear Diagnostics .....	268
8.12.2.3. Tracking Convergence Graphically .....	269
8.12.2.4. Automatic Time Stepping .....	270
8.12.2.5. Line Search .....	270
8.12.2.6. Nonlinear Stabilization .....	270
8.12.2.7. Arc-Length Method .....	270
8.12.2.8. Artificially Inhibit Divergence in Your Model's Response .....	270
8.12.2.9. Use the Rezoning Feature .....	271
8.12.2.10. Dispense with Extra Element Shapes .....	271
8.12.2.11. Using Element Birth and Death Wisely .....	271
8.12.2.12. Read Your Output .....	271
8.12.2.13. Graph the Load and Response History .....	273
8.13. Example Nonlinear Analysis (GUI Method) .....	273
8.13.1. Problem Description .....	273
8.13.2. Problem Specifications .....	273
8.13.3. Problem Sketch .....	274
8.13.3.1. Set the Analysis Title and Jobname .....	275
8.13.3.2. Define the Element Types .....	275
8.13.3.3. Define Material Properties .....	275
8.13.3.4. Specify the Kinematic Hardening material model (KINH) .....	275
8.13.3.5. Label Graph Axes and Plot Data Tables .....	276
8.13.3.6. Create Rectangle .....	276
8.13.3.7. Set Element Size .....	276
8.13.3.8. Mesh the Rectangle .....	277
8.13.3.9. Assign Analysis and Load Step Options .....	277
8.13.3.10. Monitor the Displacement .....	277
8.13.3.11. Apply Constraints .....	278
8.13.3.12. Solve the First Load Step .....	278
8.13.3.13. Solve the Next Six Load Steps .....	279
8.13.3.14. Review the Monitor File .....	279
8.13.3.15. Use the General Postprocessor to Plot Results. ....	279
8.13.3.16. Define Variables for Time-History Postprocessing .....	280
8.13.3.17. Plot Time-History Results .....	281
8.13.3.18. Exit .....	281
8.14. Example Nonlinear Analysis (Command or Batch Method) .....	281
8.15. Where to Find Other Examples .....	284
<b>9. Linear Perturbation Analysis .....</b>	<b>287</b>
9.1. Understanding Linear Perturbation .....	287
9.2. General Procedure for Linear Perturbation Analysis .....	288

9.2.1. Process Flow for a Linear Perturbation Analysis .....	288
9.2.2. The Base (Prior) Analysis .....	291
9.2.3. First Phase of the Linear Perturbation Analysis .....	292
9.2.4. Second Phase of the Linear Perturbation Analysis .....	293
9.2.4.1. Second Phase - Static Analysis .....	293
9.2.4.2. Second Phase - Modal Analysis .....	294
9.2.4.3. Second Phase - Eigenvalue Buckling Analysis .....	295
9.2.4.4. Second Phase - Harmonic Analysis .....	296
9.2.5. Stress Calculations in a Linear Perturbation Analysis .....	296
9.2.6. Reviewing Results of a Linear Perturbation Analysis .....	297
9.2.7. Downstream Analysis Following the Linear Perturbation Analysis .....	297
9.3. Considerations for Load Generation and Controls .....	298
9.3.1. Generating and Controlling Mechanical Loads .....	299
9.3.2. Generating and Controlling Non-mechanical Loads .....	299
9.4. Considerations for Perturbed Stiffness Matrix Generation .....	300
9.5. Considerations for Rotating Structures .....	301
9.6. Example Inputs for Linear Perturbation Analysis .....	302
9.7. Where to Find Other Examples .....	315
<b>10. Gasket Joints Simulation .....</b>	<b>317</b>
10.1. Performing a Gasket Joint Analysis .....	317
10.2. Finite Element Formulation .....	318
10.2.1. Element Topologies .....	318
10.2.2. Thickness Direction .....	319
10.3. Interface Elements .....	319
10.3.1. Element Selection .....	319
10.3.2. Applications .....	320
10.4. Material Definition .....	320
10.4.1. Material Characteristics .....	320
10.4.2. Input Format .....	321
10.4.2.1. Define General Parameters .....	322
10.4.2.2. Define Compression Load Closure Curve .....	322
10.4.2.3. Define Linear Unloading Data .....	322
10.4.2.4. Define Nonlinear Unloading Data .....	323
10.4.3. Temperature Dependencies .....	324
10.4.4. Plotting Gasket Data .....	327
10.5. Meshing Interface Elements .....	327
10.6. Solution Procedure and Result Output .....	331
10.6.1. Typical Gasket Solution Output Listing .....	332
10.7. Reviewing the Results .....	333
10.7.1. Points to Remember .....	334
10.7.2. Reviewing Results in POST1 .....	334
10.7.3. Reviewing Results in POST26 .....	335
10.8. Sample Gasket Element Verification Analysis (Command or Batch Method) .....	335
<b>11. Fracture Mechanics .....</b>	<b>339</b>
11.1. Introduction to Fracture .....	339
11.1.1. Fracture Modes .....	339
11.1.2. Fracture Mechanics Parameter Calculation .....	340
11.1.2.1. J-Integral .....	340
11.1.2.1.1. J-Integral as a Stress-Intensity Factor .....	340
11.1.2.1.2. Energy-Release Rate .....	341
11.1.2.1.3. Stress-Intensity Factor .....	342
11.1.2.1.4. T-Stress .....	343

11.1.2.5. Material Force .....	343
11.1.3. Crack Growth Simulation .....	346
11.1.3.1. VCCT-Based Interface Element Method .....	346
11.1.3.2. Cohesive Zone Method .....	347
11.1.3.3. Gurson's Model Method .....	347
11.2. Solving Fracture Mechanics Problems .....	347
11.2.1. Modeling the Crack-Tip Region .....	348
11.2.1.1. Modeling 2-D Linear Elastic Fracture Problems .....	349
11.2.1.2. Modeling 3-D Linear Elastic Fracture Problems .....	350
11.2.2. Calculating Fracture Parameters .....	350
11.3. Numerical Evaluation of Fracture Mechanics Parameters .....	350
11.3.1. J-Integral Calculation .....	351
11.3.1.1. Understanding the Domain Integral Method .....	351
11.3.1.1.1. Virtual Crack-Extension Nodes and J-Integral Contours .....	352
11.3.1.1.2. Element Selection and Material Behavior .....	352
11.3.1.2. J-Integral Calculation .....	353
11.3.1.2.1. Step 1: Initiate a New J-Integral Calculation .....	353
11.3.1.2.2. Step 2: Define Crack Information .....	353
11.3.1.2.2.1. Define the Crack-Tip Node Component and Crack-Plane Normal .....	354
11.3.1.2.2.2. Define the Crack-Extension Node Component and Crack-Extension Direc- tion .....	354
11.3.1.2.3. Step 3: Specify the Number of Contours to Calculate .....	356
11.3.1.2.4. Step 4: Define a Crack Symmetry Condition .....	356
11.3.1.2.5. Step 5: Specify Output Controls .....	356
11.3.2. VCCT Energy-Release Rate Calculation .....	356
11.3.2.1. Using VCCT for Energy-Release Rate Calculation .....	357
11.3.2.1.1. 2-D Crack Geometry .....	357
11.3.2.1.2. 3-D Crack Geometry .....	357
11.3.2.1.3. Element Support, Mesh and Material Behavior .....	358
11.3.2.2. Process for Calculating the Energy-Release Rate .....	359
11.3.2.2.1. Step 1: Initiate a New Energy-Release Rate Calculation .....	359
11.3.2.2.2. Step 2: Define Crack Information .....	359
11.3.2.2.2.1. Specifying Crack Information When the Crack Plane Is Flat .....	359
11.3.2.2.2.2. Specifying Crack Information When the Crack Plane Is Not Flat .....	361
11.3.2.2.3. Step 3: Define a Crack Symmetry Condition .....	362
11.3.2.2.4. Step 4: Specify Output Controls .....	363
11.3.3. Stress-Intensity Factor Calculation .....	363
11.3.3.1. Calculating Stress-Intensity Factors via Interaction Integrals .....	363
11.3.3.1.1. Understanding Interaction Integral Formulation .....	363
11.3.3.1.2. Calculating the Stress-Intensity Factors .....	364
11.3.3.1.2.1. Step 1: Initiate a New Stress-Intensity Factors Calculation .....	364
11.3.3.1.2.2. Step 2: Define Crack Information .....	364
11.3.3.1.2.3. Step 3: Specify the Number of Contours .....	368
11.3.3.1.2.4. Step 4: Define a Crack Symmetry Condition .....	368
11.3.3.1.2.5. Step 5: Specify Output Controls .....	368
11.3.3.2. Calculating Stress-Intensity Factors via Displacement Extrapolation .....	369
11.3.3.2.1. Step 1: Define a Local Crack-Tip or Crack-Front Coordinate System .....	369
11.3.3.2.2. Step 2: Define a Path Along the Crack Face .....	369
11.3.3.2.3. Step 3: Calculate $K_I$ , $K_{II}$ , and $K_{III}$ .....	369
11.3.4. T-Stress Calculation .....	370
11.3.4.1. T-Stress Interaction Integral Formulation .....	370
11.3.4.2. Element Selection and Material Behavior .....	370

11.3.4.3. Calculating the T-Stress .....	371
11.3.5. Material Force Calculation .....	371
11.3.5.1. Understanding the Material Force Approach .....	371
11.3.5.1.1. Virtual Crack-Extension Nodes and Material Force Contours .....	373
11.3.5.1.2. Element Selection and Material Behavior .....	374
11.3.5.2. Calculating Material Force .....	374
11.3.5.2.1. Step 1: Initiate a New Material Force Calculation .....	374
11.3.5.2.2. Step 2: Define Crack Information .....	375
11.3.5.2.2.1. Define the Crack-Tip Node Component and Crack-Plane Normal .....	375
11.3.5.2.2.2. Define the Crack-Extension Node Component and Crack-Extension Direc- tion .....	376
11.3.5.2.3. Step 3: Specify the Number of Contours to Calculate .....	377
11.3.5.2.4. Step 4: Define a Crack Symmetry Condition .....	377
11.3.5.2.5. Step 5: Specify Output Controls .....	378
11.4. Learning More About Fracture Mechanics .....	378
<b>12. Interface Delamination and Failure Simulation .....</b>	<b>381</b>
12.1. VCCT-Based Crack Growth Simulation .....	381
12.1.1. VCCT Crack Growth Simulation Process .....	382
12.1.1.1. Step 1. Create a Finite Element Model with a Predefined Crack Path .....	382
12.1.1.2. Step 2. Perform the Energy-Release Rate Calculation .....	383
12.1.1.3. Step 3. Perform the Crack Growth Calculation .....	383
12.1.1.3.1. Step 3a. Initiate the Crack Growth Set .....	383
12.1.1.3.2. Step 3b. Specify the Crack Path .....	383
12.1.1.3.3. Step 3c. Specify the Crack-Calculation ID and Fracture Criterion .....	383
12.1.1.3.4. Step 3d: Specify Solution Controls for Crack Growth .....	384
12.1.1.4. Example: Crack Growth Set Definition .....	385
12.1.2. Crack Extension .....	385
12.1.3. Fracture Criteria .....	386
12.1.3.1. Critical Energy-Release Rate Criterion .....	386
12.1.3.2. Linear Fracture Criterion .....	387
12.1.3.3. Bilinear Fracture Criterion .....	388
12.1.3.4. B-K Fracture Criterion .....	388
12.1.3.5. Modified B-K Fracture Criterion .....	389
12.1.3.6. Power Law Fracture Criterion .....	390
12.1.3.7. User-Defined Fracture Criterion .....	391
12.1.4. Example Crack Growth Simulation .....	393
12.1.5. VCCT Crack Growth Simulation Assumptions .....	397
12.2. Modeling Interface Delamination with Interface Elements .....	397
12.2.1. Analyzing Interface Delamination .....	398
12.2.2. Interface Elements .....	398
12.2.2.1. Element Definition .....	399
12.2.2.2. Element Selection .....	399
12.2.3. Material Definition .....	399
12.2.3.1. Material Characteristics .....	399
12.2.3.2. Material Constants -- Exponential Law .....	400
12.2.3.3. Material Constants -- Bilinear Law .....	400
12.2.3.4. Viscous Regularization for Cohesive Cone Material (CZM) .....	401
12.2.4. Meshing and Boundary Conditions .....	401
12.2.4.1. Meshing .....	401
12.2.4.2. Boundary Conditions .....	402
12.2.5. Solution Procedure and Result Output .....	402
12.2.6. Reviewing the Results .....	402



12.2.6.1. Points to Remember .....	403
12.2.6.2. Reviewing Results in POST1 .....	403
12.2.6.3. Reviewing Results in POST26 .....	404
12.3. Modeling Interface Delamination with Contact Elements .....	404
12.3.1. Analyzing Debonding .....	404
12.3.2. Contact Elements .....	404
12.3.3. Material Definition .....	405
12.3.3.1. Material Characteristics .....	405
12.3.3.2. Material Constants .....	405
12.3.4. Result Output .....	406
<b>13. Composites</b> .....	409
13.1. Modeling Composites .....	409
13.1.1. Selecting the Proper Element Type .....	409
13.1.1.1. Other Element Types with Composite Capabilities .....	410
13.1.2. Defining the Layered Configuration .....	410
13.1.2.1. Specifying Individual Layer Properties .....	411
13.1.2.2. Sandwich and Multiple-Layered Structures .....	412
13.1.2.3. Node Offset .....	412
13.1.3. Specifying Failure Criteria for Composites .....	413
13.1.3.1. Using the FC Family of Commands .....	413
13.1.3.2. User-Written Failure Criteria .....	414
13.1.4. Composite Modeling and Postprocessing Tips .....	414
13.1.4.1. Dealing with Coupling Effects .....	414
13.1.4.2. Obtaining Accurate Interlaminar Shear Stresses .....	414
13.1.4.3. Verifying Your Input Data .....	414
13.1.4.4. Specifying Results File Data .....	415
13.1.4.5. Selecting Elements with a Specific Layer Number .....	416
13.1.4.6. Specifying a Layer for Results Processing .....	416
13.1.4.7. Transforming Results to Another Coordinate System .....	416
13.2. The FiberSIM-ANSYS Interface .....	416
13.2.1. Understanding the FiberSIM XML File .....	417
13.2.2. Using FiberSIM Data in ANSYS .....	419
13.2.3. FiberSIM-to-ANSYS Translation Details .....	420
<b>14. Fatigue</b> .....	423
14.1. How Fatigue Is Calculated .....	423
14.2. Fatigue Terminology .....	424
14.3. Evaluating Fatigue .....	424
14.3.1. Enter POST1 and Resume Your Database .....	424
14.3.2. Establish the Size, Fatigue Material Properties, and Locations .....	425
14.3.3. Store Stresses and Assign Event Repetitions and Scale Factors .....	426
14.3.3.1. Storing Stresses .....	426
14.3.3.1.1. Manually Stored Stresses .....	427
14.3.3.1.2. Nodal Stresses from Jobname .RST .....	428
14.3.3.1.3. Stresses at a Cross-Section .....	428
14.3.3.2. Listing, Plotting, or Deleting Stored Stresses .....	429
14.3.3.3. Assigning Event Repetitions and Scale Factors .....	429
14.3.3.4. Guidelines for Obtaining Accurate Usage Factors .....	430
14.3.4. Activate the Fatigue Calculations .....	432
14.3.5. Review the Results .....	432
14.3.6. Other Approaches to Range Counting .....	432
14.3.7. Sample Input .....	433
<b>15. Beam Analysis and Cross Sections</b> .....	435

15.1. Overview of Cross Sections .....	435
15.2. How to Create Cross Sections .....	436
15.2.1. Defining a Section and Associating a Section ID Number .....	436
15.2.2. Defining Cross Section Geometry and Setting the Section Attribute Pointer .....	437
15.2.2.1. Determining the Number of Cells to Define .....	437
15.2.3. Meshing a Line Model with BEAM188 or BEAM189 Elements .....	438
15.3. Creating Cross Sections .....	438
15.3.1. Using the Beam Tool to Create Common Cross Sections .....	439
15.3.2. Creating Custom Cross Sections with a User-defined Mesh .....	439
15.3.2.1. Line Element Size .....	440
15.3.3. Creating Custom Cross Sections with Mesh Refinement and Multiple Materials .....	440
15.3.4. Defining Composite Cross Sections .....	441
15.3.5. Defining a Tapered Beam .....	441
15.4. Using Nonlinear General Beam Sections .....	443
15.4.1. Defining a Nonlinear General Beam Section .....	444
15.4.1.1. Strain Dependencies .....	444
15.4.2. Considerations for Using Nonlinear General Beam Sections .....	445
15.5. Using Preintegrated Composite Beam Sections .....	445
15.5.1. Defining a Composite Beam Section .....	447
15.5.1.1. Matrix Input .....	447
15.5.2. Considerations for Using Composite Beam Sections .....	448
15.5.3. Example: Composite Beam Section Input .....	448
15.6. Managing Cross Section and User Mesh Libraries .....	450
15.7. Example Lateral Torsional Buckling Analysis .....	450
15.7.1. Problem Description .....	451
15.7.2. Problem Specifications .....	451
15.7.3. Problem Sketch .....	452
15.7.4. Eigenvalue Buckling and Nonlinear Collapse .....	452
15.7.5. Set the Analysis Title and Define Model Geometry .....	452
15.7.6. Define Element Type and Cross Section Information .....	453
15.7.7. Define the Material Properties and Orientation Node .....	453
15.7.8. Mesh the Line and Verify Beam Orientation .....	454
15.7.9. Define the Boundary Conditions .....	454
15.7.10. Solve the Eigenvalue Buckling Analysis .....	455
15.7.11. Solve the Nonlinear Buckling Analysis .....	456
15.7.12. Plot and Review the Results .....	456
15.7.13. Plot and Review the Section Results .....	457
15.8. Example Problem with Cantilever Beams .....	458
15.9. Where to Find Other Examples .....	459
<b>16. Shell Analysis and Cross Sections .....</b>	<b>461</b>
16.1. Understanding Cross Sections .....	461
16.2. How to Create Cross Sections .....	461
16.2.1. Defining a Section and Associating a Section ID Number .....	463
16.2.2. Defining Layer Data .....	463
16.2.3. Overriding Program Calculated Section Properties .....	463
16.2.4. Specifying a Shell Thickness Variation (Tapered Shells) .....	463
16.2.5. Setting the Section Attribute Pointer .....	463
16.2.6. Associating an Area with a Section .....	464
16.2.7. Using the Shell Tool to Create Sections .....	464
16.2.8. Managing Cross-Section Libraries .....	465
16.3. Using Preintegrated General Shell Sections .....	465
16.3.1. Defining a Preintegrated Shell Section .....	466

---

16.3.2. Considerations for Using Preintegrated Shell Sections .....	467
<b>17. Reinforcing</b> .....	469
17.1. Assumptions About Reinforcing .....	469
17.2. Modeling Options for Reinforcing .....	469
17.3. Defining Reinforcing Sections and Elements .....	470
17.3.1. Example: Discrete Reinforcing .....	471
17.3.2. Example: Smeared Reinforcing .....	472
17.4. Reinforcing Simulation and Postprocessing .....	474
<b>18. Modeling Hydrostatic Fluids</b> .....	475
18.1. Hydrostatic Fluid Element Features .....	475
18.2. Defining Hydrostatic Fluid Elements .....	476
18.3. Material Definitions and Loading .....	477
18.3.1. Fluid Materials .....	477
18.3.2. Loads and Boundary Conditions .....	478
18.4. Example Model Using Hydrostatic Fluid Elements .....	479
18.5. Results Output .....	481
A. Example Analyses with Multiple Imposed Rotations .....	483
A.1. Problem Description .....	483
A.2. Sample Inputs for Imposed Rotations .....	483
A.2.1. Sequentially Applied Rotations .....	484
A.2.2. Simultaneously Applied Rotations .....	487
Index .....	489



---

# List of Figures

1.1. Rayleigh Damping .....	6
2.1. Diagram of Allen Wrench .....	22
3.1. Diagram of a Model Airplane Wing .....	51
4.1. Harmonic Systems .....	77
4.2. Relationship Between Real/Imaginary Components and Amplitude/Phase Angle .....	82
4.3. An Unbalanced Rotating Antenna .....	83
4.4. Two-Mass-Spring-System .....	91
5.1. Examples of Load-Versus-Time Curves .....	111
5.2. Effect of Integration Time Step on Period Elongation .....	131
5.3. Transient Input vs. Transient Response .....	131
6.1. Single-Point and Multi-Point Response Spectra .....	136
6.2. Simply Supported Beam with Vertical Motion of Both Supports .....	147
6.3. Three-Beam Frame .....	177
7.1. Buckling Curves .....	180
7.2. Adjusting Variable Loads to Find an Eigenvalue of 1.0 .....	183
7.3. Bar with Hinged Ends .....	187
8.1. Common Examples of Nonlinear Structural Behavior .....	193
8.2. A Fishing Rod Demonstrates Geometric Nonlinearity .....	194
8.3. Newton-Raphson Approach .....	195
8.4. Traditional Newton-Raphson Method vs. Arc-Length Method .....	196
8.5. Load Steps, Substeps, and Time .....	197
8.6. Nonconservative (Path-Dependent) Behavior .....	198
8.7. Load Directions Before and After Deflection .....	199
8.8. Stress-Stiffened Beams .....	201
8.9. Hyperelastic Structure .....	202
8.10. Stress Relaxation and Creep .....	211
8.11. Time Hardening Creep Analysis .....	212
8.12. Viscoplastic Behavior in a Rolling Operation .....	214
8.13. Viscoelastic Behavior (Maxwell Model) .....	215
8.14. Linear Interpolation of Nonlinear Results Can Introduce Some Error .....	252
8.15. Convergence Norms Displayed By the Graphical Solution Tracking (GST) Feature .....	269
8.16. Typical Nonlinear Output Listing .....	271
8.17. Cyclic Point Load History .....	274
9.1. Flowchart of Linear Perturbation Static Analysis .....	288
9.2. Flowchart of Linear Perturbation Modal Analysis .....	289
9.3. Flowchart of Linear Perturbation Eigenvalue Buckling Analysis .....	290
9.4. Flowchart of Linear Perturbation Full Harmonic Analysis .....	291
10.1. Element Topology of a 3-D 8-Node Interface Element .....	319
10.2. Pressure vs. Closure Behavior of a Gasket Material .....	321
10.3. Gasket Material Input: Linear Unloading Curves .....	323
10.4. Gasket Material Input: Nonlinear Unloading Curves .....	324
10.5. Gasket Compression and Unloading Curves at Two Temperatures .....	327
10.6. Gasket Finite Element Model Geometry .....	329
10.7. Whole Model Mesh with Brick Element .....	330
10.8. Interface Layer Mesh .....	330
10.9. Whole Model Tetrahedral Mesh .....	330
10.10. Interface Layer Mesh with Degenerated Wedge Elements .....	331
11.1. Schematic of the Fracture Modes .....	340
11.2. Schematic of a Crack Tip .....	342
11.3. Matrix with Inclusion .....	344

11.4. Thought Experiment Proposed by Eshelby .....	344
11.5. Crack Tip and Crack Front .....	347
11.6. Singular Element Examples .....	348
11.7. Fracture Specimen and 2-D FE Model .....	349
11.8. Using Symmetry to Your Advantage .....	349
11.9. 2-D Crack Geometry Schematic .....	357
11.10. 3-D Crack Geometry Schematic .....	358
11.11. Typical Crack Face Path Definitions .....	369
11.12. Domain Enhancement via Nodal Vectors .....	372
12.1. Crack Path Discretized with Interface Elements .....	382
12.2. Crack Growth and Merging .....	384
12.3. 2-D and 3-D Crack Extension .....	385
12.4. Crack Growth of a Double-Cantilever Beam .....	393
12.5. Double-Cantilever Beam Mesh .....	393
12.6. Double-Cantilever Beam Load-Deflection Curve .....	394
12.7. Double-Cantilever Beam Contour Plot .....	394
13.1. Layered Model Showing Dropped Layer .....	411
13.2. Sandwich Construction .....	412
13.3. Layered Shell With Nodes at Midplane .....	412
13.4. Layered Shell With Nodes at Bottom Surface .....	413
13.5. Example of an Element Display .....	415
13.6. Sample <b>LAYPLOT</b> Display for [45/-45/ - 45/45] Sequence .....	415
14.1. Cylinder Wall with Stress Concentration Factors (SCFs) .....	426
14.2. Three Loadings in One Event .....	428
14.3. Surface Nodes are Identified by <b>PPATH</b> Prior to Executing <b>FSSECT</b> .....	429
15.1. Plot of a Z Cross Section .....	435
15.2. Types of Solid Section Cell Mesh .....	438
15.3. BeamTool with Subtypes Drop Down List Displayed .....	439
15.4. Lateral-Torsional Buckling of a Cantilever I-Beam .....	451
15.5. Diagram of a Beam With Deformation Indicated .....	452
16.1. Plot of a Shell Section .....	462
16.2. Shell Tool With Layup Page Displayed .....	464
16.3. Shell Tool With Section Controls Page Displayed .....	465
16.4. Shell Tool With Summary Page Displayed .....	465
17.1. Discrete Reinforcing Modeling Option .....	470
17.2. Smeared Reinforcing Modeling Option .....	470
17.3. Discrete Reinforcing Element Display (with Translucent Base Elements) .....	472
17.4. Smeared Reinforcing Element Display (with Translucent Base Elements) .....	473
17.5. Fiber Orientation Display on Smeared Reinforcing Elements .....	474
18.1. Negative and Positive Volumes for Hydrostatic Fluid Elements .....	477
18.2. Meshed Model .....	479
18.3. Close-up View of Meshed Model .....	479
1. Rotated Configuration Resulting from Sequentially Applied Rotations .....	487
2. Rotated Configuration Resulting from Simultaneously Applied Rotations .....	488

---

# List of Tables

1.1. Damping for Full Analyses .....	3
1.2. Damping for Modal and Mode Superposition Analyses .....	3
1.3. Damping for Damped Modal and QRDAMP Mode Superposition Analyses .....	4
2.1. Basic Tab Options .....	11
2.2. Sol'n Options Tab Options .....	12
2.3. Nonlinear Tab Options .....	12
2.4. Advanced NL Tab Options .....	13
2.5. Loads Applicable in a Static Analysis .....	16
3.1. Analysis Types and Options .....	40
3.2. Loads Applicable in a Modal Analysis .....	45
3.3. Load Commands for a Modal Analysis .....	45
3.4. Load Step Options (Damping) .....	46
3.5. Symmetric System Eigensolver Options .....	61
4.1. Analysis Types and Options .....	80
4.2. Applicable Loads in a Harmonic Analysis .....	83
4.3. Load Commands for a Harmonic Analysis .....	84
4.4. Load Step Options .....	85
4.5. Expansion Pass Options .....	101
5.1. Transient Tab Options .....	114
5.2. Options for the First Load Step: Mode-Superposition Analysis .....	123
5.3. Expansion Pass Options .....	126
6.1. Analysis Types and Options .....	139
6.2. Load Step Options .....	140
6.3. Solution Items Available in a PSD Analysis .....	163
6.4. Organization of Results Data from a PSD Analysis .....	165
6.5. Analysis Types and Options .....	169
15.1. Cross Section Commands .....	436
15.2. Commands for Specifying Nonlinear General Beam Section Data .....	444
15.3. Commands for Specifying Preintegrated Composite Beam Section Data .....	447
16.1. Cross-Section Commands .....	462
16.2. Commands for Specifying Preintegrated Shell Section Data .....	466





---

# Chapter 1: Overview of Structural Analyses

---

Structural analysis is probably the most common application of the finite element method. The term *structural* (or *structure*) implies not only civil engineering structures such as bridges and buildings, but also naval, aeronautical, and mechanical structures such as ship hulls, aircraft bodies, and machine housings, as well as mechanical components such as pistons, machine parts, and tools.

The following structural analysis topics are available:

- 1.1. Types of Structural Analysis
- 1.2. Elements Used in Structural Analyses
- 1.3. Material Model Interface
- 1.4. Damping
- 1.5. Solution Method

## 1.1. Types of Structural Analysis

You can perform the following types of structural analyses:

- *Static Analysis* -- Used to determine displacements, stresses, etc. under static loading conditions. Both linear and nonlinear static analyses. Nonlinearities can include plasticity, stress stiffening, large deflection, large strain, hyperelasticity, contact surfaces, and creep.
- *Modal Analysis* -- Used to calculate the natural frequencies and mode shapes of a structure. Several mode-extraction methods are available.
- *Harmonic Analysis* -- Used to determine the response of a structure to harmonically time-varying loads.
- *Transient Dynamic Analysis* -- Used to determine the response of a structure to arbitrarily time-varying loads. All nonlinearities mentioned under Static Analysis above are allowed.
- *Spectrum Analysis* -- An extension of the modal analysis, used to calculate stresses and strains due to a response spectrum or a PSD input (random vibrations).
- *Buckling Analysis* -- Used to calculate the buckling loads and determine the buckling mode shape. Both linear (eigenvalue) buckling and nonlinear buckling analyses are possible.
- *Explicit Dynamic Analysis* -- This type of structural analysis is available via the ANSYS LS-DYNA product, which provides an interface to the LS-DYNA explicit finite element program. Explicit dynamic analysis calculates fast solutions for large deformation dynamics and complex contact problems.

Several special-purpose structural analysis capabilities are available:

- Fracture mechanics
- Composites
- Fatigue
- Beam analyses and cross sections.

The primary unknowns (nodal degrees of freedom) calculated in a structural analysis are *displacements*. Other quantities such as strains, stresses, and reaction forces are then derived from the nodal displacements.

Structural analyses are available in the ANSYS Multiphysics, ANSYS Mechanical, ANSYS Structural, and ANSYS Professional products only.

## 1.2. Elements Used in Structural Analyses

Most element types are structural elements, ranging from simple spars and beams to more complex layered shells and large strain solids. Most types of structural analyses can use any of these elements.

For more information, see [Selecting Elements for Your Analysis](#) in the *Element Reference*.

## 1.3. Material Model Interface

For analyses described in this guide, if you are using the GUI, you must specify the material that you intend to simulate using an intuitive material model interface. The interface uses a hierarchical tree structure of material categories, intended to assist you in choosing the appropriate model for your analysis.

See [Material Model Interface](#) in the *Basic Analysis Guide* for information about the material model interface.

## 1.4. Damping

Damping is present in most systems and should be specified in a dynamic analysis. The following forms of damping are available:

- Global [Alpha and Beta Damping](#) (Rayleigh Damping) (**ALPHAD**, **BETAD**)
- [Material-Dependent Alpha and Beta Damping](#) (Rayleigh Damping) (**MP**,ALPD, **MP**,BETD)
- [Constant Global Damping Ratio](#) (**DMPRAT**)
- [Constant Material-Dependent Damping Ratio](#) (**MP**,DMPR)
- [Mode-Dependent Damping Ratio](#) (**MDAMP**)
- [Constant Global Structural Damping Coefficient](#) (**DMPSTR**)
- [Material-Dependent Structural Damping Coefficient](#) (**MP**,DMPR, **TB**,SDAMP)
- [Viscoelastic Material Damping](#) (**TB**,PRONY)
- [Element Damping](#) (e.g., **COMBIN14**, **COMBIN40**, **MATRIX27**, **MPC184**)

Only the constant global damping ratio and mode-dependent damping ratio are available in the ANSYS Professional program.

The structural damping coefficients lead to damping forces which are proportional to the displacements (strains). Also known as hysteretic damping, this type of damping represents the damping that may be due to internal friction of the material or in the structural connections. The other damping forms lead to damping forces which are proportional to the velocity (or frequency of vibration). Also known as viscous damping, this type of damping represents a system immersed in a fluid.

You can specify more than one form of damping in a model. The program will formulate the damping matrix [C] as the *sum* of all the specified forms of damping. For more information about damping, see [Damping Matrices in the Mechanical APDL Theory Reference](#)

The following tables show the type of damping supported for each structural analysis type:

- [Table 1.1: Damping for Full Analyses \(p. 3\)](#)
- [Table 1.2: Damping for Modal and Mode Superposition Analyses \(p. 3\)](#)
- [Table 1.3: Damping for Damped Modal and QRDAMP Mode Superposition Analyses \(p. 4\)](#)

**Table 1.1: Damping for Full Analyses**

	Rayleigh Damping		Element Damp- ing (1)	Constant Structural Damping Coefficient		Structur- al Damp- ing Coef- ficient	Viscoelast- ic Material Damping	Damping Ratio			
	Global	Material- Depend- ent		Global	Material- Depend- ent				Material- Depend- ent	Material- Dependent	Mode-De- pendent
	<b>ALPHAD</b> and <b>BETAD</b>	<b>MP</b> , <b>BETD</b> and <b>MP</b> , <b>ALPD</b>		<b>COM- BIN14</b> , <b>MAT- RIX27</b> , ...	<b>DMPSTR</b>				<b>MP</b> , <b>DM- PR</b>	<b>TB</b> , <b>PRONY</b>	<b>TB</b> , <b>SDAMP</b>
<b>Static</b>	---	---	---	---	---	---	---	---			
<b>Buckling</b>	---	---	---	---	---	---	---	---			
<b>Substructure Generation</b>	Yes	Yes	Yes	---	---	---	---	Yes (2)			
<b>Full Harmon- ic</b>	Yes	Yes	Yes	Yes	Yes	Yes	Yes	---			
<b>Full Transi- ent</b>	Yes	Yes	Yes	---	---	No (3)	---	---			

1. Includes superelement damping matrix.
2. Supported for Component Mode Synthesis (CMS) substructuring using fixed-interface method (**CM-SOPT**,**FIX**). See [Component Mode Synthesis \(CMS\)](#) for details.
3. In a full transient analysis, viscoelastic behavior is included in the constitutive relationship and not represented via damping.

**Table 1.2: Damping for Modal and Mode Superposition Analyses**

	Rayleigh Damp- ing	Damping Ratio			Structural Damping
	Global	Global	Material-Depend- ent 2	Mode-Depend- ent 2	Global
	<b>ALPHAD</b> and <b>BETAD</b>	<b>DM- PRAT</b>	<b>MP</b> , <b>DMPR</b>	<b>MDAMP</b>	<b>DMPSTR</b>

<b>Undamped Modal</b>	---	---	No (1)	---	---
Mode Superposition Harmonic	Yes	Yes	Yes (1)	Yes	Yes
Mode Superposition Transient	Yes	Yes	Yes (1)	Yes	---
<b>Spectrum</b>					
SPRS,MPRS	Yes	Yes	Yes (1)	Yes	---
PSD	Yes	Yes	Yes (1)	Yes	---
DDAM	Yes (3)	Yes (3)	Yes (3)	Yes (3)	Yes (3)

1. **MP**,DMPR specifies an effective material damping ratio. Specify it in the modal analysis (and expand the modes, **MXPAND**,,,,YES) for use in subsequent spectrum and mode-superposition analyses.
2. The material dependent damping (**MP**,DMPR) and the mode dependent damping (**MDAMP**) cannot be cumulated. Use only one method.
3. In DDAM analyses, damping input is not used when calculating mode coefficients; however, damping may be used in the mode-combination procedure. Refer to [Combination of Modes in the Mechanical APDL Theory Reference](#) for more information.

**Table 1.3: Damping for Damped Modal and QRDAMP Mode Superposition Analyses**

	Rayleigh Damping		Element Damping (1)	Damping Ratio		Structural Damping	
	Global	Material Depend-ent		Global	Mode De-pendent	Global	Material Depend-ent
	<b>ALPHAD</b> and <b>BETAD</b>	<b>MP</b> ,BETD and <b>MP</b> ,ALPD	<b>COM-BIN14</b> , <b>MAT-RIX27</b> , ...	<b>DMPRAT</b>	<b>MDAMP</b>	<b>DMPSTR</b>	<b>MP</b> ,DMPR
<b>Damped Modal (DAMP or QRDAMP)</b>	Yes	Yes	Yes	---	---	Yes (3)	Yes (3)
<b>QRDAMP Mode Superposition Harmonic</b>	Yes (2)	Yes (2)	Yes	Yes	Yes	Yes	Yes
<b>QRDAMP Mode Superposition Transient</b>	Yes (2)	Yes (2)	Yes	Yes	Yes	---	---

1. Includes superelement damping matrix
2. **ALPHAD**, **BETAD**, **MP**,ALPD, and **MP**,BETD damping must be applied in the QR Damped modal analysis portion of the mode-superposition analysis.

3. Supported by the QR Damped eigensolver when the complex mode shapes are requested ( $Cpxmod = YES$  on the **MODOPT** command). All other types of damping are ignored.

The damping ratios may be retrieved using **\*GET,,MODE,,DAMP**. They are calculated for the following analyses:

- Spectrum analysis
- Damped modal analysis
- Mode-superposition transient and harmonic analysis

After a modal analysis (**ANTYPE,MODAL**) using the unsymmetric (**MODOPT,UNSYM**), damped (**MODOPT,DAMP**) or QR Damped (**MODOPT,QRDAMP**) method, the modal damping ratios are deduced from the complex eigenvalues using Equation 14.222 in the *Mechanical APDL Theory Reference*. These frequencies appear in the last column of the complex frequencies printout.

### 1.4.1. Alpha and Beta Damping (Rayleigh Damping)

Alpha damping and Beta damping are used to define Rayleigh damping constants  $\alpha$  and  $\beta$ . The damping matrix  $[C]$  is calculated by using these constants to multiply the mass matrix  $[M]$  and stiffness matrix  $[K]$ :

$$[C] = \alpha[M] + \beta[K]$$

The **ALPHAD** and **BETAD** commands are used to specify  $\alpha$  and  $\beta$ , respectively, as decimal numbers. The values of  $\alpha$  and  $\beta$  are not generally known directly, but are calculated from modal damping ratios,  $\xi_i$ .  $\xi_i$  is the ratio of actual damping to critical damping for a particular mode of vibration,  $i$ . If  $\omega_i$  is the natural circular frequency of mode  $i$ ,  $\alpha$  and  $\beta$  satisfy the relation

$$\xi_i = \alpha/2\omega_i + \beta\omega_i/2$$

In many practical structural problems, alpha damping (or *mass damping*) may be ignored ( $\alpha = 0$ ). In such cases, you can evaluate  $\beta$  from known values of  $\xi_i$  and  $\omega_i$ , as

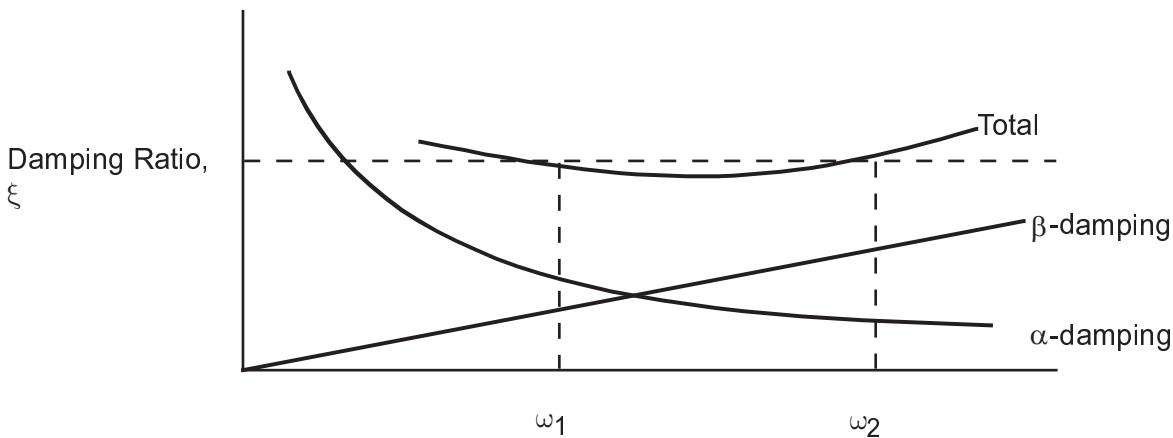
$$\beta = 2 \xi_i/\omega_i$$

Only one value of  $\beta$  can be input in a load step, so choose the most dominant frequency active in that load step to calculate  $\beta$ .

To specify both  $\alpha$  and  $\beta$  for a given damping ratio  $\xi$ , it is commonly assumed that the sum of the  $\alpha$  and  $\beta$  terms is nearly constant over a range of frequencies (see [Figure 1.1: Rayleigh Damping \(p. 6\)](#)). Therefore, given  $\xi$  and a frequency range  $\omega_1$  to  $\omega_2$ , two simultaneous equations can be solved for  $\alpha$  and  $\beta$ :

$$\alpha = 2\xi \frac{\omega_1\omega_2}{\omega_1 + \omega_2}$$

$$\beta = \frac{2\xi}{\omega_1 + \omega_2}$$

**Figure 1.1: Rayleigh Damping**

Alpha damping can lead to undesirable results if an artificially large mass has been introduced into the model. One common example is when an artificially large mass is added to the base of a structure to facilitate acceleration spectrum input. (You can use the large mass to convert an acceleration spectrum to a force spectrum.) The alpha damping coefficient, which is multiplied by the mass matrix, will produce artificially large damping forces in such a system, leading to inaccuracies in the spectrum input, as well as in the system response.

Beta damping and material damping can lead to undesirable results in a nonlinear analysis. These damping coefficients are multiplied by the stiffness matrix, which is constantly changing in a nonlinear analysis. Beta damping is not applied to the stiffness matrices generated by contact elements. The resulting change in damping can sometimes be opposite to the actual change in damping that can occur in physical structures. For example, whereas physical systems that experience softening due to plastic response will usually experience a corresponding increase in damping, an ANSYS model that has beta damping will experience a decrease in damping as plastic softening response develops.

### 1.4.2. Material-Dependent Alpha and Beta Damping (Rayleigh Damping)

Material-dependent damping allows you to specify alpha damping ( $\alpha$ ) or beta damping ( $\beta$ ) as a material property (**MP,ALPD** or **MP,BETD**). For multi-material elements such as **SOLID65**,  $\beta$  can only be specified for the element as a whole, not for each material in the element. In these cases,  $\beta$  is determined from the material pointer for the element (set with the **MAT** command), rather than the material pointed to by any real constant MAT for the element. **MP,ALPD** and **MP,BETD** are not assumed to be temperature-dependent, and are always evaluated at  $T = 0.0$ .

### 1.4.3. Constant Global Damping Ratio

The constant global damping ratio is the simplest way of specifying damping in the structure. It represents the ratio of actual damping to critical damping, and is specified as a decimal number with the **DMPRAT** command. **DMPRAT** is available only for spectrum, harmonic, and mode-superposition transient dynamic analyses. Use **MP,DMPR** to define a material dependent damping coefficient.

### 1.4.4. Constant Structural Damping Coefficient

Structural damping allows you to incorporate hysteric behavior due to internal material friction by specifying a coefficient on the stiffness matrix. In this type of damping, the damping force is proportional to the displacement rather than the velocity as in the other damping options. Both constant structural damping (**DMPSTR**) as well as material-dependent structural damping (**MP,DMPR**) are supported.

### 1.4.5. Mode-Dependent Damping Ratio

Mode Dependent Damping Ratio gives you the ability to specify different damping ratios for different modes of vibration. It is specified with the **MDAMP** command and is available only for the spectrum and mode-superposition method of solution (transient dynamic and harmonic analyses).

### 1.4.6. Viscoelastic Material Damping

**Viscoelastic materials** have a frequency-dependent complex modulus in the harmonic domain. The imaginary component of the complex modulus, also called the loss modulus, results in a material damping matrix that is added to any other forms of damping defined in the analysis.

### 1.4.7. Element Damping

Element damping involves using element types having viscous damping characteristics, including but not limited to **COMBIN14**, **COMBIN37**, and **COMBIN40**.

## 1.5. Solution Method

The ANSYS family of products uses the h-method for solving structural problems. The h-method can be used for any type of analysis.

---



---

## Chapter 2: Structural Static Analysis

---

A static analysis calculates the effects of *steady* loading conditions on a structure, while ignoring inertia and damping effects, such as those caused by time-varying loads. A static analysis can, however, include steady inertia loads (such as gravity and rotational velocity), and time-varying loads that can be approximated as static equivalent loads (such as the static equivalent wind and seismic loads commonly defined in many building codes).

Static analysis determines the displacements, stresses, strains, and forces in structures or components caused by loads that do not induce significant inertia and damping effects. Steady loading and response conditions are assumed; that is, the loads and the structure's response are assumed to vary slowly with respect to time. The types of loading that can be applied in a static analysis include:

- Externally applied forces and pressures
- Steady-state inertial forces (such as gravity or rotational velocity)
- Imposed (nonzero) displacements
- Temperatures (for thermal strain)
- Fluences (for nuclear swelling)

More information about the loads that you can apply in a static analysis appears in [Apply the Loads](#) (p. 15).

The following topics are available for structural static analysis:

- [2.1. Linear vs. Nonlinear Static Analyses](#)
- [2.2. Performing a Static Analysis](#)
- [2.3. A Sample Static Analysis \(GUI Method\)](#)
- [2.4. A Sample Static Analysis \(Command or Batch Method\)](#)
- [2.5. Where to Find Other Examples](#)

### 2.1. Linear vs. Nonlinear Static Analyses

A static analysis can be either linear or nonlinear. All types of nonlinearities are allowed - large deformations, plasticity, creep, stress stiffening, contact (gap) elements, hyperelastic elements, and so on. This chapter focuses on linear static analyses, with brief references to nonlinearities. Details of how to handle nonlinearities are described in [Nonlinear Structural Analysis](#) (p. 193).

### 2.2. Performing a Static Analysis

The procedure for a static analysis consists of these tasks:

1. [Build the Model](#) (p. 10)
2. [Set Solution Controls](#) (p. 10)

3. [Set Additional Solution Options \(p. 13\)](#)
4. [Apply the Loads \(p. 15\)](#)
5. [Solve the Analysis \(p. 18\)](#)
6. [Review the Results \(p. 19\)](#)

## 2.2.1. Build the Model

See [Building the Model](#) in the *Basic Analysis Guide*. For further details, see the *Modeling and Meshing Guide*.

### 2.2.1.1. Points to Remember

Keep the following points in mind when doing a static analysis:

- You can use both linear and nonlinear structural elements.
- Material properties can be linear or nonlinear, isotropic or orthotropic, and constant or temperature-dependent.
  - You must define stiffness in some form (for example, Young's modulus (EX), hyperelastic coefficients, and so on).
  - For inertia loads (such as gravity), you must define the data required for mass calculations, such as density (DENS).
  - For thermal loads (temperatures), you must define the coefficient of thermal expansion (ALPX).

Note the following information about mesh density:

- Regions where stresses or strains vary rapidly (usually areas of interest) require a relatively finer mesh than regions where stresses or strains are nearly constant (within an element).
- While considering the influence of nonlinearities, remember that the mesh should be able to capture the effects of the nonlinearities. For example, plasticity requires a reasonable integration point density (and therefore a fine element mesh) in areas with high plastic deformation gradients.

## 2.2.2. Set Solution Controls

Setting solution controls involves defining the analysis type and common analysis options for an analysis, as well as specifying load step options for it. When you are doing a structural static analysis, you can take advantage of a streamlined solution interface (called the Solution Controls dialog box) for setting these options. The Solution Controls dialog box provides default settings that will work well for many structural static analyses, which means that you may need to set only a few, if any, of the options. Because the streamlined solution interface is the *recommended* tool for setting solution controls in a structural static analysis, it is the method that is presented in this chapter.

If you prefer *not* to use the Solution Controls dialog box (**Main Menu**> **Solution**> **Analysis Type**> **Sol'n Controls**), you can set solution controls for your analysis using the standard set of ANSYS solution commands and the standard corresponding menu paths (**Main Menu**> **Solution**> **Unabridged Menu**> **option**). For a general overview of the Solution Controls dialog box, see [Using Special Solution Controls for Certain Types of Structural Analyses](#) in the *Basic Analysis Guide*.

### 2.2.2.1. Access the Solution Controls Dialog Box

To access the Solution Controls dialog box, choose menu path **Main Menu> Solution> Analysis Type> Sol'n Controls**. The following sections provide brief descriptions of the options that appear on each tab of the dialog box. For details about how to set these options, select the tab that you are interested in (from within the ANSYS program), and then click the **Help** button. [Nonlinear Structural Analysis \(p. 193\)](#) also contains details about the nonlinear options introduced in this chapter.

### 2.2.2.2. Using the Basic Tab

The **Basic** tab is active when you access the dialog box.

The controls that appear on the **Basic** tab provide the minimum amount of data that ANSYS needs for the analysis. 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 adjust the default settings for the more advanced controls. 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.

You can use the **Basic** tab to set the options listed in [Table 2.1: Basic Tab Options \(p. 11\)](#). For specific information about using the Solution Controls dialog box to set these options, access the dialog box, select the **Basic** tab, and click the **Help** button.

**Table 2.1: Basic Tab Options**

Option	For more information on this option, see:
Specify analysis type [ <b>ANTYPE</b> , <b>NLGEOM</b> ]	<ul style="list-style-type: none"> <li>• <a href="#">Specifying the Analysis Type and Analysis Options in the <i>Basic Analysis Guide</i></a></li> <li>• <a href="#">Nonlinear Structural Analysis (p. 193) in the <i>Structural Analysis Guide</i></a></li> <li>• <a href="#">Restarting an Analysis in the <i>Basic Analysis Guide</i></a></li> </ul>
Control time settings, including: time at end of load step [ <b>TIME</b> ], automatic time stepping [ <b>AUTOTS</b> ], and number of substeps to be taken in a load step [ <b>NSUBST</b> or <b>DELTIM</b> ]	<ul style="list-style-type: none"> <li>• <a href="#">The Role of Time in Tracking in the <i>Basic Analysis Guide</i></a></li> <li>• <a href="#">Setting General Options in the <i>Basic Analysis Guide</i></a></li> </ul>
Specify solution data to write to database [ <b>OUTRES</b> ]	<ul style="list-style-type: none"> <li>• <a href="#">Setting Output Controls in the <i>Basic Analysis Guide</i></a></li> </ul>

Special considerations for setting these options in a static analysis include:

- When setting **ANTYPE** and **NLGEOM**, choose **Small Displacement Static** if you are performing a new analysis and you want to ignore large deformation effects such as large deflection, large rotation, and large strain. Choose **Large Displacement Static** if you expect large deflections (as in the case of a long, slender bar under bending) or large strains (as in a metal-forming problem). Choose **Restart Current Analysis** if you want to restart a failed nonlinear analysis, if you have previously completed a static analysis and you want to specify additional loads, or if you wish to use the `JOBNAME.RSX` information from a previous VT Accelerator run. Note that in a VT Accelerator run, you cannot restart a job in the middle; you can only rerun the job from the beginning with changes in the input parameters.

- When setting **TIME**, remember that this load step option specifies time at the end of the load step. The default value is 1.0 for the first load step. For subsequent load steps, the default is 1.0 plus the time specified for the previous load step. Although time has no physical meaning in a static analysis (except in the case of creep, viscoplasticity, or other rate-dependent material behavior), it is used as a convenient way of referring to load steps and substeps (see [Loading](#) in the *Basic Analysis Guide*).

### 2.2.2.3. The Transient Tab

The **Transient** tab contains transient analysis controls; it is available only if you choose a transient analysis and remains grayed out when you choose a static analysis. For these reasons, it is not described here.

### 2.2.2.4. Using the Sol'n Options Tab

You can use the **Sol'n Options** tab to set the options listed in [Table 2.2: Sol'n Options Tab Options \(p. 12\)](#). For specific information about using the Solution Controls dialog box to set these options, access the dialog box, select the **Sol'n Options** tab, and click the **Help** button.

**Table 2.2: Sol'n Options Tab Options**

Option	For more information about this option, see the following section(s) in the <i>Basic Analysis Guide</i> :
Specify equation solver [ <b>EQSLV</b> ]	<ul style="list-style-type: none"> <li>Selecting a Solver</li> </ul>
Specify parameters for multiframe restart [ <b>RESCONTROL</b> ]	<ul style="list-style-type: none"> <li>Multiframe Restart</li> </ul>

Special considerations for setting these options in a static analysis include:

- When setting **EQSLV**, specify one of these solvers:
  - Program chosen solver (ANSYS selects a solver for you, based on the physics of the problem)
  - Sparse direct solver (default for linear and nonlinear, static and full transient analyses)
  - Preconditioned Conjugate Gradient (PCG) solver (recommended for large size models, bulky structures)
  - Iterative solver (auto-select; for linear static/full transient structural or steady-state thermal analyses only; recommended)

### 2.2.2.5. Using the Nonlinear Tab

You can use the **Nonlinear** tab to set the options listed in [Table 2.3: Nonlinear Tab Options \(p. 12\)](#). For specific information about using the Solution Controls dialog box to set these options, access the dialog box, select the **Nonlinear** tab, and click the **Help** button.

**Table 2.3: Nonlinear Tab Options**

Option	For more information about this option, see the following section(s) in the <i>Structural Analysis Guide</i> :
Activate line search [ <b>LNSRCH</b> ]	<ul style="list-style-type: none"> <li>Line Search Option (p. 247)</li> <li>Line Search (p. 270)</li> </ul>

Option	For more information about this option, see the following section(s) in the <i>Structural Analysis Guide</i> :
Activate a predictor on the DOF solution [ <b>PRED</b> ]	<ul style="list-style-type: none"> <li>Predictor-Corrector Option (p. 246)</li> </ul>
Activate an advanced predictor ( <b>STAOPT</b> )	<ul style="list-style-type: none"> <li>VT Accelerator (p. 246)</li> </ul>
Specify the maximum number of iterations allowed per substep [ <b>NEQIT</b> ]	<ul style="list-style-type: none"> <li>Maximum Number of Equilibrium Iterations (p. 246)</li> </ul>
Specify whether you want to include creep calculation [ <b>RATE</b> ]	<ul style="list-style-type: none"> <li>Creep Material Model (p. 210)</li> <li>Creep Criteria (p. 249)</li> </ul>
Set convergence criteria [ <b>CNVTOL</b> ]	<ul style="list-style-type: none"> <li>Convergence Criteria (p. 245)</li> </ul>
Control bisections [ <b>CUTCONTROL</b> ]	<ul style="list-style-type: none"> <li>Cutback Criteria (p. 247)</li> </ul>

### 2.2.2.6. Using the Advanced NL Tab

You can use the **Advanced NL** tab to set the options listed in [Table 2.4: Advanced NL Tab Options \(p. 13\)](#). For specific information about using the Solution Controls dialog box to set these options, access the dialog box, select the **Advanced NL** tab, and click the **Help** button.

**Table 2.4: Advanced NL Tab Options**

Option	For more information about this option, see the following section(s) in the <i>Structural Analysis Guide</i> :
Specify analysis termination criteria [ <b>NCNV</b> ]	<ul style="list-style-type: none"> <li>Maximum Number of Equilibrium Iterations (p. 246)</li> </ul>
Control activation and termination of the arc-length method [ <b>ARCLEN, ARCTRM</b> ]	<ul style="list-style-type: none"> <li>Using the Arc-Length Method (p. 263)</li> <li>Loading in the <i>Basic Analysis Guide</i></li> </ul>

### 2.2.3. Set Additional Solution Options

This section discusses additional options that you can set for the solution. These options do not appear on the Solution Controls dialog box because they are used very infrequently, and their default settings rarely need to be changed. ANSYS menu paths are provided in this section to help you access these options for those cases in which you choose to override the ANSYS-assigned defaults.

Many of the options that appear in this section are nonlinear options, and are described further in [Nonlinear Structural Analysis \(p. 193\)](#).

#### 2.2.3.1. Stress Stiffening Effects

Most element types include stress stiffening effects automatically when **NLGEOM** is ON. To determine whether an element includes stress stiffening, refer to the appropriate element description in the [Element Reference](#).

### 2.2.3.2. Newton-Raphson Option

Use this analysis option only in a nonlinear analysis. This option specifies how often the tangent matrix is updated during solution. You can specify one of these values:

- Program-chosen (default)
- Full
- Modified
- Initial stiffness
- Full with unsymmetric matrix

**Command(s):** **NROPT**

**GUI:** **Main Menu> Solution> Unabridged Menu> Analysis Options**

### 2.2.3.3. Prestress Effects Calculation

Use this analysis option to perform a prestressed analysis on the same model when the base analysis is linear (such as a prestressed modal analysis). The prestress effects calculation controls the generation of the stress stiffness matrix. The default for the prestress effects calculation is OFF.

**Command(s):** **PSTRES**

**GUI:** **Main Menu> Solution> Unabridged Menu> Analysis Options**

If the base analysis is nonlinear, the [linear perturbation procedure](#) is recommended. In this case, the prestress effects are automatically included and **PSTRES** is not needed.

### 2.2.3.4. Mass Matrix Formulation

Use this analysis option if you plan to apply inertial loads on the structure (such as gravity and spinning loads). You can specify one of these values:

- Default (depends on element type)
- Lumped mass approximation

---

#### Note

For a static analysis, the mass matrix formulation you use does not significantly affect the solution accuracy (assuming that the mesh is fine enough). However, if you want to do a prestressed dynamic analysis on the same model, the choice of mass matrix formulation may be important; see the appropriate dynamic analysis section for recommendations.

**Command(s):** **LUMPM**

**GUI:** **Main Menu> Solution> Unabridged Menu> Analysis Options**

---

### 2.2.3.5. Reference Temperature

This load step option is used for thermal strain calculations. Reference temperature can be made material-dependent with the **MP,REFT** command.

**Command(s):** **TREF**

**GUI:** **Main Menu> Solution> Load Step Opts> Other> Reference Temp**

### 2.2.3.6. Mode Number

This load step option is used for axisymmetric harmonic elements.

**Command(s):** **MODE**

**GUI:** Main Menu> Solution> Load Step Opts> Other> For Harmonic Ele

### 2.2.3.7. Creep Criteria

This nonlinear load step option specifies the creep criterion for automatic time stepping.

**Command(s):** **CRPLIM**

**GUI:** Main Menu> Solution> Unabridged Menu> Load Step Opts> Nonlinear> Creep Criterion

### 2.2.3.8. Printed Output

Use this load step option to include any results data on the output file (Jobname.OUT).

**Command(s):** **OUTPR**

**GUI:** Main Menu> Solution> Unabridged Menu> Load Step Opts> OutputCtrls> Solu Printout

---

#### Caution

Proper use of multiple **OUTPR** commands can sometimes be a little tricky. See [Setting Output Controls](#) in the *Basic Analysis Guide* for more information on how to use this command.

---

### 2.2.3.9. Extrapolation of Results

Use this load step option to review element integration point results by copying them to the nodes instead of extrapolating them (default when no material nonlinearities are present).

**Command(s):** **ERESX**

**GUI:** Main Menu> Solution> Unabridged Menu> Load Step Opts> OutputCtrls> Integration Pt

## 2.2.4. Apply the Loads

After you set the desired solution options, you are ready to apply loads to the model.

### 2.2.4.1. Load Types

All of the following load types are applicable in a static analysis.

#### 2.2.4.1.1. Displacements (UX, UY, UZ, ROTX, ROTY, ROTZ)

These are DOF constraints usually specified at model boundaries to define rigid support points. They can also indicate symmetry boundary conditions and points of known motion. The directions implied by the labels are in the nodal coordinate system.

#### 2.2.4.1.2. Velocities (VELX, VELY, VELZ, OMGX, OMGY, OMGZ)

The displacement constraints can be replaced by the equivalent differentiation forms, which are the corresponding velocity loads. If a velocity load is present, the displacement constraint at the current time step is calculated as the displacement constraint at the previous time step plus the input velocity value times the time step. For example, if VELX is input the ux constraint is:  $ux(t+dt) = ux(t) + v(t)*dt$ .

The directions implied by the velocity load labels are in the nodal coordinate system.



### 2.2.4.1.3. Forces (FX, FY, FZ) and Moments (MX, MY, MZ)

These are concentrated loads usually specified on the model exterior. The directions implied by the labels are in the nodal coordinate system.

### 2.2.4.1.4. Pressures (PRES)

These are surface loads, also usually applied on the model exterior. Positive values of pressure act towards the element face (resulting in a compressive effect).

### 2.2.4.1.5. Temperatures (TEMP)

These are applied to study the effects of thermal expansion or contraction (that is, thermal stresses). The coefficient of thermal expansion must be defined if thermal strains are to be calculated. You can read in temperatures from a thermal analysis [LDREAD], or you can specify temperatures directly, using the BF family of commands.

### 2.2.4.1.6. Fluences (FLUE)

These are applied to study the effects of swelling (material enlargement due to neutron bombardment or other causes) or creep. They are used only if you input a swelling or creep equation.

### 2.2.4.1.7. Gravity, Spinning, Etc.

These are inertia loads that affect the entire structure. Density (or mass in some form) must be defined if inertia effects are to be included.

## 2.2.4.2. Apply Loads to the Model

Except for inertia loads (which are independent of the model) and velocity loads, you can define loads either on the solid model (keypoints, lines, and areas) or on the finite element model (nodes and elements). You can also apply boundary conditions via TABLE type array parameters (see [Applying Loads Using TABLE Type Array Parameters](#) (p. 17)) or as function boundary conditions (see [Using the Function Tool](#)).

[Table 2.5: Loads Applicable in a Static Analysis](#) (p. 16) summarizes the loads applicable to a static analysis. In an analysis, loads can be applied, removed, operated on, or listed.

**Table 2.5: Loads Applicable in a Static Analysis**

Load Type	Category	For details on commands and menu paths for defining these loads, see...
Displacement (UX, UY, UZ, ROTX, ROTY, ROTZ)	Con-straints	<a href="#">DOF Constraints</a> in the <i>Basic Analysis Guide</i>
Velocities (VELX, VELY, VELZ, OMGX, OMGY, OMGZ)	Con-straints	<a href="#">DOF Constraints</a> in the <i>Basic Analysis Guide</i>
Force, Moment (FX, FY, FZ, MX, MY, MZ)	Forces	<a href="#">Forces (Concentrated Loads)</a> in the <i>Basic Analysis Guide</i>
Pressure (PRES)	Surface Loads	<a href="#">Surface Loads</a> in the <i>Basic Analysis Guide</i>
Temperature (TEMP), Fluence (FLUE)	Body Loads	<a href="#">Applying Body Loads</a> in the <i>Basic Analysis Guide</i>



Load Type	Category	For details on commands and menu paths for defining these loads, see...
Gravity, Spinning, and so on	Inertia Loads	<a href="#">Applying Inertia Loads</a> in the <i>Basic Analysis Guide</i>

### 2.2.4.2.1. Applying Loads Using TABLE Type Array Parameters

You can also apply loads using TABLE type array parameters. For details on using tabular boundary conditions, see [Applying Loads Using TABLE Type Array Parameters](#) in the *Basic Analysis Guide*.

In a structural analysis, valid primary variables are TIME, TEMP, and location (X, Y, Z).

When defining the table, TIME must be in ascending order in the table index (as in any table array).

You can define a table array parameter via command or interactively. For more information on defining table array parameters, see the *ANSYS Parametric Design Language Guide*.

### 2.2.4.3. Calculating Inertia Relief

You can use a static analysis to perform *inertia relief* calculations, which calculate the accelerations that will counterbalance the applied loads. You can think of inertia relief as an equivalent free-body analysis.

Your model should meet the following requirements:

- The model should *not* contain axisymmetric or generalized plane strain elements, or nonlinearities. Models with a mixture of 2-D and 3-D element types are not recommended.
- Symmetry models are not valid for inertia relief analysis.
- Data required for mass calculations (such as density) must be specified.
- Specify only the minimum number of displacement constraints - those required to prevent rigid-body motion. Three constraints (or fewer, depending on the element type) are necessary for 2-D models and six (or fewer) are necessary for 3-D models. Additional constraints, such as those required to impose symmetry conditions, are permitted, but check for zero reaction forces at all the constraints to make sure that the model is not overconstrained for inertia relief.
- The loads for which inertia relief calculations are desired should be applied.

Issue the **IRLF** command *before* the **SOLVE** command as part of the inertia load commands.

**Command(s):** **IRLF**,1

**GUI:** Main Menu> Solution> Load Step Opts> Other> Inertia Relief

#### Inertia Relief for Substructures

For *substructures*, inertia relief calculations (**MATRIX50**) use the equations described in [Inertia Relief](#) in the *Mechanical APDL Theory Reference*. ANSYS obtains the mass matrix of a substructure via [matrix reduction](#) to condense it to the master nodes (**MASTER**). The inertia relief calculations in a substructure are therefore consistent with the reduced mass contribution at the master nodes.

The **IRLF** command has no effect in the [generation pass](#) of a substructure. If you intend to perform inertia relief calculations on a substructure, do not apply DOF constraints (**D**) on the substructure during its generation pass; instead, apply them during the [use pass](#). (Otherwise, the substructure reduction

logic condenses out the mass associated with the constrained DOFs in the generation pass, and the inertia relief calculations in the use pass of the substructure reflect the *condensed* mass distribution.)

The choice of master nodes during the generation pass will have a critical effect on how well the mass is represented in the condensed substructure mass matrix. When you choose the Master Degrees of Freedom nodes to represent a 'bounding box' of the model, the substructure's inertia relief calculations should closely represent that of the model. To verify your Master dof selections, check the reaction forces (**PRRSOL** in **/POST1**) to ensure they are close to zero.

In the **expansion pass**, precalculation of masses for summary printout (**IRLF,-1**) occurs only on elements that are part of the substructure.

### 2.2.4.3.1. Inertia Relief Output

Use the **IRLIST** command to print the output from inertia relief calculations. This output consists of the translational and rotational accelerations required to balance the applied loads and can be used by other programs to perform kinematics studies. The summary listing of mass and moments of inertia (produced during solution) is accurate, not approximate. The reaction forces at the constraints will be zero because the calculated inertia forces balance the applied forces.

Inertia relief output is stored in the database rather than in the results file (`Jobname.RST`). When you issue **IRLIST**, ANSYS pulls the information from the database, which contains the inertia relief output from the most recent solution [**SOLVE**].

**Command(s): IRLIST**

**GUI:** No GUI equivalent.

### 2.2.4.3.2. Using a Macro to Perform Inertia Relief Calculations

If you need to do inertia relief calculations frequently, you can write a macro containing the above commands. Macros are described in the *ANSYS Parametric Design Language Guide*.

## 2.2.5. Solve the Analysis

You are now ready to solve the analysis.

1. Save a backup copy of the database to a named file. You can then retrieve your model by reentering the ANSYS program and issuing **RESUME**.

**Command(s): SAVE**

**GUI: Utility Menu> File> Save as**

2. Start solution calculations.

**Command(s): SOLVE**

**GUI: Main Menu> Solution> Solve> Current LS**

3. If you want the analysis to include additional loading conditions (that is, multiple load steps), you will need to repeat the process of applying loads, specifying load step options, saving, and solving for each load step. (Other methods for handling multiple load steps are described in [Loading](#) in the *Basic Analysis Guide*.)

4. Leave SOLUTION.

**Command(s): FINISH**

**GUI:** Close the Solution menu.

## 2.2.6. Review the Results

Results from a static analysis are written to the structural results file, `Jobname.RST`. They consist of the following data:

- Primary data:
  - Nodal displacements (UX, UY, UZ, ROTX, ROTY, ROTZ)
- Derived data:
  - Nodal and element stresses
  - Nodal and element strains
  - Element forces
  - Nodal reaction forces
  - and so on

### 2.2.6.1. Postprocessors

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

- POST1 is used to review results over the entire model at specific substeps (time-points). Some typical POST1 operations are explained below.
- POST26 is used in nonlinear static analyses to track specific result items over the applied load history. See [Nonlinear Structural Analysis \(p. 193\)](#) for the use of POST26 in a nonlinear static analysis. For a complete description of all postprocessing functions, see [An Overview of Postprocessing](#) in the *Basic Analysis Guide*.

### 2.2.6.2. Points to Remember

- To review results in POST1 or POST26, the database must contain the same model for which the solution was calculated.
- The results file (`Jobname.RST`) must be available.

### 2.2.6.3. Reviewing Results Data

1. Read in the database from the database file.
 

**Command(s): RESUME**  
**GUI: Utility Menu > File > Resume from**
2. Read in the desired set of results. Identify the data set by load step and substep numbers or by time. (If you specify a time value for which no results are available, the ANSYS program will perform linear interpolation on all the data to calculate the results at that time.)
 

**Command(s): SET**  
**GUI: Main Menu > General Postproc > Read Results > By Load Step**
3. Perform the necessary POST1 operations. Typical static analysis POST1 operations are explained below.

## 2.2.6.4. Typical Postprocessing Operations

### Option: Display Deformed Shape

Use the **PLDISP** command to display a deformed shape (**Main Menu**> **General Postproc**> **Plot Results**> **Deformed Shape**). The *KUND* field on **PLDISP** gives you the option of overlaying the undeformed shape on the display.

### Option: List Reaction Forces and Moments

The **PRRSOL** command lists reaction forces and moments at the constrained nodes (**Main Menu**> **General Postproc**> **List Results**> **Reaction Solu**).

To display reaction forces, issue **/PBC,RFOR,,1** and then request a node or element display [**NPLOT** or **EPLOT**]. (Use **RMOM** instead of **RFOR** for reaction moments.)

### Option: List Nodal Forces and Moments

Use the **PRESOL,F** (or **M**) command to list nodal forces and moments (**Main Menu**> **General Postproc**> **List Results**> **Element Solution**).

You can list the sum of all nodal forces and moments for a selected set of nodes. Select a set of nodes and use this feature to find out the total force acting on those nodes:

**Command(s): FSUM**

**GUI: Main Menu**> **General Postproc**> **Nodal Calcs**> **Total Force Sum**

You can also check the total force and total moment at each selected node. For a body in equilibrium, the total load is zero at all nodes except where an applied load or reaction load exists:

**Command(s): NFORCE**

**GUI: Main Menu**> **General Postproc**> **Nodal Calcs**> **Sum @ Each Node**

The **FORCE** command (**Main Menu**> **General Postproc**> **Options for Outp**) dictates which component of the forces is being reviewed:

- Total (default)
- Static component
- Damping component
- Inertia component

For a body in equilibrium, the total load (using all **FORCE** components) is zero at all nodes except where an applied load or reaction load exists.

### Option: Line Element Results

For line elements, such as beams, spars, and pipes, use **ETABLE** to gain access to derived data (stresses, strains, and so on) (**Main Menu**> **General Postproc**> **Element Table**> **Define Table**). Results data are identified by a combination of a label and a sequence number or component name on the **ETABLE** command. See the **ETABLE** discussion in *The General Postprocessor (POST1)* in the *Basic Analysis Guide* for details.

### Option: Error Estimation

For linear static analyses using solid or shell elements, use the **PRERR** command to list the estimated solution error due to mesh discretization (**Main Menu**> **General Postproc**> **List Results**> **Percent Error**). This command calculates and lists the percent error in structural energy norm (SEPC), which represents the error relative to a particular mesh discretization.

*Option: Structural Energy Error Estimation*

Use **PLESOL**,**SERR** to contour the element-by-element *structural energy error* (SERR) (**Main Menu**> **General Postproc**> **Plot Results**> **Contour Plot**> **Element Solu**). Regions of high SERR on the contour display are good candidates for mesh refinement. (You can activate automatic mesh refinement by means of the **ADAPT** command - see the *Modeling and Meshing Guide* for more information.) See *Estimating Solution Error* in the *Basic Analysis Guide* for more details about error estimation.

*Option: Contour Displays*

Use **PLNSOL** and **PLESOL** to contour almost any result item, such as stresses (SX, SY, SZ...), strains (EPELX, EPELY, EPELZ...), and displacements (UX, UY, UZ...) (**Main Menu**> **General Postproc**> **Plot Results**> **Contour Plot**> **Nodal Solu** or **Element Solu**).

The *KUND* field on **PLNSOL** and **PLESOL** gives you the option of overlaying the undeformed shape on the display.

Use **PLETAB** and **PLLS** to contour element table data and line element data (**Main Menu**> **General Postproc**> **Element Table**> **Plot Element Table** and **Main Menu**> **General Postproc**> **Plot Results**> **Contour Plot**> **Line Elem Res**).

---

### Caution

Derived data, such as stresses and strains, are averaged at the nodes by the **PLNSOL** command. This averaging results in "smeared" values at nodes where elements of different materials, different shell thicknesses, or other discontinuities meet. To avoid the smearing effect, use selecting (described in *Selecting and Components* in the *Basic Analysis Guide*) to select elements of the same material, same shell thickness, and so on before issuing **PLNSOL**. Alternatively, use PowerGraphics with the **AVRES** command (**Main Menu**> **General Postproc**> **Options for Outp**) to not average results across different materials and/or different shell thicknesses.

---

*Option: Vector Displays*

Use **PLVECT** to view vector displays (**Main Menu**> **General Postproc**> **Plot Results**> **Vector Plot**> **Predefined**) and **PRVECT** to view vector listings (**Main Menu**> **General Postproc**> **List Results**> **Vector Data**).

Vector displays (not to be confused with vector *mode*) are an effective way of viewing vector quantities, such as displacement (DISP), rotation (ROT), and principal stresses (S1, S2, S3).

*Option: Tabular Listings*

Use these commands to produce tabular listings:

**Command(s):** **PRNSOL** (nodal results), **PRESOL** (element-by-element results) **PRRSOL** (reaction data), and so on

**GUI:** **Main Menu**> **General Postproc**> **List Results**> *solution option*

Use the **NSORT** and **ESORT** commands to sort the data before listing them (**Main Menu**> **General Postproc**> **List Results**> **Sorted Listing**> **Sort Nodes** or **Sort Elems**).

### Other Postprocessing Capabilities

Many other postprocessing functions - mapping results onto a path, load case combinations, and so on - are available in POST1. See [An Overview of Postprocessing](#) in the *Basic Analysis Guide* for details.

## 2.3. A Sample Static Analysis (GUI Method)

In this sample analysis, you will run a static analysis of an Allen wrench.

### 2.3.1. Problem Description

An Allen wrench (10 mm across the flats) is torqued by means of a 100 N force at its end. Later, a 20 N downward force is applied at the same end, at the same time retaining the original 100 N torquing force. The objective is to determine the stress intensity in the wrench under these two loading conditions.

### 2.3.2. Problem Specifications

The following dimensions are used for this problem:

Width across flats = 10 mm

Configuration = hexagonal

Length of shank = 7.5 cm

Length of handle = 20 cm

Bend radius = 1 cm

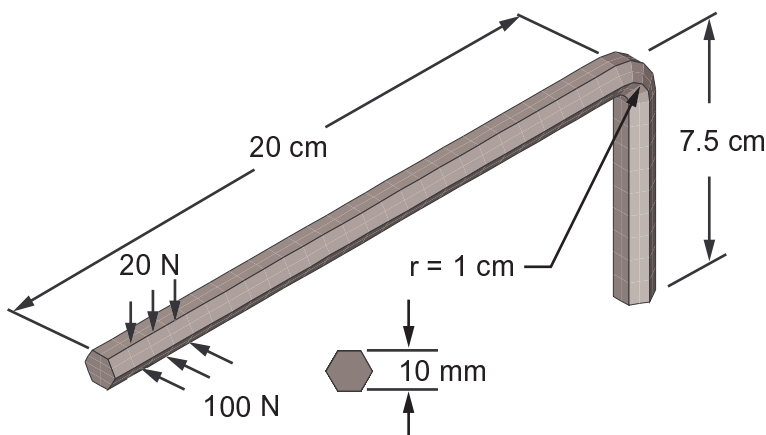
Modulus of elasticity =  $2.07 \times 10^{11}$  Pa

Applied torquing force = 100 N

Applied downward force = 20 N

### 2.3.3. Problem Sketch

**Figure 2.1: Diagram of Allen Wrench**



### 2.3.3.1. Set the Analysis Title

1. Choose menu path **Utility Menu> File> Change Title**.
2. Type the text "Static Analysis of an Allen Wrench" and click on OK.

### 2.3.3.2. Set the System of Units

1. Click once in the Input Window to make it active for text entry.
2. Type the command **/UNITS,SI** and press ENTER. Notice that the command is stored in the history buffer, which can be accessed by clicking on the down arrow at the right of the input window.
3. Choose menu path **Utility Menu> Parameters> Angular Units**. The Angular Units for Parametric Functions dialog box appears.
4. In the drop down menu for Units for angular parametric functions, select "Degrees DEG."
5. Click on OK.

### 2.3.3.3. Define Parameters

1. Choose menu path **Utility Menu> Parameters> Scalar Parameters**. The Scalar Parameters dialog box appears.
2. Type the following parameters and their values in the Selection field. Click on Accept after you define each parameter. For example, first type "exx = 2.07e11" in the Selection field and then click on Accept. Continue entering the remaining parameters and values in the same way.

Parameter	Value	Description
EXX	2.07E11	Young's modulus is 2.07E11 Pa
W_HEX	.01	Width of hex across flats = .01 m
W_FLAT	W_HEX* TAN(30)	Width of flat = .0058 m
L_SHANK	.075	Length of shank (short end) .075 m
L_HANDLE	.2	Length of handle (long end) .2 m
BENDRAD	.01	Bend radius .01 m
L_ELEM	.0075	Element length .0075 m
NO_D_HEX	2	Number of divisions along hex flat = 2
TOL	25E-6	Tolerance for selecting node = 25E-6 m

#### Note

You can type the labels in upper- or lowercase; ANSYS always displays the labels in uppercase.

3. Click on Close.
4. Click on SAVE\_DB on the Toolbar.

### **2.3.3.4. Define the Element Types**

1. Choose menu path **Main Menu> Preprocessor> Element Type> Add/Edit/Delete**.
2. Click on Add. The Library of Element Types dialog box appears.
3. In the scroll box on the left, click once on "Structural Solid."
4. In the scroll box on the right, click once on "Brick 8node 185."
5. Click on OK to define it as element type 1. The Library of Element Types dialog box closes.
6. Click on Options. The **SOLID185** element type options dialog box appears.
7. In the element technology scroll box, scroll to "Simple Enhanced Str" and select it.
8. Click OK. The element type options dialog box closes. Click on Add in the element types box.
9. Scroll up the list on the right to "Quad 4node 182." Click once to select it.
10. Click on OK to define Quad 4node182 as element type 2. The Library of Element Types dialog box closes.
11. Click on Options. The **PLANE182** element type options dialog box appears.
12. In the scroll box for element technology, scroll to "Simple Enhanced Str" and select it.
13. Click on Close in the Element Types dialog box.

### **2.3.3.5. Define Material Properties**

1. Choose menu path **Main Menu> Preprocessor> Material Props> Material Models**. The Define Material Model Behavior dialog box appears.
2. In the Material Models Available window, double-click on the following options: Structural, Linear, Elastic, Isotropic. A dialog box appears.
3. Type the text EXX in the EX field (for Young's modulus), and .3 for PRXY.  
  
Click on **OK**. This sets Young's modulus to the parameter specified above. Material Model Number 1 appears in the Material Models Defined window on the left.
4. Choose menu path **Material> Exit** to remove the Define Material Model Behavior dialog box.

### **2.3.3.6. Create Hexagonal Area as Cross-Section**

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Areas> Polygon> By Side Length**. The Polygon by Side Length dialog box appears.
2. Enter 6 for number of sides.
3. Enter W\_FLAT for length of each side.
4. Click on OK. A hexagon appears in the ANSYS Graphics window.



### 2.3.3.7. Create Keypoints Along a Path

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Keypoints> In Active CS**. The Create Keypoints in Active Coordinate System dialog box appears.
2. Enter 7 for keypoint number. Type a 0 in each of the X, Y, Z location fields.
3. Click on Apply.
4. Enter 8 for keypoint number.
5. Enter 0,0,-L\_SHANK for the X, Y, Z location, and click on Apply.
6. Enter 9 for keypoint number.
7. Enter 0,L\_HANDLE,-L\_SHANK for the X, Y, Z location, and click on OK.

### 2.3.3.8. Create Lines Along a Path

1. Choose menu path **Utility Menu> PlotCtrls> Window Controls> Window Options**. The Window Options dialog box appears.
2. In the Location of triad drop down menu, select "At top left."
3. Click on OK.
4. Choose menu path **Utility Menu> PlotCtrls> Pan/Zoom/Rotate**. The Pan-Zoom-Rotate dialog box appears.
5. Click on "Iso" to generate an isometric view and click on Close.
6. Choose menu path **Utility Menu> PlotCtrls> View Settings> Angle of Rotation**. The Angle of Rotation dialog box appears.
7. Enter 90 for angle in degrees.
8. In the Axis of rotation drop down menu, select "Global Cartes X."
9. Click on OK.
10. Choose menu path **Utility Menu> PlotCtrls> Numbering**. The Plot Numbering Controls dialog box appears.
11. Click the Keypoint numbers radio button to turn keypoint numbering on.
12. Click the Line numbers radio button to turn line numbering on.
13. Click on OK.
14. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Lines> Lines> Straight Line**. The Create Straight Line picking menu appears.
15. Click once on keypoints 4 and 1 to create a line between keypoints 1 and 4. (If you have trouble reading the keypoint numbers in the ANSYS Graphics window, use the controls on the Pan-Zoom-Rotate dialog box (**Utility Menu> PlotCtrls> Pan/Zoom/Rotate**) to zoom in.)
16. Click once on keypoints 7 and 8 to create a line between keypoints 7 and 8.

17. Click once on keypoints 8 and 9 to create a line between keypoints 8 and 9.

18. Click on OK.

### **2.3.3.9. Create Line from Shank to Handle**

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Lines> Line Fillet**. The Line Fillet picking menu appears.
2. Click once on lines 8 and 9.
3. Click on OK in the picking menu. The Line Fillet dialog box appears.
4. Enter BENDRAD for Fillet radius and click on OK.
5. Click on SAVE\_DB on the Toolbar.

### **2.3.3.10. Cut Hex Section**

In this step, you cut the hex section into two quadrilaterals. This step is required to satisfy mapped meshing.

1. Choose menu path **Utility Menu> PlotCtrls> Numbering**. The Plot Numbering Controls dialog box appears.
2. Click the Keypoint numbers radio button to Off.
3. Click on OK.
4. Choose menu path **Utility Menu> Plot> Areas**.
5. Choose menu path **Main Menu> Preprocessor> Modeling> Operate> Booleans> Divide> With Options> Area by Line**. The Divide Area by Line picking menu appears.
6. Click once on the shaded area, and click on OK.
7. Choose menu path **Utility Menu> Plot> Lines**.
8. Click once on line 7. (If you have trouble reading the line numbers in the ANSYS Graphics window, use the controls on the Pan-Zoom-Rotate dialog box (**Utility Menu> PlotCtrls> Pan/Zoom/Rotate**) to zoom in.)
9. Click on OK. The Divide Area by Line with Options dialog box appears. In the Subtracted lines will be drop down menu, select Kept. Click OK.
10. Choose menu path **Utility Menu> Select> Comp/Assembly> Create Component**. The Create Component dialog box appears.
11. Enter BOTAREA for component name.
12. In the Component is made of drop down menu, select "Areas."
13. Click on OK.

### 2.3.3.11. Set Meshing Density

1. Choose menu path **Main Menu> Preprocessor> Meshing> Size Cntrls> Lines> Picked Lines**. The Element Size on Picked Lines picking menu appears.
2. Enter 1,2,6 in the picker, then press ENTER.
3. Click on OK in the picking menu. The Element Sizes on Picked Lines dialog box appears.
4. Enter NO\_D\_HEX for number of element divisions and click on OK.

### 2.3.3.12. Set Element Type for Area Mesh

In this step, set the element type to **PLANE182**, all quadrilaterals for the area mesh.

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Elements> Elem Attributes**. The Element Attributes dialog box appears.
2. In the Element type number drop down menu, select "2 **PLANE182**" and click on OK.
3. Choose menu path **Main Menu> Preprocessor> Meshing> Mesher Opts**. The Mesher Options dialog box appears.
4. In the Mesher Type field, click on the Mapped radio button and then click on OK. The Set Element Shape dialog box appears.
5. Click on OK to accept the default of Quad for 2-D shape key.
6. Click on SAVE\_DB on the Toolbar.

### 2.3.3.13. Generate Area Mesh

In this step, generate the area mesh you will later drag.

1. Choose menu path **Main Menu> Preprocessor> Meshing> Mesh> Areas> Mapped> 3 or 4 sided**. The Mesh Areas picking box appears.
2. Click on Pick All.
3. Choose menu path **Utility Menu> Plot> Elements**.

### 2.3.3.14. Drag the 2-D Mesh to Produce 3-D Elements

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Elements> Elem Attributes**. The Element Attributes dialog box appears.
2. In the Element type number drop down menu, select "1 **SOLID185**" and click on OK.
3. Choose menu path **Main Menu> Preprocessor> Meshing> Size Cntrls> Global> Size**. The Global Element Sizes dialog box appears.
4. Enter L\_ELEM for element edge length and click on OK.
5. Choose menu path **Utility Menu> PlotCtrls> Numbering**.

6. Click the Line numbers radio button to on if it is not already selected.
7. Click on OK.
8. Choose menu path **Utility Menu> Plot> Lines**.
9. Choose menu path **Main Menu> Preprocessor> Modeling> Operate> Extrude> Areas> Along Lines**. The Sweep Areas along Lines picking box appears.
10. Click on Pick All. A second picking box appears.
11. Click once on lines 8, 10, and 9 (in that order).
12. Click on OK. The 3-D model appears in the ANSYS Graphics window.
13. Choose menu path **Utility Menu> Plot> Elements**.
14. Click on SAVE\_DB on the Toolbar.

### **2.3.3.15. Select BOTAREA Component and Delete 2-D Elements**

1. Choose menu path **Utility Menu> Select> Comp/Assembly> Select Comp/Assembly**. The Select Component or Assembly dialog appears.
2. Click on OK to accept the default of select BOTAREA component.
3. Choose menu path **Main Menu> Preprocessor> Meshing> Clear> Areas**. The Clear Areas picking menu appears.
4. Click on Pick All.
5. Choose menu path **Utility Menu> Select> Everything**.
6. Choose menu path **Utility Menu> Plot> Elements**.

### **2.3.3.16. Apply Displacement Boundary Condition at End of Wrench**

1. Choose menu path **Utility Menu> Select> Comp/Assembly> Select Comp/Assembly**. The Select Component or Assembly dialog appears.
2. Click on OK to accept the default of select BOTAREA component.
3. Choose menu path **Utility Menu> Select> Entities**. The Select Entities dialog box appears.
4. In the top drop down menu, select "Lines."
5. In the second drop down menu, select "Exterior."
6. Click on Apply.
7. In the top drop down menu, select "Nodes."
8. In the second drop down menu, select "Attached to."
9. Click on the "Lines, all" radio button to select it.

10. Click on OK.
11. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On Nodes**. The Apply U,ROT on Nodes picking menu appears.
12. Click on Pick All. The Apply U,ROT on Nodes dialog box appears.
13. In the scroll list for DOFs to be constrained, click on "ALL DOF."
14. Click on OK.
15. Choose menu path **Utility Menu> Select> Entities**.
16. In the top drop down menu, select "Lines."
17. Click on the "Sele All" button, then click on Cancel.

### **2.3.3.17. Display Boundary Conditions**

1. Choose menu path **Utility Menu> PlotCtrls> Symbols**. The Symbols dialog box appears.
2. Click on the "All Applied BCs" radio button for Boundary condition symbol.
3. In the Surface Load Symbols drop down menu, select "Pressures."
4. In the "Show pres and convect as" drop down menu, select "Arrows."
5. Click on OK.

### **2.3.3.18. Apply Pressure on Handle**

In this step, apply pressure on the handle to represent 100 N finger force.

1. Choose menu path **Utility Menu> Select> Entities**. The Select Entities dialog appears.
2. In the top drop down menu, select "Areas."
3. In the second drop down menu, select "By Location."
4. Click on the "Y coordinates" radio button to select it.
5. Enter BENDRAD,L\_HANDLE for Min, Max, and click on Apply.
6. Click on "X coordinates" to select it.
7. Click on Reselect.
8. Enter W\_FLAT/2,W\_FLAT for Min, Max, and click on Apply.
9. In the top drop down menu, select "Nodes."
10. In the second drop down menu, select "Attached to."
11. Click on the "Areas, all" radio button to select it.
12. Click on the "From Full" radio button to select it.

13. Click on Apply.
14. In the second drop down menu, select "By Location."
15. Click on the "Y coordinates" radio button to select it.
16. Click on the "Reselect" radio button.
17. Enter  $L\_HANDLE+TOL, L\_HANDLE-(3.0*L\_ELEM)-TOL$  for Min, Max.
18. Click on OK.
19. Choose menu path **Utility Menu> Parameters> Get Scalar Data**. The Get Scalar Data dialog box appears.
20. In the scroll box on the left, scroll to "Model Data" and select it.
21. In the scroll box on the right, scroll to "For selected set" and select it.
22. Click on OK. The Get Data for Selected Entity Set dialog box appears.
23. Enter "minyval" for the name of the parameter to be defined.
24. In the scroll box on the left, click once on "Current node set" to select it.
25. In the scroll box on the right, click once on "Min Y coordinate" to select it.
26. Click on Apply.
27. Click on OK again to select the default settings. The Get Data for Selected Entity Set dialog box appears.
28. Enter "maxyval" for the name of the parameter to be defined.
29. In the scroll box on the left, click once on "Current node set" to select it.
30. In the scroll box on the right, click once on "Max Y coordinate" to select it.
31. Click on OK.
32. Choose menu path **Utility Menu> Parameters> Scalar Parameters**. The Scalar Parameters dialog box appears.
33. Type the text  $PTORQ=100/(W\_HEX*(MAXYVAL-MINYVAL))$  in the Selection text box and click on Accept.
34. Click on Close.
35. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Pressure> On Nodes**. The Apply PRES on Nodes picking menu appears.
36. Click on Pick All. The Apply PRES on Nodes dialog box appears.
37. Enter PTORQ for Load PRES value and click on OK.
38. Choose menu path **Utility Menu> Select> Everything**.
39. Choose menu path **Utility Menu> Plot> Nodes**.

40. Click on SAVE\_DB on the Toolbar.

### 2.3.3.19. Write the First Load Step

1. Choose menu path **Main Menu> Solution> Load Step Opts> Write LS File**. The Write Load Step File dialog appears.
2. Enter 1 for load step file number n.
3. Click on OK.

### 2.3.3.20. Define Downward Pressure

In this step, you define the downward pressure on top of the handle, representing 20N (4.5 lb) of force.

1. Choose menu path **Utility Menu> Parameters> Scalar Parameters**. The Scalar Parameters dialog box appears.
2. Type the text  $PDOWN=20/(W\_FLAT*(MAXYVAL-MINYVAL))$  in the Selection text box and click on Accept.
3. Click on Close.
4. Choose menu path **Utility Menu> Select> Entities**. The Select Entities dialog appears.
5. In the top drop down menu, select "Areas."
6. In the second drop down menu, select "By Location."
7. Click on the "Z coordinates" radio button to select it.
8. Click on the "From Full" radio button to select it.
9. Enter  $-(L\_SHANK+(W\_HEX/2))$  for Min, Max.
10. Click on Apply.
11. In the top drop down menu, select "Nodes."
12. In the second drop down menu, select "Attached to."
13. Click on the Areas, all radio button to select it, and click on Apply.
14. In the second drop down menu, select "By Location."
15. Click on the "Y coordinates" radio button to select it.
16. Click on the "Reselect" radio button.
17. Enter  $L\_HANDLE+TOL, L\_HANDLE-(3.0*L\_ELEM)-TOL$  for Min, Max.
18. Click on OK.
19. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Pressure> On Nodes**. The Apply PRES on Nodes picking menu appears.

20. Click on Pick All. The Apply PRES on Nodes dialog box appears.
21. Enter PDOWN for Load PRES value and click on OK.
22. Choose menu path **Utility Menu> Select> Everything**.
23. Choose menu path **Utility Menu> Plot> Nodes**.

### **2.3.3.21. Write Second Load Step**

1. Choose menu path **Main Menu> Solution> Load Step Opts> Write LS File**. The Write Load Step File dialog box appears.
2. Enter 2 for Load step file number n, and click on OK.
3. Click on SAVE\_DB on the Toolbar.

### **2.3.3.22. Solve from Load Step Files**

1. Choose menu path **Main Menu> Solution> Solve> From LS Files**. The Solve Load Step Files dialog box appears.
2. Enter 1 for Starting LS file number.
3. Enter 2 for Ending LS file number, and click on OK.
4. Click on the Close button after the Solution is done! window appears.

### **2.3.3.23. Read First Load Step and Review Results**

1. Choose menu path **Main Menu> General Postproc> Read Results> First Set**.
2. Choose menu path **Main Menu> General Postproc> List Results> Reaction Solu**. The List Reaction Solution dialog box appears.
3. Click on OK to accept the default of All Items.
4. Review the information in the status window, and click on Close.
5. Choose menu path **Utility Menu> PlotCtrls> Symbols**. The Symbols dialog box appears.
6. Click on the "None" radio button for Boundary condition symbol, and click on OK.
7. Choose menu path **Utility Menu> PlotCtrls> Style> Edge Options**. The Edge Options dialog box appears.
8. In the Element outlines for non-contour/contour plots drop down menu, select "Edge Only/All."
9. Click on OK.
10. Choose menu path **Main Menu> General Postproc> Plot Results> Deformed Shape**. The Plot Deformed Shape dialog box appears.
11. Click on the "Def + undeformed" radio button and click on OK.
12. Choose menu path **Utility Menu> PlotCtrls> Save Plot Ctrl**s. The Save Plot Controls dialog box appears.



13. Type "pldisp.gsa" in the Selection box, and click on OK.
14. Choose menu path **Utility Menu> PlotCtrls> View Settings> Angle of Rotation**. The Angle of Rotation dialog box appears.
15. Enter 120 for Angle in degrees.
16. In the Relative/absolute drop down menu, select "Relative angle."
17. In the Axis of rotation drop down menu, select "Global Cartes Y."
18. Click on OK.
19. Choose menu path **Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu**. The Contour Nodal Solution Data dialog box appears.
20. In the scroll box on the left, click on "Stress." In the scroll box on the right, click on "Intensity SINT."
21. Click on OK.
22. Choose menu path **Utility Menu> PlotCtrls> Save PlotCtrls**. The Save Plot Controls dialog box appears.
23. Type "plnsol.gsa" in the Selection box, and click on OK.

### **2.3.3.24. Read the Next Load Step and Review Results**

1. Choose menu path **Main Menu> General Postproc> Read Results> Next Set**.
2. Choose menu path **Main Menu> General Postproc> List Results> Reaction Solu**. The List Reaction Solution dialog box appears.
3. Click on OK to accept the default of All Items.
4. Review the information in the status window, and click on Close.
5. Choose menu path **Utility Menu> PlotCtrls> Restore PlotCtrls**.
6. Type "pldisp.gsa" in the Selection box, and click on OK.
7. Choose menu path **Main Menu> General Postproc> Plot Results> Deformed Shape**. The Plot Deformed Shape dialog box appears.
8. Click on the "Def + undeformed" radio button if it is not already selected and click on OK.
9. Choose menu path **Utility Menu> PlotCtrls> Restore PlotCtrls**.
10. Type "plnsol.gsa" in the Selection box, and click on OK.
11. Choose menu path **Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu**. The Contour Nodal Solution Data dialog box appears.
12. In the scroll box on the left, click on "Stress." In the scroll box on the right, scroll to "Intensity SINT" and select it.
13. Click on OK.

### 2.3.3.25. Zoom in on Cross-Section

1. Choose menu path **Utility Menu> WorkPlane> Offset WP by Increments**. The Offset WP tool box appears.
2. Enter 0,0,-0.067 for X, Y, Z Offsets and click on OK.
3. Choose menu path **Utility Menu> PlotCtrls> Style> Hidden Line Options**. The Hidden-Line Options dialog box appears.
4. In the drop down menu for Type of Plot, select "Capped hidden."
5. In the drop down menu for Cutting plane is, select "Working plane."
6. Click on OK.
7. Choose menu path **Utility Menu> PlotCtrls> Pan-Zoom-Rotate**. The Pan-Zoom-Rotate tool box appears.
8. Click on "WP."
9. Drag the Rate slider bar to 10.
10. On the Pan-Zoom-Rotate dialog box, click on the large round dot several times to zoom in on the cross section.

### 2.3.3.26. Exit ANSYS

1. Choose QUIT from the Toolbar.
2. Choose Quit - No Save!
3. Click on OK.

## 2.4. A Sample Static Analysis (Command or Batch Method)

You can perform the example static analysis of an Allen wrench using the ANSYS commands shown below instead of GUI choices. Items prefaced with an exclamation point (!) are comments.

```

! define parameters for future use
exx=2.07e11      ! young's modulus (2.07e11 pa = 30e6 psi)
w_hex=.01       ! width of hex across flats (.01m=.39in)
*afun,deg       ! units for angular parametric functions
w_flat=w_hex*tan(30) ! width of flat
l_shank=.075    ! length of shank (short end) (.075m=3.0in)
l_handle=.2     ! length of handle (long end) (.2m=7.9 in)
bendrad=.01    ! bend radius of allen wrench (.01m=.39 in)
l_elem=.0075   ! element length (.0075 m = .30 in)
no_d_hex=2     ! number of divisions on hex flat
tol=25e-6      ! tolerance for selecting nodes (25e-6 m = .001 in)

/prep7
et,1,solid185,,3 ! eight-node brick element using simplified enhanced
                ! strain formulation
et,2,plane182,3 ! four-node quadrilateral (for area mesh)using
                ! simplified enhanced strain formulation
mp,ex,1,exx     ! young's modulus for material 1
mp,prxy,1,0.3   ! poisson's ratio for material 1
rpoly,6,w_flat  ! hexagonal area
k,7             ! keypoint at (0,0,0)
k,8,,, -l_shank ! keypoint at shank-handle intersection
k,9,,l_handle,-l_shank ! keypoint at end of handle

```

```

1,4,1          ! line through middle of hex shape
1,7,8          ! line along middle of shank
1,8,9          ! line along handle
ifillt,8,9,bendrad ! line along bend radius between shank and
                ! handle
/view,,1,1,1   ! isometric view in window 1
/angle,,90,xm  ! rotates model 90 degrees about x
/pnum,line,1   ! line numbers turned on
lplot
/pnum,line,0   ! line numbers off
1,1,4          ! hex section is cut into two quadrilaterals
asbl,1,7,,keep ! to satisfy mapped meshing requirements for bricks
cm,botarea,area ! component name botarea for the two areas

! generate area mesh for later drag

lesize,1,,no_d_hex ! number of divisions along line 1
lesize,2,,no_d_hex
lesize,6,,no_d_hex
type,2           ! plane182 elements to be meshed first
mshape,0,2d     ! mapped quad mesh
mshkey,1
save            ! save database before meshing
amesh,all
/title,meshed hex wrench end to be used in vdrag
eplot

! now drag the 2-d mesh to produce 3-d elements

type,1          ! type pointer set to solid185
esize,1_elem   ! element size
vdrag,2,3,,,,,8,10,9 ! drag operation to create 3-d mesh
/type,,hidp    ! precise hidden line display
/title,meshed hex wrench
eplot
cmsel,,botarea ! select botarea component and
aclear,all     ! delete the 2-d elements
asel,all
finish

! apply loads and obtain the solution

/solu
antype,static ! static analysis (default)
/title,allen wrench -- load step 1

! first fix all nodes around bottom of shank

cmsel,,botarea ! bottom areas of shank
lsel,,ext      ! exterior lines of those areas
nsll,,1        ! nodes on those lines
d,all,all     ! displacement constraints
lsel,all
/psc,u,,1     ! displacement symbols turned on
/title,boundary conditions on end of wrench
nplot

!now apply pressure on handle to represent 100-n (22.5-lb) finger
!force

asel,,loc,y,bendrad,l_handle ! areas on handle
asel,r,loc,x,w_flat/2,w_flat ! two areas on one side of handle...
nsla,,1        ! ...and all corresponding nodes
nssel,r,loc,y,l_handle+tol,l_handle-(3.0*l_elem)-tol ! reselects
                ! nodes at back end of handle (3
                ! element lengths)
*get,minyval,node,,mnloc,y ! get minimum y value of selected
                ! nodes
*get,maxyval,node,,mxloc,y ! get maximum y value of selected
                ! nodes
ptorq=100/(w_hex*(maxyval-minyval)) ! pressure equivalent to 100 n
sf,all,pres,ptorq ! ptorq pressure on all selected

```

```

! nodes
allsel ! restores full set of all entities
/psf,pres,,2 ! pressure symbols turned on
/title,boundary conditions on wrench for load step 1
nplot
lswrite ! writes first load step
/title, allen wrench -- load step 2

! downward pressure on top of handle, representing 20-n (4.5 -lb)
! force

pdown=20/(w_flat*(maxyval-minyval))

asel,,loc,z,-(l_shank+(w_hex/2)) ! area on top flat of handle...
nsla,,1 ! ...and all corresponding nodes
nselect,r,loc,y,l_handle+tol,l_handle-(3.0*l_elem)-tol ! reselects
! nodes at back end of handle (3
! element lengths)
sf,all,pres,pdown ! pdown pressure at all selected
! nodes

allsel
/title,boundary conditions on wrench for load step 2
nplot
lswrite ! writes second load step
save ! save database before solution
lssolve,1,2 ! initiates solution for load step files 1
! and 2

finish

!review the results
/show,jpeg
/post1
set,1 ! reads load step 1 results
prrsol ! reaction solution listing
/pcb,defa ! no bc symbols
/psf,defa ! no surface load symbols
/edge,,1 ! edges only, no interior element outlines
/title,deformed allen wrench caused by torque

pldisp,2 ! deformed shape overlaid with undeformed
! edge plot
/gsave,pldisp,gsav ! saves graphics specifications on
! pldisp.gsav
/plopts,info,on ! turns on entire legend column
/plopts,leg1,off ! turns off legend header
/angle,,120,ym,1 ! additional rotation about model y (to see
! high stress areas)
/title,stress intensity contours caused by torque
plnsol,s,int ! stress intensity contours
*get,smax,plnsol,0,max
*stat,smax
/gsave,plnsol,gsav ! saves graphics specifications to
! plnsol.gsav
set,2 ! reads load step 2 results
prrsol ! reaction solution listing
/gresume,pldisp,gsav ! resumes graphics specifications from
! pldisp.gsav
/title,deformed allen wrench caused by torque and force
pldisp,2
/gresume,plnsol,gsav ! resumes graphics specifications from
! plnsol.gsav
/title,stress intensity contours caused by torque and force
plnsol,s,int
*get,smax,plnsol,0,max
*stat,smax
wpof,,,-0.067 ! offset the working plane for cross-section
! view
/type,1,5 ! capped hidden display
/cplane,1 ! cutting plane defined to use the wp
/view,1,wp ! view will be normal to the wp
/dist,1,.01 ! zoom in on the cross section
/title,cross section of the allen wrench under torque and force loading

```

```
plnsol,s,int
*get,smax,plnsol,0,max
*stat,smax
finish
```

## 2.5. Where to Find Other Examples

Several ANSYS publications, particularly the *Mechanical APDL Verification Manual* and the *Mechanical APDL Introductory Tutorials*, describe additional structural static analyses.

The *Mechanical APDL Verification Manual* consists of test case analyses demonstrating the analysis capabilities of the ANSYS family of products. While these test cases demonstrate solutions to realistic analysis problems, the *Mechanical APDL Verification Manual* does not present them as step-by-step examples with lengthy data input instructions and printouts. However, most ANSYS users who have at least limited finite element experience should be able to fill in the missing details by reviewing each test case's finite element model and input data with accompanying comments.

The *Mechanical APDL Verification Manual* includes the following structural static analysis test cases:

- VM1 - Statically Indeterminate Reaction Force Analysis
- VM2 - Beam Stresses and Deflections
- VM4 - Deflection of a Hinged Support
- VM11 - Residual Stress Problem
- VM12 - Combined Bending and Torsion
- VM13 - Cylindrical Shell Under Pressure
- VM16 - Bending of a Solid Beam
- VM18 - Out-of-plane Bending of a Curved Bar
- VM20 - Cylindrical Membrane Under Pressure
- VM25 - Stresses in a Long Cylinder
- VM29 - Friction on a Support Block
- VM31 - Cable Supporting Hanging Loads
- VM36 - Limit Moment Analysis
- VM39 - Bending of a Circular Plate with a Center Hole
- VM41 - Small Deflection of a Rigid Beam
- VM44 - Bending of an Axisymmetric Thin Pipe Under Gravity Loading
- VM53 - Vibration of a String Under Tension
- VM59 - Lateral Vibration of an Axially Loaded Bar
- VM63 - Static Hertz Contact Problem
- VM78 - Transverse Shear Stresses in a Cantilever Beam
- VM82 - Simply Supported Laminated Plate Under Pressure
- VM127 - Buckling of a Bar with Hinged Ends
- VM135 - Bending of a Beam on an Elastic Foundation
- VM141 - Diametric Compression of a Disk
- VM148 - Bending of a Parabolic Beam
- VM183 - Harmonic Response of a Spring-Mass System
- VM199 - Viscoplastic Analysis of a Body Undergoing Shear Deformation
- VM201 - Rubber Cylinder Pressed Between Two Plates
- VM206 - Stranded Coil with Voltage Excitation
- VM211 - Rubber Cylinder Pressed Between Two Plates
- VM216 - Lateral Buckling of a Right-Angle Frame



---

## Chapter 3: Modal Analysis

---

Use modal analysis to determine the vibration characteristics (natural frequencies and mode shapes) of a structure or a machine component while it is being designed. It can also serve as a starting point for another, more detailed, dynamic analysis, such as a transient dynamic analysis, a harmonic analysis, or a spectrum analysis.

The following modal analysis topics are available:

- 3.1. Uses for Modal Analysis
- 3.2. Understanding the Modal Analysis Process
- 3.3. Building the Model for a Modal Analysis
- 3.4. Applying Loads and Obtaining the Solution
- 3.5. Reviewing the Results
- 3.6. Applying Prestress Effects in a Modal Analysis
- 3.7. Modal Analysis Examples
- 3.8. Comparing Mode-Extraction Methods
- 3.9. Modal Analysis Tools for Subsequent Mode Superposition Analysis

### 3.1. Uses for Modal Analysis

You use modal analysis to determine the natural frequencies and mode shapes of a structure. The natural frequencies and mode shapes are important parameters in the design of a structure for dynamic loading conditions. They are also required if you want to perform a spectrum analysis or a mode-superposition harmonic or transient analysis.

You can perform a modal analysis on a prestressed structure, such as a spinning turbine blade. Another useful feature is modal cyclic symmetry, which allows you to review the mode shapes of a cyclically symmetric structure by modeling just a sector of it.

Modal analysis in the ANSYS, Inc. family of products is a linear analysis. Any nonlinearities, such as plasticity and contact (gap) elements, are ignored even if they are defined. You can select from among several mode-extraction methods: Block Lanczos, Supernode, Subspace, PCG Lanczos, unsymmetric, damped, and QR damped. The damped and QR damped methods allow you to include damping in the structure. The QR damped method also allows for unsymmetrical damping and stiffness matrices. Details about mode-extraction methods are covered later in this section.

### 3.2. Understanding the Modal Analysis Process

The general process for a modal analysis consists of these primary operations:

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

### 3.3. Building the Model for a Modal Analysis

When **building your model** with the intention of performing a modal analysis, the following conditions apply:

- Only linear behavior is valid in a modal analysis.

If you specify nonlinear elements, Mechanical APDL treats them as linear. For example, if you include contact elements, their stiffnesses are calculated based on their initial status and never change. For a prestressed modal analysis, the program assumes that the initial status of the contact elements is the status at the completion of the static prestress analysis.

- Material properties can be linear, isotropic or orthotropic, and constant or temperature-dependent.

Define both Young's modulus (EX) (or stiffness in some form) and density (DENS) (or mass in some form). Mechanical APDL ignores nonlinear properties.

- If applying element damping, define the required real constants for the specific element type (COMBIN14, COMBIN37, and so on). For more details about damping definition, see [Damping \(p. 2\)](#).

### 3.4. Applying Loads and Obtaining the Solution

In this step, you define the analysis type and options, apply loads, specify load step options, and begin the finite element solution for the natural frequencies, as follows:

- 3.4.1. Enter the Solution Processor
- 3.4.2. Define Analysis Type and Options
- 3.4.3. Apply Loads
- 3.4.4. Specify Load Step Options
- 3.4.5. Solve
- 3.4.6. Participation Factor Table Output
- 3.4.7. Exit the Solution Processor

#### 3.4.1. Enter the Solution Processor

1. Enter the solution processor.  
**Command(s): /SOLU**  
**GUI: Main Menu> Solution**

#### 3.4.2. Define Analysis Type and Options

After you have entered the solution processor, you define the analysis type and analysis options. Mechanical APDL offers the options listed in [Table 3.1: Analysis Types and Options \(p. 40\)](#) for a modal analysis. Each of the options is explained in detail below.

**Table 3.1: Analysis Types and Options**

Option	Command	GUI Path
New Analysis	<b>ANTYPE</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; New Analysis</b>
Analysis Type: Modal (see Note below)	<b>ANTYPE</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; New Analysis&gt; Modal</b>



Option	Command	GUI Path
Mode-extraction Method	<b>MODOPT</b>	Main Menu> Solution> Analysis Type> Analysis Options
Number of Modes to Extract	<b>MODOPT</b>	Main Menu> Solution> Analysis Type> Analysis Options
Number of Modes to Expand (see Note below)	<b>MXPAND</b>	Main Menu> Solution> Analysis Type> Analysis Options
Mass Matrix Formulation	<b>LUMPM</b>	Main Menu> Solution> Analysis Type> Analysis Options
Prestress Effects Calculation	<b>PSTRES</b>	Main Menu> Solution> Analysis Type> Analysis Options
Control Output to the Results File	<b>OUTRES</b>	Main Menu> Solution> Load Step Opts > OutputCtrls > DB/Results Files
Residual Vector Calculation	<b>RESVEC</b>	This command cannot be accessed from a menu.

When you specify a modal analysis, a **Solution** menu that is appropriate for modal analyses appears. The **Solution** menu is either “abridged” or “unabridged,” depending on the actions you took prior to this step in the current session. The abridged menu contains only those solution options that are valid and/or recommended for modal analyses. If you are on the abridged **Solution** menu and you want to access other solution options (that is, solution options that are valid for you to use, but their use may not be encouraged for this type of analysis), select the **Unabridged Menu** option. For details, see [Using Abridged Solution Menus](#) in the *Basic Analysis Guide*.

### 3.4.2.1. Option: New Analysis (**ANTYPE**)

Select New Analysis. If you wish to compute additional load vectors, residual vectors, or enforced motion terms subsequent to a prior modal analysis, select Restart.

### 3.4.2.2. Option: Analysis Type: Modal (**ANTYPE**)

Use this option to specify a modal analysis.

### 3.4.2.3. Option: Mode-Extraction Method (**MODOPT**)

Select one of the following mode-extraction methods:

Mode-Extraction Method	Comments
Block Lanczos	Used for large symmetric eigenvalue problems. This method uses the sparse matrix solver, overriding any solver specified via the <b>EQSLV</b> command.
PCG Lanczos	Used for very large symmetric eigenvalue problems (500,000+ degrees of freedom), and is especially useful to obtain a solution for the lowest modes to learn how the model will behave. This method uses the PCG iterative solver and therefore has the same limitations (that is, it does not support superle-

Mode-Extraction Method	Comments
	<p>ments, Lagrange multiplier option on contact elements, mixed u-P formulation elements, etc.).</p> <p>This method works with the various <i>Lev_Diff</i> values on the <b>PCGOPT</b> command. It also works with <b>MSAVE</b> to reduce memory usage.</p> <p>By default, this method does not perform a Sturm sequence check; however, internal heuristics have been developed to guard against missing modes. If a Sturm sequence check is absolutely necessary, it can be activated via the <b>PCGOPT</b> command.</p> <p>This method is the only eigenvalue solver optimized to run in a distributed manner in Distributed ANSYS.</p>
Supernode	<p>Used to solve for many modes (up to 10,000) in one solution. Typically, the reason for seeking many modes is to perform a subsequent mode-superposition or PSD analysis to solve for the response in a higher frequency range.</p> <p>This method typically offers faster solution times than Block Lanczos if the number of modes requested is more than 200. The accuracy of the solution can be controlled via the <b>SNOPTION</b> command.</p>
Subspace	<p>Used for large symmetric eigenvalue problems.</p> <p>This method uses the sparse matrix solver. The advantage of the Subspace method over the Block Lanczos method is that both [K] and [S] / [M] matrices can be indefinite at the same time. Some options can be controlled via the <b>SUBOPT</b> command.</p>
Unsymmetric	<p>Used for problems with unsymmetric matrices, such as fluid-structure interaction problems.</p>
Damped	<p>Used for problems where damping cannot be ignored, such as bearing problems.</p>
QR damped	<p>Faster and achieves better calculation efficiency than the damped method. It uses the reduced modal damped matrix to calculate complex damped frequencies in modal coordinates.</p>

For more detailed information, see [Comparing Mode-Extraction Methods \(p. 61\)](#).

For most applications, you will use the Block Lanczos, PCG Lanczos, Subspace, or Supernode method. The unsymmetric, damped, and QR damped methods are applicable in special applications. (The damped, unsymmetric, and QR damped methods may not be available, depending upon the Mechanical APDL license in use at your site.)

When you specify a mode-extraction method, the program automatically chooses the appropriate equation solver.

### 3.4.2.4. Option: Number of Modes to Extract (**MODOPT**)

This option is required for all mode-extraction methods.

For the unsymmetric and damped methods, requesting a larger number of modes than necessary reduces the possibility of missed modes, but results in more solution time.

### 3.4.2.5. Option: Number of Modes to Expand (**MXPAND**)

The **MXPAND** command specifies which extracted modes to expand (i.e., which extracted modes to write to the results file) and how to compute element quantities (e.g., stress, strains, forces, energies, etc.) for subsequent postprocessing or for downstream mode-superposition analyses (e.g., spectrum, PSD, transient, or harmonic).

You can specify the following methods of mode expansion using the **MXPAND** command:

- Input the number of modes to expand (*NMODE*).
- Select the modes using a table (input using *NMODE*).
- Select the modes based on their modal mass or mode coefficient (input as *ModeSelMethod*). Refer to [Using Mode Selection \(p. 71\)](#) for more information.
- Specify a frequency range to extract (*FREQB, FREQE*).

In a mode-superposition analysis, the DOF results can either be combined in the expansion pass and the element results calculated from the combined DOF solution, or the stresses can be combined directly during the expansion pass to reduce computation time. This behavior is controlled by *MSUPkey* at the modal analysis level.

If *MSUPkey* is set to YES (**MXPAND**,ALL,,,YES,,YES), the element results are written to the mode file and then directly combined during the expansion pass. This option is the default setting. It will reduce the computation time, but it has the following limitations for the downstream mode-superposition analyses:

- Strain and kinetic energies are not available.
- Equivalent strains are evaluated using *EFFNU*=0.0. Use **AVPRIN** to set to the appropriate Poisson's ratio value.
- Thermal loads are not supported.
- NMISC results are not available.
- Superelements are not supported (Mechanical APDL automatically sets *MSUPkey* to NO).
- To include the damping and inertia contributions for reactions in a harmonic analysis, the element nodal loads must be output (**OUTRES**,NLOAD,ALL). Otherwise the reactions will only reflect the static contributions.
- The damping element forces and reaction forces are not available in a mode superposition harmonic analysis based on QRDAMP eigensolver.
- Strains and stresses of PIPE elements are not supported when pressure load is applied.

If *MSUPkey* is set to NO (**MXPAND**,ALL,,,YES,,NO), the element results are not written to the mode file. For subsequent downstream mode-superposition analysis, the DOF results are combined in the expansion

pass and the element results are calculated from the combined DOF results. This option is used to obtain the energies or NMISC results, or it can be used to compute thermal loads in the modal analysis for use in a subsequent mode superposition harmonic or transient analysis.

#### **3.4.2.6. Option: Results File Output (OUTRES)**

Use **OUTRES** to expand only items of interest and in the areas of interest to limit the size of the results file `Jobname.RST`. Note that for subsequent mode-superposition analyses (e.g., spectrum, PSD, transient, or harmonic), you should use **OUTRES** during the modal analysis to control their output as well.

---

#### **Note**

The `FREQ` field on **OUTRES** (and **OUTPR**) can only be ALL or NONE, meaning the data can be requested for all modes or no modes. For instance, you cannot write information for every other mode.

---

#### **3.4.2.7. Option: Mass Matrix Formulation (LUMPM)**

Use this option to specify the default formulation (which is element-dependent) or lumped mass approximation. We recommend the default formulation for most applications. However, for some problems involving "skinny" structures such as slender beams or very thin shells, the lumped mass approximation often yields better results. Also, the lumped mass approximation can result in a shorter run time and lower memory requirements.

#### **3.4.2.8. Option: Prestress Effects Calculation (PSTRES)**

Use this option to calculate the modes of a prestressed structure. By default, no prestress effects are included; that is, the structure is assumed to be stress-free. To include prestress effects, element files from a previous static (or transient) analysis must be available; see [Performing a Prestressed Modal Analysis from a Linear Base Analysis](#) (p. 49). If prestress effects are turned on, the lumped mass setting (**LUMPM**) in this and subsequent solutions must be the same as it was in the prestress static analysis.

You can use only axisymmetric loads for prestressing harmonic elements such as `PLANE25` and `SHELL61`.

#### **3.4.2.9. Option: Residual Vector Calculation (RESVEC)**

Use this option to include the contribution of higher frequency modes in a subsequent mode superposition analysis. If rigid body modes are present, you must define pseudo-constraints using the **D** command with `Value=SUPPORT`. Only the minimum number of constraints must be specified. Those constraints will only be considered for the residual vector calculation.

#### **3.4.2.10. Additional Modal Analysis Options**

After you complete the fields on the Modal Analysis Options dialog box, click OK. A dialog box specific to the selected extraction method appears. You see some combination of the following fields: `FREQB`, `FREQE`, `PRMODE`, `Nrmkey`. Refer to the **MODOPT** command description for the meaning of these fields.

### **3.4.3. Apply Loads**

In a modal analysis, the only "loads" that affect the solution are displacement constraints. If you input a nonzero displacement constraint, the program assigns a zero-value constraint to that degree of freedom instead. For directions in which no constraints are specified, the program calculates rigid-body (zero-

frequency) as well as higher (nonzero frequency) free body modes. [Table 3.2: Loads Applicable in a Modal Analysis \(p. 45\)](#) shows the commands to apply displacement constraints. Notice that you can apply them either on the solid model (keypoints, lines, and areas) or on the finite element model (nodes and elements). For a general discussion of solid-model loads versus finite-element loads, see [Loading in the Basic Analysis Guide](#).

If you are going to perform a downstream harmonic or transient mode-superposition analysis, you can apply other load types as well (see [Generating and Using Multiple Loads in Mode-Superposition Analyses \(p. 68\)](#)).

### Note

Loads specified using tabular boundary conditions with TIME as the primary variable (see the **\*DIM** command) will have the table value at TIME equal to zero.

**Table 3.2: Loads Applicable in a Modal Analysis**

Load Type	Category	Cmd Family	GUI Path
Displacement (UX, UY, UZ, ROTX, ROTY, ROTZ)	Constraints	<b>D</b>	<b>Main Menu&gt; Solution&gt; Define Loads&gt; Apply&gt; Structural&gt; Displacement</b>

In an analysis, loads can be applied, removed, operated on, or listed.

### 3.4.3.1. Applying Loads Using Commands

[Table 3.3: Load Commands for a Modal Analysis \(p. 45\)](#) lists all the commands you can use to apply loads in a modal analysis.

**Table 3.3: Load Commands for a Modal Analysis**

Load Type	Solid Model or FE	Entity	Apply	Delete	List	Operate	Apply Settings
Displacement	Solid Model	Keypoints	<b>DK</b>	<b>DKDELE</b>	<b>DKLIST</b>	<b>DTRAN</b>	-
	Solid Model	Lines	<b>DL</b>	<b>DLDELE</b>	<b>DLLIST</b>	<b>DTRAN</b>	-
	Solid Model	Areas	<b>DA</b>	<b>DADELE</b>	<b>DALIST</b>	<b>DTRAN</b>	-
	Finite Elem	Nodes	<b>D</b>	<b>DDELE</b>	<b>DLIST</b>	<b>DSCALE</b>	<b>DSYM, DCUM</b>

### 3.4.3.2. Applying Loads Using the GUI

All loading operations (except List; see [Listing Loads \(p. 46\)](#)) are accessed through a series of cascading menus. From the Solution menu, you select the operation (apply, delete, and so on), then the load type (displacement, force, and so on), and then the object to which you are applying the load (keypoint, line, node, and so on).

For example, to apply a displacement load to a line, follow this GUI path:

**GUI:**

**Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On lines**

**3.4.3.3. Listing Loads**

To list existing loads, follow this GUI path:

**GUI:**

**Utility Menu> List>Loads> load type**

**3.4.4. Specify Load Step Options**

The only load step options available for a modal analysis are damping options.

**Table 3.4: Load Step Options (Damping)**

Option	Command
Global Alpha (mass) Damping	<b>ALPHAD</b>
Global Beta (stiffness) Damping	<b>BETAD</b>
Material-Dependent Alpha (mass) Damping	<b>MP,ALPD</b>
Material-Dependent Beta (stiffness) Damping	<b>MP,BETD,</b>
Element Damping (applied via element real constant or material tables)	<b>R, TB</b>
Material-Dependent Constant Structural Damping Ratio	<b>MP,DMPR</b>

Damping is valid only for the damped and QR damped mode-extraction methods. Damping is ignored for the other mode-extraction methods. Constant structural damping ratio (**MP,DMPR**) is not supported in a damped modal analysis (**MODOPT,DAMP**).

If you include damping and specify the *damped* mode-extraction method, the calculated eigenvalues and eigenvectors are complex. If you include damping and specify the *QR damped* mode-extraction method, the eigenvalues are complex, but the eigenvectors can be real or complex (*Cpxmod* on the **MODOPT** command). The real eigenvectors are used for the mode-superposition analysis. See [Comparing Mode-Extraction Methods \(p. 61\)](#) for details.

For more information about different forms of damping, see [Damping \(p. 2\)](#).

**Damping specified in a non-damped modal analysis** Damping (**MP,DMPR**) can be specified in a non-damped modal analysis if a spectrum or mode-superposition analysis is to follow the modal analysis. Although the damping does not affect the eigenvalue solution, it is used to calculate an effective damping ratio for each mode, which is then used in the subsequent analysis; see the [Mechanical APDL Theory Reference](#).

**3.4.5. Solve**

Before you solve, save (**SAVE**) a backup copy of the database to a named file. You can then retrieve your model by reentering the program and issuing **RESUME**.

Now start the solution calculations.

**Command(s): SOLVE**

GUI: Main Menu> Solution> Solve> Current LS

### 3.4.5.1. Output

The output from the solution consists mainly of the natural frequencies, which are printed as part of the printed output (`Jobname. OUT`) and also written to the mode shape file (`Jobname. MODE`). The printed output may include mode shapes and the participation factor table, depending on your analysis options and output controls. Element results are written to the results file (`Jobname. RST`) using the **MPXAND** command. If the **MPXAND** command is not issued, the element results are not written by default and they cannot be post-processed.

### 3.4.6. Participation Factor Table Output

The participation factor table lists participation factors, mode coefficients, and mass distribution percentages for each mode extracted. The participation factors and mode coefficients are calculated based on an assumed unit displacement spectrum in each of the global Cartesian directions and rotation about each of these axes. The mass distribution is also listed. Rotational participation factors will be calculated when a real eigensolver mode-extraction method (such as Block Lanczos, PCG Lanczos, Supernode, or Subspace) is used. This table is not output when the modal analysis is for a downstream mode-superposition cyclic harmonic analysis (**CYCOPT,MSUP,1**).

**Retrieving a participation factor or mode coefficient** You can retrieve a participation factor or mode coefficient by issuing a **\*GET** command (`Entity=MODE`).

### 3.4.7. Exit the Solution Processor

You must now exit the solution processor.

**Command(s): FINISH**

GUI: Main Menu> Finish

## 3.5. Reviewing the Results

Results from a modal analysis (that is, the modal expansion pass) are written to the structural results file, `Jobname. RST`. Results consist of:

- Natural frequencies
- Expanded mode shapes
- Relative stress and force distributions (if requested).

You can review these results in POST1 (**/POST1**), the general postprocessor. Some typical postprocessing operations for a modal analysis are described below. For a complete description of all postprocessing functions, see [An Overview of Postprocessing](#) in the *Basic Analysis Guide*.

### 3.5.1. Points to Remember

- If you want to review results in POST1, the database must contain the same model for which the solution was calculated.
- The results file (`Jobname. RST`) must be available.



### 3.5.2. Reviewing Results Data

1. Read in results data from the appropriate substep. Each mode is stored on the results file as a separate substep. If you expand six modes, for instance, your results file will have one load step consisting of six substeps.

**Command(s):** SET, LSTEP, SBSTEP

**GUI:** Main Menu > General Postproc > Read Results > *By Load Step*

If the results data are complex, you can retrieve the real part, the imaginary part, the amplitude or the phase using KIMG in the SET command.

SET, LSTEP, SBSTEP , , KIMG

2. Perform any desired POST1 operations. Typical modal analysis POST1 operations are explained below:

### 3.5.3. Option: Listing All Frequencies

You may want to list the frequencies of all modes expanded. A sample output from this command is shown below.

```
***** INDEX OF DATA SETS ON RESULTS FILE *****
SET   TIME/FREQ   LOAD STEP   SUBSTEP   CUMULATIVE
  1   22.973       1           1         1
  2   40.476       1           2         2
  3   78.082       1           3         3
  4  188.34        1           4         4
```

**Command(s):** SET,LIST

**GUI:** Main Menu > General Postproc > List Results

### 3.5.4. Option: Display Deformed Shape

**Command(s):** PLDISP

**GUI:** Main Menu > General Postproc > Plot Results > Deformed Shape

Use the KUND field on PLDISP to overlay the nondeformed shape on the display.

### 3.5.5. Option: Line Element Results

**Command(s):** ETABLE

**GUI:** Main Menu > General Postproc > Element Table > Define Table

For line elements, such as beams, spars, and pipes, use the ETABLE command to access derived data (stresses, strains, and so on). Results data are identified by a combination of a label and a sequence number or component name on the ETABLE command. See the ETABLE discussion in [The General Postprocessor \(POST1\)](#) in the *Basic Analysis Guide* for details.

### 3.5.6. Option: Contour Displays

**Command(s):** PLNSOL or PLESOL

**GUI:** Main Menu > General Postproc > Plot Results > Contour Plot > Nodal Solu or Element Solu

Use these options to contour almost any result item, such as stresses (SX, SY, SZ...), strains (EPELX, EPELY, EPELZ...), and displacements (UX, UY, UZ...).

The KUND field on PLNSOL and PLESOL gives you the option of overlaying the nondeformed shape on the display.



You can also contour element table data and line element data:

**Command(s):** PLETAB, PLLS

**GUI:** Main Menu > General Postproc > Element Table > Plot Element Table

Main Menu > General Postproc > Plot Results > Contour Plot > Line Elem Res

---

### Caution

Derived data, such as stresses and strains, are averaged at the nodes by the **PLNSOL** command. This averaging results in "smeared" values at nodes where elements of different materials, different shell thicknesses, or other discontinuities meet. To avoid the smearing effect, use selecting (described in [Selecting and Components](#) in the *Basic Analysis Guide*) to select elements of the same material, same shell thickness, and so on before issuing **PLNSOL**.

---

### 3.5.7. Option: Tabular Listings

**Command(s):** PRNSOL (nodal results) PRESOL (element-by-element results) PRRSOL (reaction data), and so on NSORT, ESORT

**GUI:** Main Menu > General Postproc > List Results > *solution option*

Main Menu > General Postproc > List Results > Sorted Listing > Sort Nodes

Main Menu > General Postproc > List Results > Sorted Listing > Sort Elems

Use the **NSORT** and **ESORT** commands to sort the data before listing them.

### 3.5.8. Other Capabilities

Many other postprocessing functions - mapping results onto a path, load case combinations, and so on - are available in POST1. See [The General Postprocessor \(POST1\)](#) in the *Basic Analysis Guide* for details.

## 3.6. Applying Prestress Effects in a Modal Analysis

The following topics concerning adding prestress effect to a modal analysis are available:

3.6.1. [Performing a Prestressed Modal Analysis from a Linear Base Analysis](#)

3.6.2. [Performing a Prestressed Modal Analysis from a Large-Deflection Base Analysis](#)

### 3.6.1. Performing a Prestressed Modal Analysis from a Linear Base Analysis

Use a prestressed modal analysis to calculate the frequencies and mode shapes of a prestressed structure, such as a spinning turbine blade. This procedure is applicable only if the prior (base) analysis is a purely linear, small deflection solution.

If the prior static analysis includes large-deflection effects (**NLGEOM,ON**), use the linear perturbation procedure described in [Performing a Prestressed Modal Analysis from a Large-Deflection Base Analysis](#) (p. 50). If the base analysis includes other nonlinearities, use the linear perturbation procedure as described in [Linear Perturbation Analysis](#). (Note that the linear perturbation analysis procedure is also valid for cases where the base analysis is linear and, therefore, can be used instead of the prestressed modal analysis procedure described here.)

The procedure for performing a prestressed modal analysis from a linear base analysis is essentially the same as that of a standard modal analysis, except that you first need to prestress the structure by performing a static analysis:

1. Build the model and obtain a static solution *with prestress effects turned on* (**PSTRES,ON**). The same lumped mass setting (**LUMPM**) used here must also be used in the later prestressed modal analysis.

[Structural Static Analysis \(p. 9\)](#) describes the procedure to obtain a static solution. Use **EMAT-WRITE,YES** if you want to look at strain energies from the modal analysis.

This step can also be a transient analysis. If so, save the `EMAT` and `ESAV` files at the desired time point.

2. Enter the solution processor once again and obtain the modal solution, also with prestress effects activated (reissue **PSTRES,ON**). Files `Jobname.EMAT` (if created) and `Jobname.ESAV` from the static analysis must be available.

If another analysis is performed between the static and prestressed modal analyses, it is necessary to rerun the static analysis, or keep a copy of the `EMAT` file from the static analysis.

3. Expand the modes and review them in the postprocessor.

Keep in mind that if you specify nonlinear elements in the modal analysis, Mechanical APDL treats them as linear. For example, if you include contact elements, their stiffnesses are calculated based on their initial status and never change. For a prestressed modal analysis, the program assumes that the initial status of the contact elements is the status at the completion of the static prestress analysis.

### 3.6.2. Performing a Prestressed Modal Analysis from a Large-Deflection Base Analysis

To calculate the frequencies and mode shapes of a deformed structure or a structure involving nonlinear (sliding) contact, you can use the [linear perturbation analysis procedure](#) to perform a prestressed modal analysis following a *large-deflection* (**NLGEOM,ON**) static analysis.

To obtain the modal solution of a deformed structure, follow these steps:

1. Perform a nonlinear static solution with the prestress load. Use the **RESCONTROL** command to define the necessary restart files.
2. Restart the previous static solution from the desired load step and substep.
3. Issue the **PERTURB** command to define the analysis type, material behavior to be used, contact status (*ContKey* = CURRENT, STICKING, or BONDED) and load values to be retained from the previous static solution (*LoadControl* = ALLKEEP, INERKEEP, PARKEEP, or NOKEEP).
4. Modify the behavior of individual contact pairs, as needed, using the **CNKMOD** command.
5. Issue the **SOLVE,ELFORM** command to regenerate the matrices.
6. Issue the **MODOPT** and **MXPAND** commands to specify the modal analysis option.
7. Issue the **SOLVE** command to perform the eigensolution.
8. Postprocess the results from the `Jobname.RSTP` file.

For detailed information about the linear perturbation analysis procedure, see [Linear Perturbation Analysis](#) in the *Structural Analysis Guide*.

## 3.7. Modal Analysis Examples

The following modal analysis example topics are available:

- 3.7.1. An Example Modal Analysis (GUI Method)
- 3.7.2. An Example Modal Analysis (Command or Batch Method)
- 3.7.3. Brake Squeal (Prestressed Modal) Analysis
- 3.7.4. Reuse of Jobname.MODESYM in QRDAMP eigensolver
- 3.7.5. Calculate the Complex Mode Contribution Coefficients (CMCC)
- 3.7.6. Where to Find Other Modal Analysis Examples

### 3.7.1. An Example Modal Analysis (GUI Method)

In this example, you perform a modal analysis on the wing of a model plane to demonstrate the wing's modal degrees of freedom.

#### 3.7.1.1. Problem Description

This is a modal analysis of a wing of a model plane. The wing is of uniform configuration along its length, and its cross-sectional area is defined to be a straight line and a spline, as shown. It is held fixed to the body on one end and hangs freely at the other. The objective of the problem is to demonstrate the wing's modal degrees of freedom.

#### 3.7.1.2. Problem Specifications

The dimensions of the wing are shown in the problem sketch. The wing is made of low density polyethylene with the following values:

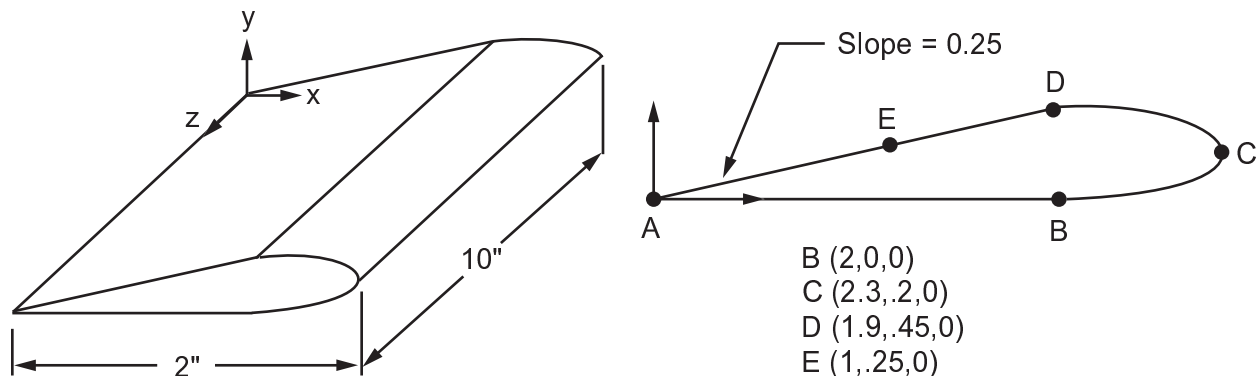
Young's modulus =  $38 \times 10^3$  psi

Poisson's ratio = .3

Density =  $8.3 \times 10^{-5}$  lb-sec<sup>2</sup>/in<sup>4</sup>

#### 3.7.1.3. Problem Sketch

Figure 3.1: Diagram of a Model Airplane Wing



The detailed step-by-step procedure for this example, Modal Analysis of a Model Airplane Wing, is included in the [Modal Tutorial](#).

### 3.7.2. An Example Modal Analysis (Command or Batch Method)

You can perform the example modal analysis of a model airplane wing using the commands shown below instead of GUI choices. Items prefaced with an exclamation point (!) are comments.

You may receive warning messages when you run this problem. The version of the problem that appears in the [Modal Tutorial](#) contains an explanation of the warnings.

```

/filename,modal

/prep7
et,1,plane182          ! define plane182 as element type 1
et,2,solid185,,3      ! define solid185 as element type 2
mp,ex,1,38000
mp,dens,1,8.3e-5
mp,nuxy,1,.3
k,1                   ! define keypoint 1 at 0,0,0
k,2,2                 ! define keypoint 2 at 2,0,0
k,3,2.3,.2           ! define keypoint 3 at 2.3,.2,0
k,4,1.9,.45          ! define keypoint 4 at 1.9,.45,0
k,5,1,.25            ! define keypoint 5 at 1,.25,0
lstr,1,2              ! create a straight line between keypoints 1 and 2
lstr,5,1              ! create a straight line between keypoints 5 and 1
bsplin,2,3,4,5,,, -1,, -1,-.25 ! create a b-spline
al,1,3,2
esize,.25
amesh,1
esize,,10
type,2
vext,all,,,,,10
/view,,1,1,1
/ang,1
/rep
eplot
finish

/solu
antype,modal          ! select modal analysis type
modopt,lanb,5         ! select the block lanczos mode-extraction method,
                     ! extracting 5 modes
esel,u,type,,1       ! unselect element type 1
nselect,s,loc,z,0
d,all,all
nselect,all
mexpand,5
solve
finish

/post1
set,list,2
set,first
*get,FREQ1,active,0,set,freq
!pdisp,0              ! uncomment to plot the displacement
!anmode,10,.5e-1     ! uncomment to animate the displacement
set,next
*get,FREQ2,active,0,set,freq
!pdisp,0              ! uncomment to plot the displacement
!anmode,10,.5e-1     ! uncomment to animate the displacement
set,next
*get,FREQ3,active,0,set,freq
!pdisp,0              ! uncomment to plot the displacement
!anmode,10,.5e-1     ! uncomment to animate the displacement
set,next
*get,FREQ4,active,0,set,freq
!pdisp,0              ! uncomment to plot the displacement
!anmode,10,.5e-1     ! uncomment to animate the displacement
set,next
*get,FREQ5,active,0,set,freq
!pdisp,0              ! uncomment to plot the displacement
!anmode,10,.5e-1     ! uncomment to animate the displacement

*stat
finish

```

### 3.7.3. Brake Squeal (Prestressed Modal) Analysis

Vehicle brakes can generate several kinds of noises. Among them is squeal, a noise in the 1-12 kHz range. It is commonly accepted that brake squeal is initiated by instability due to the friction forces, leading to self-excited vibrations.

To predict the onset of instability, you can perform a modal analysis of the prestressed structure. This type of analysis typically involves 3-D surface-to-surface contact between the brake pad and the rotating disk. The applicable contact elements are [CONTA173](#), [CONTA174](#), and [CONTA175](#).

An unsymmetric stiffness matrix is a result of the friction coupling between the brake pad and disc; this may lead to complex eigenfrequencies. If the real part of the complex frequency is positive, then the system is unstable as the vibrations grow exponentially over time.

Three different methods to perform a brake squeal analysis are presented here:

[3.7.3.1. Full Nonlinear Perturbed Modal Analysis](#)

[3.7.3.2. Partial Nonlinear Perturbed Modal Analysis](#)

[3.7.3.3. Linear Non-prestressed Modal Analysis](#)

A full nonlinear perturbed modal analysis is the most accurate method for modeling the brake squeal problem. This method uses nonlinear static solutions to both establish the initial contact and compute the sliding contact.

A partial nonlinear perturbed modal analysis is used when a nonlinear solution is required to establish contact but a linear analysis can be used to compute the sliding contact.

A linear non-prestressed modal analysis is effective when the stress-stiffening effects are not critical. This method requires less run time than the other two methods, as no nonlinear base solution is required. The contact-stiffness matrix is based on the initial contact status.

Each method involves several solution steps. The table below outlines the differences between the methods. Since the eigensolution step is the most computationally intensive step, the QRDAMP eigensolver ([MODOPT](#),[QRDAMP](#)) is generally recommended for fast turnaround time in a parametric brake squeal study environment. However, since this solver approximates the unsymmetric stiffness matrix by symmetrizing it, the unsymmetric eigensolver ([MODOPT](#),[UNSYM](#)) should be used to verify the eigenfrequencies and mode shapes.

Method	Base Static Analysis		Modal Analysis (Linear Perturbation Analysis or Linear Modal Analysis)	
	First Solve Establish initial contact status; compute prestress effects	Second Solve	First Solve	Second Solve QR damped or unsymmetric modal analysis
Full nonlinear perturbed modal analysis	Full nonlinear solution	Force frictional sliding ( <a href="#">CMRO-TATE</a> command) and perform a full nonlinear solution	Generate unsymmetric matrix	Linear perturbation modal solution

Partial perturbed modal analysis	Full nonlinear solution	N/A	Establish forced sliding contact first ( <b>CMROTATE</b> command).	Linear perturbation modal solution
Linear non-prestressed modal analysis	N/A	N/A	N/A	Force frictional sliding ( <b>CMROTATE</b> command) and perform a linear modal solution

### 3.7.3.1. Full Nonlinear Perturbed Modal Analysis

A full nonlinear perturbed modal analysis is the most accurate method to model brake squeal. The following solution steps are required:

1. Perform a nonlinear static contact analysis to establish initial contact conditions:
  - Activate large-deflection effects (**NLGEOM,ON**); (optional).
  - Use the unsymmetric stiffness matrix option (**NROPT,UNSYM**).
  - Specify the restart control points needed for the linear perturbation analysis (**RESCONTROL** command).
  - Create the element component to apply the rotational velocity (**CMROTATE** command) during the first load step.
2. Perform a forced frictional sliding contact analysis as an additional load step.

This step is needed if you want to model steady-state frictional sliding between a brake pad and the associated rotating disc (brake rotor) with different velocities. In this case, the sliding direction no longer follows the nodal displacements; instead, it is predefined through the **CMROTATE** command. This command defines the velocities on the contact and target nodes of the element component which are used to determine the sliding direction for the rest of analysis. The rotating element component (**CM** command) that is specified on the **CMROTATE** command should include only the contact elements or only the target elements that are on the brake rotor.

For isotropic friction defined via the **MP** command, if the friction coefficient is modified after the first load step (that is, a new constant value specified via **MP**) and the ramping option is set (**KBC,0**), the friction coefficient used for the friction calculation at each substep is linearly interpolated from the value of the previous load step to the value of the current load steps.

3. Perform the first phase of the linear perturbation analysis:
  - Specify a restart point (load step number and substep number) using the **ANTYPE** command (for example, **ANTYPE,STATIC,RESTART,LDSTEP,SUBSTEP,PERTURB**).
  - Specify the type of linear perturbation analysis as modal (**PERTURB,MODAL** command).
  - Issue the **RMODIF** command if you need to modify the contact stiffness values (FKN and FKT) from the base analysis. You can also issue the **CNKMOD** command to change the units of the contact stiffness from FORCE/LENGTH<sup>3</sup> (in the base static analysis) to FORCE/LENGTH.
  - Issue **SOLVE,ELFORM** to regenerate the element stiffness matrices, which are generally unsymmetric.

4. Perform a QR damped or unsymmetric modal analysis (second phase of the linear perturbation analysis):
  - Specify the QR damped or unsymmetric mode extraction method (**MODOPT,QRDAMP** or **UNSYM**).
  - Issue the **SOLVE** command.

The eigensolver uses the unsymmetric stiffness matrix generated in the contact elements, and it may lead to complex eigenfrequencies.

5. Expand the modes and postprocess the results from the Jobname .RSTP file.

The following example illustrates the full nonlinear perturbed modal analysis method for brake squeal analysis:

```

/prep7
!
! Create the brake model and apply force normal to the contact surface to
! simulate contact pressure between brake pad and disc.
!
...
!
finish
!
/solu
!
! Non-linear prestress static analysis
!
antype,static
nlgeom,on      ! include large-deflection effects (optional)
nropt,unsym    ! unsymmetric stiffness matrix
rescontrol,define ! specify restart files needed for multiframe restart
!
! Create an element component (for example, BrakeCM) consisting of brake
! rotor and contact/target element.
...
solve
!
! Pseudo rotation of disc and contact elements
! (this step generates the unsymmetric [K] in contact elements)
!
cmrotate,BrakeCM,ROTATX,ROTATY,ROTATZ
nsubst,1
solve
finish
!
/solu
antype,static,restart,,,perturb ! restart from last load step and substep
!                               ! of previous static solution to perform perperbuation analysis
perturb,modal      ! perform perturbed modal solve
solve,elform       ! regenerate element matrices
!
! Use QR damped or UNSYM eigensolver
!
modopt,qr damp
mexpand,...
solve
finish

```

### 3.7.3.2. Partial Nonlinear Perturbed Modal Analysis

When large deflection and/or stress stiffening effects play an important role in the final eigensolution, you can perform a partial prestressed modal analysis as described here.

1. Perform a static contact analysis to establish contact conditions:
  - Activate large deflection effects (**NLGEOM,ON**); optional.

- Use the unsymmetric stiffness matrix option (**NROPT,UNSYM**).
  - Specify the restart control points needed for the linear perturbation analysis (**RESCONTROL** command).
  - Create components to be used with the **CMROTATE** command.  
The initial contact condition will be established and a prestressed matrix will be generated in the end of this step under external loading.
2. Perform the first phase of the linear perturbation analysis:
    - Specify a restart point (load step number and substep number) using the **ANTYPE** command (for example, **ANTYPE,STATIC,RESTART,LDSTEP,SUBSTEP,PERTURB**).
    - Specify the type of linear perturbation analysis as modal (**PERTURB,MODAL** command).
    - Issue the **CMROTATE** command. The contact stiffness matrix is based only on the contact status at the restart point. The sliding direction no longer follows the nodal displacements; instead, it is defined through the **CMROTATE** command.
    - Issue the **RMODIF** command if you need to modify the contact stiffness values (FKN and FKT) from the base analysis. You can also issue the **CNKMOD** command to change the units of the contact stiffness from FORCE/LENGTH<sup>3</sup> (in the base static analysis) to FORCE/LENGTH.
    - Issue **SOLVE,ELFORM** to regenerate the element stiffness matrices, which are generally unsymmetric.
  3. Perform a QR damped or unsymmetric modal analysis (second phase of the linear perturbation analysis):
    - Specify the QR damped or unsymmetric mode extraction method (**MODOPT,QRDAMP** or **UNSYM**).
    - If you are planning to perform multiple solutions and want to reuse the Block Lanczos eigensolution from the previous load steps, issue **QRDOPT,ON** (QRDAMP only).
    - Issue the **SOLVE** command.
  4. Expand the modes and postprocess the results from `Jobname.RSTP`.
  5. To perform friction sensitivity studies by reusing the normal modes from the Block Lanczos solution (QRDAMP only), repeat steps 2, 3, and 4 by changing the coefficient of friction at step 2 and reissuing the **QRDOPT,ON** command at step 3.

The difference between this procedure and the procedure described in [Full Nonlinear Perturbed Modal Analysis \(p. 54\)](#) is that here the **CMROTATE** command is used in the first phase of the linear perturbation analysis, while by the other procedure the **CMROTATE** command is invoked in the nonlinear base analysis. The procedure described here is computationally less expensive compared to the [Full Nonlinear Perturbed Modal Analysis \(p. 54\)](#).

The following example illustrates the partial nonlinear perturbed modal analysis method for brake squeal analysis:

```
!
! Create the brake model and apply force normal to contact surface to
! simulate contact pressure between brake pad and disc.
!
...
!
finish
!
/solu
```



```

!
! Non-linear prestress static analysis
!
antype,static
nlgeom,on      ! include large-deflection effects (optional)
nropt,unsym    ! unsymmetric stiffness matrix
rescontrol,define ! specify restart files for multifrme restart
solve
finish
!
/solu
antype,static,restart,,,perturb ! restart from last load steop and substep
                                ! of previous static solution to perform
                                ! perturbation analysis
perturb,modal      ! perform perturbed modal analysis
!
! Pseudo rotation of disc and contact elements
! (this step generates the unsymmetric [K] in contact elements)
!
! Create an element component (for example, BrakeCM) consisting of brake
! rotor and contact/target element.
...
!
cmrotate,BrakeCM,ROTATX,ROTATY,ROTATZ
solve,elform ! regenerate the element matrices
!
! Use QR damped or UNSYM eigensolver
!
qrdopt,on          ! Generate Jobname.MODESYM
modopt,qr damp,... ! (or) modopt,unsym
mxpand,...
solve
finish
!
/clear,nostart
!
/solu
antype,static,restart,,,perturb ! restart from last load step and substep
                                ! of previous static solution to perform perperbuation analysis
mp,mu,cid,0.3 ! Change coefficient of friction
perturb,modal ! perform perturbed modal solve
solve,elform ! regenerate element matrices
!
! Use QR damped or UNSYM eigensolver
!
qrdopt,on ! Reuse Jobname.MODESYM
modopt,qr damp
mxpand,...
solve
finish

```

For further information, see [Linear Perturbation Analysis](#) (p. 287).

### 3.7.3.3. Linear Non-prestressed Modal Analysis

The [full nonlinear perturbed modal analysis method](#) described above requires a nonlinear static stress analysis. If large-deflection or stress-stiffening effects are not critical, you can use the linear non-prestressed modal analysis method instead. This method involves a single linear QR damped or unsymmetric eigensolution. Since it is a linear analysis, this method is less time consuming. The following steps are required:

1. Perform a QR damped (QRDAMP) or unsymmetric (UNSYM) modal analysis using the linear modal analysis procedure:
  - Use the unsymmetric stiffness matrix option (**NROPT,UNSYM**) to generate the unsymmetric stiffness matrix. (No Newton-Raphson iterations are performed.)

- Generate frictional sliding force via the **CMROTATE** command. The contact stiffness matrix is based only on the initial contact status.
- Specify the QR damped or unsymmetric mode extraction method (**MODOPT,QRDAMP** or **UNSYM**).
- If you are planning to perform multiple solutions and want to reuse the Block Lanczos eigensolution from the previous load steps, issue **QRDOPT,ON** (QRDAMP only).
- Issue the **SOLVE** command.
- To perform friction sensitivity studies by reusing the normal modes from the Block Lanczos solution (QRDAMP only), just change the friction coefficients and issue the **SOLVE** command.

2. Expand the modes and postprocess the results from `Jobname.RST`.

The following example illustrates the linear non-prestressed modal analysis method for brake squeal analysis:

```

/prep7
!
! Create the brake model
!
finish
!
/solu
!
! Perform a linear modal analysis
!
antype,modal
nropt,unsym      ! unsymmetric stiffness matrix
!
! Create an element component (for example, BrakeCM) consisting of brake
! rotor and contact/target elements.
!
...
cmrotate,BrakeCM,ROTATX,ROTATY,ROTATZ
!
! Use QR damped or UNSYM eigensolver
!
qrdopt,on          ! Generate Jobname.MODESYM
modopt,qr damp,... ! (or) modopt,unsym
mxpand,...
solve
!
mp,mu,cid,0.3      ! Change freiction coefficient
solve
finish

```

### 3.7.4. Reuse of `Jobname.MODESYM` in QRDAMP eigensolver

This section describes the reuse of `Jobname.MODESYM` in QRDAMP eigensolver. Reuse is activated by the reuse flag (*ReuseKey*) in the **QRDOPT** command. A typical example of the reuse of `Jobname.MODESYM` would be for determining instability predictions in brake squeal analysis ([Brake Squeal \(Prestressed Modal\) Analysis](#) (p. 53)).

To reuse `Jobname.MODESYM` in QRDAMP eigensolver, turn on the *ReuseKey* reuse flag in the **QRDOPT** command. The following solution steps occur:

1. In the first load step, if `Jobname.MODESYM` does not exist, the block Lanczos eigensolution is performed to create the `Jobname.MODESYM` mode file. This file contains symmetric normal modes of the eigenproblem, given by:

$$[K]\{x\} = \lambda[M]\{x\}$$

Where the eigenvalues  $\lambda$  and  $\{x\}$  are output as symmetric modes to `Jobname.MODESYM`. The  $[K]$  matrix in the above expression is the stiffness matrix, symmetrized as outlined in [QR Damped Method in the Mechanical APDL Theory Reference](#).

- Then in the first and all the subsequent load steps, the symmetric Lanczos eigenmodes are used to build the subspace eigenproblem and create the `Jobname.MODE`. This mode file contains the complex eigenmodes of the non-symmetric eigenproblem, given by:

$$[K_u]\{X_u\} = \lambda_u[M]\{X_u\}$$

Where  $\lambda_u$  and  $\{x_u\}$  are output as unsymmetric modes to `Jobname.MODE`.

- The Lanczos eigenmodes from the symmetrized eigenproblem written out to the `Jobname.MODESYM` mode file are available for later use by the QRDAMP eigensolver.

When analyzing for brake squeal or Campbell plot generation in rotordynamics, the reuse approach can improve performance by avoiding the Block Lanczos run that typically occurs in a QRDAMP eigensolution. Exercise caution when comparing eigensolutions from a reuse run with a non-reuse run, as symmetrization differs in these runs. In a non-reuse run the  $[K]$  matrix gets symmetrized at each load step of a QR damp eigenanalysis. In a reuse run the symmetrization occurs at the first load step and the symmetric normal modes are reused in all subsequent load steps.

The QRDAMP eigensolver will attempt to reuse the normal modes from `Jobname.MODESYM` if the file is present in the folder where the job is run, so ensure that `Jobname.MODESYM` is created by the same model as the subsequent QR damp model.

### 3.7.5. Calculate the Complex Mode Contribution Coefficients (CMCC)

In a brake squeal analysis, the complex mode contribution coefficients (CMCC) can be used to determine how much the symmetric normal modes contribute to the complex modes. They can be directly output from the solver to a text file using the `CMCCOutKey` on the **QRDOPT** command. The CMCC are the components of vector  $\{y\}$  defined in [Equation 14.210 in the Mechanical APDL Theory Reference](#). The equations to recalculate the CMCC are described in this section. Using the upper part of [Equation 14.212 in the Mechanical APDL Theory Reference](#), the relationship between the real and the complex modes can be written as:

$$\{\underline{\psi}\} = [\underline{\phi}]\{y\}$$

Where:

$\{\underline{\psi}\}$  is the vector of complex modes (upper part only) in:

$$\{\psi\} = \begin{Bmatrix} \{\underline{\psi}\} \\ \{\underline{\dot{\psi}}\} \end{Bmatrix}$$

$[\underline{\phi}]$  is the matrix of real modes (upper part only) in:

$$[\phi] = \begin{bmatrix} [\underline{\phi}] \\ [\underline{\dot{\phi}}] \end{bmatrix}$$

Premultiplying by  $[\underline{\phi}]^T [M]$  when real modes are mass normalized obtains:

$$\{y\} = [\underline{\phi}]^T [M] \{\psi\} \quad (3.1)$$

### 3.7.6. Where to Find Other Modal Analysis Examples

Several ANSYS, Inc. publications, particularly the *Mechanical APDL Verification Manual*, describe additional modal analyses.

The *Mechanical APDL Verification Manual* consists of test case analyses demonstrating the analysis capabilities of the ANSYS family of products. While these test cases demonstrate solutions to realistic analysis problems, the *Mechanical APDL Verification Manual* does not present them as step-by-step examples with lengthy data input instructions and printouts. However, most ANSYS users who have at least limited finite element experience should be able to fill in the missing details by reviewing each test case's finite element model and input data with accompanying comments.

The *Mechanical APDL Verification Manual* includes variety of modal analysis test cases:

- VM45 - Natural Frequency of a Spring-mass System
- VM47 - Torsional Frequency of a Suspended Disk
- VM48 - Natural Frequency of a Motor-generator
- VM50 - Fundamental Frequency of a Simply Supported Beam
- VM52 - Automobile Suspension System Vibrations
- VM53 - Vibration of a String Under Tension
- VM54 - Vibration of a Rotating Cantilever Blade
- VM55 - Vibration of a Stretched Circular Membrane
- VM57 - Torsional Frequencies of a Drill Pipe
- VM59 - Lateral Vibration of an Axially-loaded Bar
- VM60 - Natural Frequency of a Cross-ply Laminated Spherical Shell
- VM61 - Longitudinal Vibration of a Free-free Rod
- VM62 - Vibration of a Wedge
- VM66 - Vibration of a Flat Plate
- VM67 - Radial Vibrations of a Circular Ring from an Axisymmetric Model
- VM68 - PSD Response of a Two-degrees of freedom Spring-mass System
- VM69 - Seismic Response
- VM70 - Seismic Response of a Beam Structure
- VM76 - Harmonic Response of a Guitar String
- VM89 - Natural Frequencies of a Two-mass-spring System
- VM151 - Nonaxisymmetric Vibration of a Circular Plate
- VM152 - Nonaxisymmetric Vibration of a Stretched Circular Membrane (Harmonic Els)
- VM153 - Nonaxisymmetric Vibration of a Stretched Circular Membrane (Modal)
- VM154 - Vibration of a Fluid Coupling
- VM175 - Natural Frequency of a Piezoelectric Transducer
- VM181 - Natural Frequency of a Flat Circular Plate with a Clamped Edge
- VM182 - Transient Response of a Spring-mass System
- VM183 - Harmonic Response of a Spring-mass System
- VM202 - Transverse Vibrations of a Shear Beam
- VM203 - Dynamic Load Effect on Simply-supported Thick Square Plate
- VM212 - Modal Analysis of a Rectangular Cavity

## 3.8. Comparing Mode-Extraction Methods

The basic equation solved in a typical undamped modal analysis is the classical eigenvalue problem:

$$[K]\{\phi_i\} = \omega_i^2 [M]\{\phi_i\}$$

where:

[K] = stiffness matrix

{ $\Phi_i$ } = mode shape vector (eigenvector) of mode i

$\Omega_i$  = natural circular frequency of mode i ( $\omega_i^2$  is the eigenvalue)

[M] = mass matrix

Many numerical methods are available to solve the equation. Mechanical APDL offers these methods:

3.8.1. Block Lanczos Method

3.8.2. PCG Lanczos Method

3.8.3. Supernode (SNODE) Method

3.8.4. Subspace Method

3.8.5. Unsymmetric Method

3.8.6. Damped Method

3.8.7. QR Damped Method

3.8.8. Storage of Complex Results

The damped and QR damped methods solve *different* equations. For more information, see [Damped Method](#) and [QR Damped Method](#) in the *Mechanical APDL Theory Reference*.

The Block Lanczos, PCG Lanczos, and Supernode mode-extraction methods are the most commonly used:

**Table 3.5: Symmetric System Eigensolver Options**

Eigensolver	Application	Memory Re-quired	Disk Re-quired
<a href="#">Block Lanczos</a>	To find many modes (about 40+) of large models. Recommended when the model consists of poorly shaped solid and shell elements. This solver performs well when the model consists of shells or a combination of shells and solids.	Medium	High
<a href="#">PCG Lanczos</a>	To find few modes (up to about 100) of very large models (500,000+ degrees of freedom). This solver performs well when the lowest modes are sought for models that are dominated by well-shaped 3-D solid elements (that is, models that would typically be good candidates for the PCG iterative solver for a similar static or full transient analysis).	Medium	Low
<a href="#">Supernode</a>	To find many modes (up to 10,000) efficiently. Use this method for 2-D plane or shell/beam structures (100 modes or more) and for 3-D solid structures (250 modes or more).	Medium	Low

Eigsolver	Application	Memory Re-quired	Disk Re-quired
Subspace	To find many modes (about 40+) of large models. This solver is more robust when the mass matrix is partially zero, or when there are u-P formulation elements in the model. It is recommended when using Distributed ANSYS.	Medium	High

The PCG Lanczos, Subspace, unsymmetric, and damped methods are the only eigenvalue solvers that will run a fully distributed solution in Distributed ANSYS.

### 3.8.1. Block Lanczos Method

The Block Lanczos eigenvalue solver uses the Lanczos algorithm where the Lanczos recursion is performed with a block of vectors. The Block Lanczos method uses the sparse matrix solver, overriding any solver specified via the **EQSLV** command.

The Block Lanczos method is especially powerful when searching for eigenfrequencies in a given part of the eigenvalue spectrum of a given system. The convergence rate of the eigenfrequencies will be about the same when extracting modes in the midrange and higher end of the spectrum as when extracting the lowest modes. Therefore, when you use a shift frequency (*FREQB* on **MODOPT**) to extract *n* modes beyond the starting value of *FREQB*, the algorithm extracts the *n* modes beyond *FREQB* at about the same speed as it extracts the lowest *n* modes.

While the Block Lanczos eigensolver is typically very robust for a wide range of applications, some element types may cause the eigensolver difficulty in achieving a final solution. These include **MPC184** elements that use the Lagrange multiplier method and **FLUID30**, **FLUID220**, and **FLUID221** elements that use the symmetric element matrix formulation for modal analyses (KEYOPT(2) = 2).

### 3.8.2. PCG Lanczos Method

The PCG Lanczos method internally uses the Lanczos algorithm, combined with the PCG iterative solver. This method will be significantly faster than the Block Lanczos method for the following cases:

- Large models that are dominated by 3-D solid elements and do not have ill-conditioned matrices due, for example, to poorly shaped elements
- Only a few of the lowest modes are requested

Having ill-conditioned matrices or asking for many modes (e.g., more than 100 modes) can lead to an inefficient solution time with this method.

The PCG Lanczos method finds only the lowest eigenvalues. If a range of eigenfrequencies is requested on the **MODOPT** command, the PCG Lanczos method will find all of the eigenfrequencies below the lower value of the eigenfrequency range as well as the number of requested eigenfrequencies in the given eigenfrequency range. Thus the PCG Lanczos method is not recommended for problems when the lower value of the input eigenfrequency range is far from zero.

#### When to Choose PCG Lanczos over Block Lanczos

The Block Lanczos eigensolver is currently the recommended eigensolver for most applications. However, the PCG Lanczos eigensolver can be more efficient than the Block Lanczos eigensolver for certain cases. Follow these guidelines when deciding whether to use the Block Lanczos or PCG Lanczos eigensolver.

Typically, the following conditions must be met for the PCG Lanczos eigensolver to be most efficient:

1. The model would be a good candidate for using the PCG solver in a similar static or full transient analysis.
2. The number of requested modes is less than a hundred.
3. The beginning frequency input on the **MODOPT** command is zero (or near zero).

The PCG Lanczos eigensolver (like all iterative solvers) is most efficient when the solution converges quickly. So if the model would not converge quickly in a similar static or full transient analysis, it is expected that the PCG Lanczos eigensolver will also not converge quickly and thus be less efficient. The PCG Lanczos eigensolver is most efficient when finding the lowest natural frequencies of the system. (For detailed information on measuring PCG Lanczos solver efficiency, see [PCG Lanczos Solver Performance Output](#) in the *Performance Guide*.) As the number of requested modes begins to approach one hundred, or when requesting only higher modes, the Block Lanczos eigensolver becomes a better choice.

Other factors such as the size of the problem and the hardware being used can affect the eigensolver selection strategy. For example, when solving problems where the Block Lanczos eigensolver runs in an out-of-core mode on a system with very slow hard drive speed (that is, bad I/O performance) the PCG Lanczos eigensolver may be the better choice as it does significantly less I/O, assuming a *Lev\_Diff* value of 1 through 4 is used (**PCGOPT** command). Another example would be solving a model with 15 million degrees of freedom. In this case, the Block Lanczos eigensolver would need approximately 300 GB of hard drive space. If computer resources are limited, the PCG Lanczos eigensolver may be the only viable choice for solving this problem since the PCG Lanczos eigensolver does much less I/O than the Block Lanczos eigensolver. The PCG Lanczos eigensolver has been successfully used on problems exceeding 100 million degrees of freedom.

### 3.8.3. Supernode (SNODE) Method

The Supernode (SNODE) solver is used to solve large, symmetric eigenvalue problems for many modes (up to 10,000 and beyond) in one solution. Typically, the reason for seeking many modes is to perform a subsequent mode-superposition or PSD analysis to solve for the response in a higher frequency range.

A supernode is a group of nodes from a group of elements. The supernodes for the model are generated automatically by the program. This method first calculates eigenmodes for each supernode in the range of 0.0 to  $FREQE * RangeFact$  (where *RangeFact* is specified by the **SNOPTION** command and defaults to 2.0), and then uses the supernode eigenmodes to calculate the global eigenmodes of the model in the range of *FREQB* to *FREQE* (where *FREQB* and *FREQE* are specified by the **MODOPT** command).

Typically, this method offers faster solution times than Block Lanczos or PCG Lanczos if the number of modes requested is more than 200. The accuracy of the Supernode solution can be controlled by the **SNOPTION** command. For more information, see [Supernode Method](#) in the *Mechanical APDL Theory Reference*.

The lumped mass matrix option (**LUMPM,ON**) is not allowed when using the Supernode mode-extraction method. The consistent mass matrix option will be used regardless of the **LUMPM** setting.

#### When to Choose Supernode

Typically, the following conditions must be met for the Supernode eigensolver to be most efficient:

1. The model is a good candidate for using the sparse solver in a similar static or full transient analysis (that is, the model is dominated with beams/shells or has a thin structure).



2. The number of requested modes is greater than 200.
3. The beginning frequency input on the **MODOPT** command is zero (or near zero).

For models that are dominated by solid elements or have a bulkier geometry, the Supernode eigensolver can still be more efficient than other eigensolvers; however, it may require higher numbers of modes to become the best choice in terms of performance. Other factors, such as the hardware being used, can also affect the decision of which eigensolver to use. For example, on machines with slow I/O performance the Supernode eigensolver may be the faster eigensolver (compared to Block Lanczos).

### 3.8.4. Subspace Method

The Subspace method is an iterative algorithm that is appropriate for problems having symmetric stiffness and mass matrices ([K] and [M]). It internally uses the sparse solver or the distributed sparse solver (depending on whether or not the solution uses Distributed ANSYS) for the shift-invert logic. The Subspace solver performs well when the goal is to obtain a moderate to medium number of eigenvalues on large models run in distributed parallel mode.

Sturm sequence checking is available for this method (it is off by default) and can be controlled via the **SUBOPT** command. The memory usage of the algorithm can also be configured via the **SUBOPT** command.

### 3.8.5. Unsymmetric Method

The *unsymmetric* method, which also uses the full [K] and [M] matrices, is meant for problems where the stiffness and mass matrices are unsymmetric (for example, acoustic fluid-structure interaction problems). The real part of the eigenvalue represents the natural frequency and the imaginary part is a measure of the stability of the system - a negative value means the system is stable, whereas a positive value means the system is unstable. Sturm sequence checking is not available for this method. Therefore, missed modes are a possibility at the higher end of the frequencies extracted.

### 3.8.6. Damped Method

The *damped* method (**MODOPT,DAMP**) is meant for problems where damping cannot be ignored, such as rotor dynamics applications. It uses full matrices ([K], [M], and the damping matrix [C]). Sturm sequence checking is not available for this method. Therefore, missed modes are a possibility at the higher end of the frequencies extracted.

#### 3.8.6.1. Damped Method--Real and Imaginary Parts of the Eigenvalue

The imaginary part of the eigenvalue,  $\Omega$ , represents the steady-state circular frequency of the system. The real part of the eigenvalue,  $\sigma$ , represents the stability of the system. If  $\sigma$  is less than zero, then the displacement amplitude will decay exponentially, in accordance with  $\text{EXP}(\sigma)$ . If  $\sigma$  is greater than zero, then the amplitude will increase exponentially. (Or, in other words, negative  $\sigma$  gives an exponentially decreasing, or stable, response; and positive  $\sigma$  gives an exponentially increasing, or unstable, response.) If there is no damping, the real component of the eigenvalue will be zero.

The eigenvalue results reported are actually divided by  $(2^* \pi)$ , giving the frequency in Hz (cycles/second). In other words:

Imaginary part of eigenvalue, as reported =  $\Omega / (2^* \pi)$

Real part of eigenvalue, as reported =  $\sigma / (2^* \pi)$



### 3.8.6.2. Damped Method-Real and Imaginary Parts of the Eigenvector

In a damped system, the response at different nodes can be out of phase. At any given node, the amplitude will be the vector sum of the real and imaginary components of the eigenvector.

### 3.8.7. QR Damped Method

The *QR damped* method (**MODOPT**,QRDAMP) combines the advantages of the Block Lanczos method with the complex Hessenberg method. The key concept is to approximately represent the first few complex damped eigenvalues by modal transformation using a small number of eigenvectors of the undamped system. After the undamped mode shapes are evaluated by using the real eigensolution (Block Lanczos method), the equations of motion are transformed to these modal coordinates.

Using the QR algorithm, a smaller eigenvalue problem is then solved in the modal subspace. This approach gives good results for lightly damped systems and can also apply to any arbitrary damping type (proportional or non-proportional symmetric damping or nonsymmetrical gyroscopic damping matrix). This approach also supports nonsymmetrical stiffness if present in the model.

The QRDAMP eigensolver applies to models having an unsymmetrical global stiffness matrix where only a few elements contribute nonsymmetrical element stiffness matrices. For example, in a brake-friction problem, the local part of a model with friction contacts generates a nonsymmetrical stiffness matrix in contact elements. When a non-symmetric stiffness matrix is encountered the eigenfrequencies and mode shapes obtained by the QRDAMP eigensolver must be verified by rerunning the analysis with the non-symmetric eigensolver. If a non-symmetric stiffness matrix is encountered a warning message cautioning the user is output by the QRDAMP eigensolver right after the completion of the Block Lanczos eigensolution.

The QRDAMP eigensolver works best when there is a larger “modal subspace” to converge and is therefore the best option for larger models. Because the accuracy of this method is dependent on the number of modes used in the calculations, a sufficient number of fundamental modes should be present (especially for highly damped systems) to provide good results. The QR damped method is not recommended for critically damped or overdamped systems. This method outputs both the real and imaginary eigenvalues (frequencies), but outputs only the real eigenvectors (mode shapes). When requested however, complex mode shapes of damped systems are computed.

In general, ANSYS, Inc. recommends using the **Damp** eigensolver for small models. It produces more accurate eigensolutions than the QR Damped eigensolver for damped systems.

### 3.8.8. Storage of Complex Results

While the UNSYM, DAMP, and QRDAMP methods all generate complex eigensolutions, there is a difference in the way eigenresults are stored in the result files, particularly the eigenvalues.

Eigensolver	Eigenvalue	Eigenvector
UNSYM	<p><b>Real Part:</b> natural frequency  <b>Imaginary Part:</b> measure of the stability of the system</p> <p>A negative imaginary part means the system is stable, whereas a positive value means the system is unstable</p>	<p><b>Real Part:</b> Mode shape associated to the Real Part of the eigenvalue. (Natural frequency)  <b>Imaginary Part:</b> Mode shape associated to the imaginary part of</p>

Eigensolver	Eigenvalue	Eigenvector
		<p>the eigenvalue. (stability)</p> <p>If the eigenvalue is real, the corresponding modal shape is real (the imaginary slot will be zero).</p>
<p>DAMP - QRDAMP</p>	<p><b>Real Part (<math>\sigma</math>):</b> measure of the stability of the system  <b>Imaginary Part (<math>\Omega</math>):</b> steady-state circular frequency of the system</p> <p>A negative <math>\sigma</math> gives an exponentially decreasing, or stable, response.</p> <p>A positive <math>\sigma</math> gives an exponentially increasing, or unstable, response.</p>	<p><b>Real Part:</b> Mode shape associated to the Imaginary Part of the eigenvalue. (Natural frequency)  <b>Imaginary Part:</b> Mode shape associated to the real part of the eigenvalue. (Stability)</p>

### 3.9. Modal Analysis Tools for Subsequent Mode Superposition Analysis

The following modal analysis tools are available for subsequent mode superposition analyses:

- 3.9.1. Using the Residual-Vector Method to Improve Accuracy
- 3.9.2. Reusing Eigenmodes
- 3.9.3. Generating and Using Multiple Loads in Mode-Superposition Analyses
- 3.9.4. Restarting a Modal Analysis
- 3.9.5. Enforced Motion Method for Mode-Superposition Transient and Harmonic Analyses
- 3.9.6. Using Mode Selection

#### 3.9.1. Using the Residual-Vector Method to Improve Accuracy

The residual vector method improves the accuracy of a mode-superposition transient, mode-superposition harmonic, PSD, or spectrum analysis.

A mode-superposition solution tends to be less accurate when the applied dynamic loads excite the higher resonant frequency modes of a structure. Many modes are often necessary to render an accurate mode-superposition solution.

The residual vector method can help in such cases. The method's improved convergence properties require fewer natural frequencies and modes from the eigensolution.

##### 3.9.1.1. Understanding the Residual Vector Method

To use the residual vector method, you must first calculate residual vectors in the modal analysis. You can use either of these modal analysis mode-extraction methods:

- Block Lanczos (**MODOPT**,LANB)
- PCG Lanczos (**MODOPT**,LANPCG)

- SNODE method (**MODOPT**,SNODE)
- SUBSP method (**MODOPT**,SUBSP)

Mechanical APDL stores the calculated residual vectors in the `Jobname.MODE` file and uses them in the subsequent mode-based analysis (e.g., mode superposition harmonic, mode superposition transient, spectrum, or random vibration (PSD) analysis).

When using the residual vector method, set `FREQB` on the **MODOPT** command to zero or blank. Residual vectors can not characterize the high frequency range if significant modes in the low frequency range are not extracted.

Because the residual vectors are calculated before the modes are expanded, they are based on all the extracted modes. As a result, a partial expansion driven by the **MXPAND** command will not affect the residual vectors. If you do not expand the residual vectors, they will not be available on the results file for post-processing.

### 3.9.1.2. Using the Residual Vector Method

Use the following procedure to calculate residual vectors:

1. Build the model.
2. Specify the mode-extraction method (**MODOPT**,LANB or **MODOPT**,LANPCG, **MODOPT**,SNODE, or **MODOPT**,SUBSP).
3. Activate residual vector calculation (**RESVEC**,ON).
4. Specify pseudo-constraints (**D**,,,SUPPORT) if rigid body motion is present.
5. Specify the load vectors (**F**, **BF**, **SF**, etc.).
6. Solve the modal analysis. (Mechanical APDL generates a `Jobname.MODE` file containing the residual vectors.)
7. Issue a **FINISH** command.

The residual vector calculation can be performed during a modal analysis restart. Refer to [Modal Analysis Restart in the Mechanical APDL Basic Analysis Guide](#) for more details.

8. Set up a [mode-superposition transient](#), [mode-superposition harmonic](#), [PSD](#), or [spectrum](#) analysis, and include the previously calculated residual vectors (**RESVEC**,ON).

A load vector is also generated in Step 6. Ensure that you do not duplicate any loading.

9. Solve the mode-superposition analysis. Mechanical APDL includes the residual vectors in those calculations.

#### Specifying Pseudo-Constraints

If rigid body motion exists, specify only the minimum number of displacement constraints necessary to prevent rigid body motion: three constraints (or fewer, depending on the element type) for 2-D models and six (or fewer) for 3-D models.

## 3.9.2. Reusing Eigenmodes

Mechanical APDL analyses that require the eigenmodes from the modal analysis can reuse the modes from an earlier modal analysis solution. The user can reuse the `Jobname.MODE` (or `Jobname.MODESYM`) file that is created in a modal analysis for use in the following modal based methods:

- Spectrum Analysis (**ANTYPE,SPECTRUM**) (p. 68)
- Modal Transient Analysis/Harmonic Analysis (p. 68)
- QR Damp Complex Modes Extraction (p. 68)

This section outlines the procedures for saving and reusing the eigenmodes from an earlier modal analysis.

### 3.9.2.1. Spectrum Analysis (**ANTYPE,SPECTRUM**)

To run a spectrum analysis, users must first perform a modal analysis to generate the MODE file. For multiple spectrum analyses, a unique MODE file can be used when the `modeReuseKey` on the **SPOPT** command is activated. This prepares the database and the necessary files for a new spectrum analysis that reuses an existing `Jobname.MODE`.

### 3.9.2.2. Modal Transient Analysis/Harmonic Analysis

To use new load vectors, residual vector, and/or enforced motion in modal transient or modal harmonic analyses with existing modal analysis results, refer to [Restarting a Modal Analysis \(p. 69\)](#)

### 3.9.2.3. QR Damp Complex Modes Extraction

In QRDAMP eigensolver the solution occurs in two steps. First the Block Lanczos eigensolver is used to extract the symmetric matrix eigenmodes. These eigenmodes then are used in the second pass to build the modal subspace (see [QR Damped Method](#)) matrix of the non-symmetric eigensystem and compute the complex eigenmodes.

When an existing `Jobname.MODESYM` containing the eigenmodes of the symmetric eigensolution of the model is available, it can be reused in the second pass by turning on the reuse flag (`ReuseKey`) in the **QRDOPT** command.

## 3.9.3. Generating and Using Multiple Loads in Mode-Superposition Analyses

In a mode-superposition harmonic, transient, or PSD analysis, only nodal forces can be defined directly during the analysis. The other loads (**SF, SFE, BF, ACEL, OMEGA**, etc.) must be defined in the modal analysis. They can then be retrieved, scaled, and applied using the **LVSCALE** command in the mode-superposition analysis. Spring preloads, initial strains, pretension loads, and thermal loads are ignored and do not generate a load.

Nodal forces can also be applied more efficiently using this multiple loads generation method.

### 3.9.3.1. Understanding the Multiple Loads Method

To use the multiple loads method, you must first generate load vectors in the modal analysis. You can use any of these modal analysis mode-extraction methods:

- Block Lanczos (**MODOPT,LANB**)

- PCG Lanczos (**MODOPT**,LANPCG)
- Supernode method (**MODOPT**,SNODE)
- Subspace method (**MODOPT**,SUBSP)
- Unsymmetric method (**MODOPT**,UNSYM)
- QR damped method with no complex solutions (**MODOPT**,QRDAMP,,,,OFF)

Mechanical APDL stores the generated load vectors in the `Jobname.MODE` file so that they can be used in the subsequent mode-superposition analysis.

### 3.9.3.2. Using the Multiple Loads Method

The procedure for generating multiple load vectors is as follows:

1. Build the model.
2. Specify the mode-extraction method (**MODOPT**,LANB; LANPCG; SNODE; SUBSP; UNSYM; or QRDAMP).
3. Activate multiple load vectors generation (**MODCONT**,ON). Use the **THEXPAND** command if you wish to ignore thermal strains in the generated load vector.
4. Specify the first set of loads (**F**, **BF**, **SF**, etc.).
5. Issue **SOLVE**. This first solution performs the modal extraction and generates the first load vector.
6. Delete the first set of loads and apply the second set of loads.
7. Issue **SOLVE**. This solution generates the second load vector.
8. Repeat step 6 and 7 for any additional load vectors.
9. Issue a **FINISH** command.

The load vectors calculation can be performed during a modal analysis restart. Refer to [Modal Analysis Restart in the Mechanical APDL Basic Analysis Guide](#) for more details.

The load vectors can then be used in a mode-superposition analysis as follows:

1. Set up a [mode-superposition transient](#), [harmonic](#), or [PSD](#) analysis. Ensure that any loads used in the creation of the load vector in the modal analysis are deleted; otherwise, duplicate loads may result.
2. Scale the load vectors by using **LVSCALE** command. All load vectors must be scaled before the first **SOLVE**. Use a zero scale factor if load vectors are not actually used in this first load step.
3. Solve the mode-superposition analysis. Ensure that the load vectors that are not used at a particular load step have a zero scale factor on the **LVSCALE** command.

### 3.9.4. Restarting a Modal Analysis

Reusing eigenmodes that have already been generated can save significant time in an analysis. Modal extraction typically requires more time than element loads generation, residual vector calculation, and enforced static modes calculation. The procedure for restarting the modal analysis to enrich the `Jobname.MODE` file is described in [Modal Analysis Restart](#).

### 3.9.5. Enforced Motion Method for Mode-Superposition Transient and Harmonic Analyses

In a mode superposition transient or harmonic analysis, the enforced motion method can be used when acceleration or displacement base excitation is present.

#### 3.9.5.1. Understanding the Enforced Motion Method

In a modal analysis, Mechanical APDL calculates pseudo-static modes and writes them to the `Jobname.MODE` file. They can then be used in a subsequent mode-superposition analysis with base excitation.

#### 3.9.5.2. Using the Enforced Motion Method

The procedure for calculating pseudo-static modes in a modal analysis is as follows:

1. Specify modal analysis (**ANTYPE**, MODAL).
2. Select one of the mode-extraction methods:
  - Block Lanczos (**MODOPT**,LANB)
  - SNOBE method (**MODOPT**,SNOBE)
3. Turn on the enforced motion calculation key by using the **MODCONT** command (*EnforcedKey=ON*).
4. Using the **D** command, specify the support points where imposed motion will be applied (node and degree of freedom label), as well as the enforced base identification number. This number is input as *VALUE*, which must be an integer.
5. Issue a **SOLVE** command.
6. Issue a **FINISH** command.

---

#### Note

The enforced motion pseudo-static modes calculation can be performed during a modal analysis restart. Refer to [Modal Analysis Restart in the Mechanical APDL Basic Analysis Guide](#) for more details.

---

To use the pseudo-static modes in the mode-superposition transient/harmonic analysis, issue the **DVAL** command.

#### 3.9.5.3. Sample Input for Enforced Motion Mode-Superposition Analysis

A sample input listing for a mode-superposition transient analysis follows:

```
! Build the Model
/FILNAM,... ! Jobname
/TITLE,... ! Title
/PREP7
---
--- ! Generate model
---
NSEL,...
CM, BASE1,NODE ! Define component BASE1 with selected support nodes undergoing the first enforced motion
```

```

NSEL,ALL
NSEL,...
CM, BASE2,NODE ! Define component BASE2 with selected support nodes undergoing the second enforced motion
FINISH

! Generate the pseudo-static modes in Modal Analysis
/SOLU
ANTYPE, MODAL
MODOPT, LANB,10
MODCONT,, ON ! Activate enforced motion

--- ! Define constraints

CMSEL,S,BASE1,NODE ! Select nodes of component BASE1
D, ALL, UX, 1 ! Define enforced motion 1 to be applied on all nodes along UX
CMSEL,S,BASE2,NODE ! Select nodes of component BASE2
D, ALL, UY, 2 ! Define enforced motion 2 to be applied on all nodes along UY
SOLVE
FINISH

! Mode-Superposition Transient Analysis
/SOLU
ANTYPE, TRANSIENT
TRNOPT, MSUP
OUTRES, ALL,ALL
KBC,1
DELTIM,...
DVAL, 1, ACC, 100.0 ! Define constant acceleration for enforced motion 1
DVAL, 2, ACC, %ACCTAB% ! Define tabular acceleration for enforced motion 2
SOLVE

TIME,...
SOLVE

FINISH
    
```

### 3.9.6. Using Mode Selection

When performing a mode superposition-based analysis (spectrum, PSD, transient, or harmonic), modes can be selected for mode expansion. The selected modes can be included in a downstream mode superposition-based analysis. Consider using mode selection in the following cases:

- When spurious modes have been extracted that may lead to erroneous mode superposition results.
- When the model and/or the number of extracted modes is large.

Modes are selected before mode expansion, such that the `Jobname.MODE` file and the `Jobname.RST` file (from the modal analysis) contain only the selected modes. By reducing the total number of modes included in the superposition analysis, mode selection saves disk space and increases the performance of the post-processing of the modal results. This means that mode selection can be useful even when a downstream mode superposition analysis is not intended.

Mode selection is generally performed during a modal analysis restart. See [Modal Analysis Restart in the Mechanical APDL Basic Analysis Guide](#) for more details.

The following methods are available for mode selection:

- [Mode Selection Based on a User Defined Array \(p. 72\)](#)
- [Mode Selection Based on Modal Effective Mass \(p. 73\)](#)
- [Mode Selection Based on the Mode Coefficients \(p. 74\)](#)



- [Mode Selection Based on DDAM Procedure \(p. 75\)](#)

Zero-frequency modes are ignored when mode selection is based on a criterion (Modal Effective Mass, Mode Coefficients, and DDAM methods). Only the User Defined Array procedure can be used to select zero frequency modes.

### 3.9.6.1. Mode Selection Based on a User Defined Array

The mode selection is performed during a modal analysis or a modal analysis restart. It is applicable to real and complex eigensolvers.

The user defined array is a **\*DIM** array with a dimension equal to the number of extracted modes. The array values are 1 for selected modes and 0 otherwise.

Damped eigensolver solutions (**MODOPT,DAMP**) are pairs of complex conjugate modes. When filling in the table array, it is important to select the pairs (and not individual modes) to ensure that the **MODE** file remains consistent. (For example, modes #3 and #4 are a pair, so indices 3 and 4 in the array should have the same value, 0 or 1.)

---

#### Note

For the selection of complex modes from the QR damped eigensolver (**MODOPT,QRDAMP**), you must input **MXPAND,-1** in the modal solution. This will force the generation of a complex **MODE** file that will enable the complex mode selection.

---

A sample input listing of the procedure follows:

```
! Build the Model
/filnam,...
/title,...
/prep7
...
finish

! Obtain the Modal Solution
/solu
antype, MODAL
modopt, UNSYM, 10 ! Unsymmetric eigensolver, 10 modes are requested
solve
finish
save

/post1
... ! Post-process the modes to identify the modes of interest
finish

/clear, nostart ! Start a new session
resume,, db

*dim, tab, ARRAY, 10 ! Create the array
tab(1) = 1
tab(2) = 0
tab(3) = 0
tab(4) = 0
tab(5) = 1
tab(6) = 1
tab(7) = 1
tab(8) = 0
tab(9) = 1
tab(10) = 1

! Perform the modal selection and expansion
/solu
```



```

antype, MODAL, RESTART
mxpand, %tab%,,, yes ! Mode selection is based on user input array, and element results are
! requested
solve
finish

! Post-process the modal results or perform a mode superposition analysis

```

### 3.9.6.2. Mode Selection Based on Modal Effective Mass

The mode selection is performed during a modal analysis restart. It is applicable to real eigensolvers only.

You can select the modes per direction using one of the following methods:

- Use a significance threshold (*SIGNIF* on the **MXPAND** command). The significance level of a mode is equal to the modal effective mass divided by the total mass. The threshold is the same for all directions, but the user may define the specific global x, y, z directions used (**MODSELOPTIO**n command).
- Set a minimum ratio for the total modal effective mass to the total mass (**MODSELOPTIO**n command). This ratio can be input for each direction.

A sample input listing of this procedure using a minimum ratio follows:

```

! Build the Model
/filnam,...
/title,...
/prep7
...
finish

! Obtain the Modal Solution
/solu
antype, MODAL
modopt, LANB, ...          ! Block Lanczos
solve
finish

! Perform the modal selection and expansion
/solu
antype, MODAL, RESTART
mxpand,,,, yes,,, MODM ! Mode selection is based on the modal effective mass, and element
! results are requested
modseloptio, .90, no,no, no,no,no      ! Only direction X is selected, a minimum of 90% of the total
! mass is requested in this direction
solve
finish

! Post-process the modal results or perform a mode superposition analysis

```

---

#### Note

If the minimum ratio cannot be achieved, you may need to extract more modes and run the procedure again.

---

If you want to calculate residual vectors, do so during the modal restart analysis. Note that the residual vectors calculation is always based on all the extracted modes and not the selected modes.

### 3.9.6.3. Mode Selection Based on the Mode Coefficients

The mode selection is performed during a modal analysis restart. It is applicable to real eigensolvers only. This mode selection method is used when a Single Point Response Spectrum (SPRS) analysis is performed following the modal analysis.

You can select the modes using a significance threshold (*SIGNIF* on the **MXPAND** command). The significance level of a mode is equal to the mode coefficient divided by the maximum mode coefficients (of all modes per direction). The threshold is the same for all directions.

A sample input listing of the procedure follows:

```
! Build the Model
/filnam, ...
/title, ...
/prep7
...
finish

! Obtain the Modal Solution
/solu
antype, MODAL
modopt, LANB, ... ! Block Lanczos
solve
finish

! Obtain the SPRS Mode Coefficients
/solu
antype, SPECTRUM
spot, SPRS
svtyp, 2, 386.4      ! Input spectrum type
sed, 1, 0, 0        ! Input spectrum direction
freq, 1, 2, 5, 15, 30 ! Input spectrum frequencies
sv,, 0.1, 0.2, 0.2, 0.05, 0.05 ! Input spectrum values
solve
finish

! Perform the modal selection and expansion
/solu
antype, MODAL, RESTART
mxpand,,,, yes, 1e-2,, MODC ! Mode selection is based on the mode coefficients, the significance
! threshold is 0.01 and element results are requested
solve
finish

! Perform a spectrum analysis (combination step)
/solu
antype, SPECTRUM
spopt, SPRS ! Single Point Response Spectrum analysis
srss                      ! Square Root of Sum of Square combination
solve
finish
```

For multiple input spectra in a Single Point Response Spectrum, to obtain the mode coefficients in the spectrum analysis, the spectrum input definitions are requested, and each one will be followed by a **SOLVE** command.

---

#### Note

This procedure is not supported for Multiple Point Response Spectrum (MPRS) and Power Spectrum Density (PSD) analyses. However, you can use a dummy SPRS input spectrum to perform the mode selection for the subsequent MPRS and PSD analysis, or else use the mode selection based on the modal effective mass.

---

### 3.9.6.4. Mode Selection Based on DDAM Procedure

The mode selection is performed during a modal analysis restart. It is applicable to real eigensolvers only. The intent of this mode selection method is to perform a Dynamic Design Analysis Method (DDAM) Spectrum analysis following the modal analysis. It is based on the procedure described in NAVSEA [411].

In this procedure, the modes are selected using a significance threshold equal to 0.01. The significance level of a mode is equal to the modal effective weight divided by the total weight. The threshold is the same for all directions.

The sum of the modal effective weights of all calculated modes must be greater than 80% of the total weight, otherwise you may need to extract more modes and run the procedure again.

After the significance calculation, the procedure can select additional modes based on the nodal acceleration check. This check accounts for the localized response of relatively light-weight sub-components of critical areas. To apply the acceleration check, you need to select all the nodes of interest and create a nodal component named “\_DDAM\_NODCHK” (see [Creating Components](#))

The additional modes included correspond to those for which the nodal acceleration of any of the nodes of interest exceeds by 10% the maximum nodal acceleration calculated from the selected modes.

If you want to calculate residual vectors, do so during the modal restart analysis. Note that the residual vectors calculation is always based on all the extracted modes and not the selected modes.

A sample input listing of the procedure follows:

```
! Build the Model
/filnam, ...
/title, ...
/prep7
...
esel,s,,,78,89
esel,a,,,96
esel,a,,,112
nsle      ! Select nodes of interest
cm,_DDAM_NODCHK,node      ! Create a component for the
      ! nodal acceleration check
allsel,all,all
...

finish

! Obtain the modal solution
/solu
antype, MODAL
modopt, LANB, ...      ! Block Lanczos

solve
finish

! Obtain the DDAM Mode Coefficients
/solu
antype, SPECTRUM
spopt, DDAM      ! DDAM Spectrum analysis
addam,1.0,10.0,37.5,12.0,6.0 ! Input spectral acceleration
vddam,1.0,30.0,12.0,6.0 ! Input spectral velocity
sed,0,1,0      ! Input spectrum direction
solve
finish

! Perform the modal selection and expansion
/solu
antype, MODAL, RESTART
mxpand,,,, yes,,, DDAM ! Mode selection is based on DDAM
```

```
! procedure, the default significance
! threshold is 0.01 and element results are ! requested
solve
finish

! Perform a DDAM spectrum analysis (Combination Step)
/solu
antype, SPECTRUM
spopt, DDAM ! DDAM Spectrum analysis
nrls,... ! NRLSUM mode combination
dmprat,... ! Define damping ratio
solve
finish
```

---

## Chapter 4: Harmonic Analysis

---

Any sustained cyclic *load* will produce a sustained cyclic *response* (a *harmonic response*) in a structural system. *Harmonic analysis* gives you the ability to predict the sustained dynamic behavior of your structures, thus enabling you to verify whether or not your designs will successfully overcome resonance, fatigue, and other harmful effects of forced vibrations.

The following harmonic analysis topics are available:

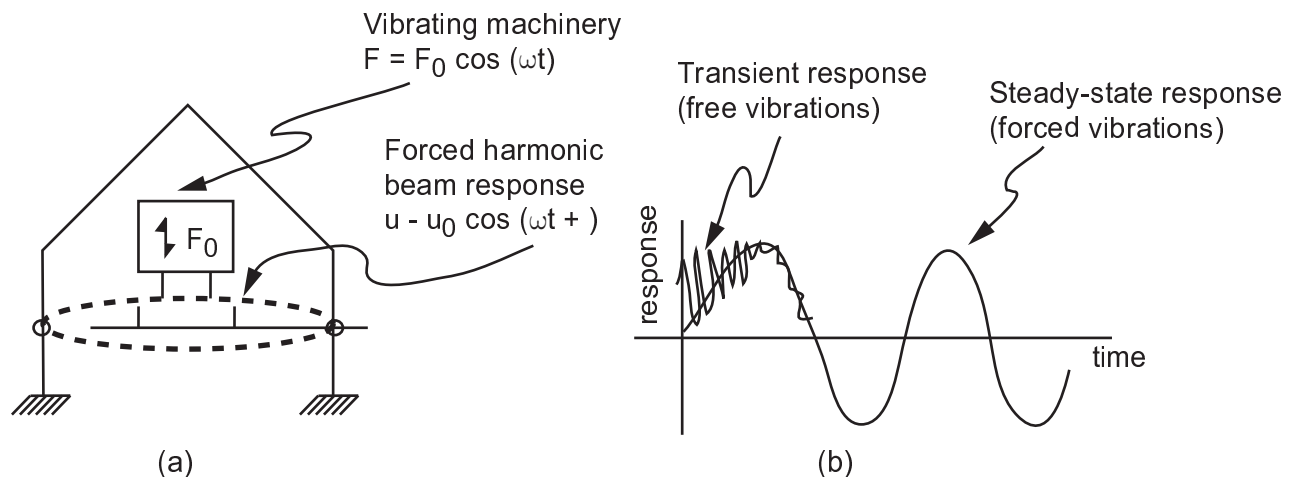
- 4.1. Uses for Harmonic Analysis
- 4.2. Commands Used in a Harmonic Analysis
- 4.3. Two Solution Methods
- 4.4. Performing a Harmonic Analysis
- 4.5. Sample Harmonic Analysis (GUI Method)
- 4.6. Example Harmonic Analysis (Command or Batch Method)
- 4.7. Where to Find Other Examples
- 4.8. Mode-Superposition Harmonic Analysis
- 4.9. Additional Harmonic Analysis Details

### 4.1. Uses for Harmonic Analysis

Harmonic analysis is a technique used to determine the steady-state response of a linear structure to loads that vary sinusoidally (*harmonically*) with time. The idea is to calculate the structure's response at several frequencies and obtain a graph of some response quantity (usually displacements) versus frequency. "Peak" responses are then identified on the graph and stresses reviewed at those peak frequencies.

This analysis technique calculates only the steady-state, forced vibrations of a structure. The transient vibrations, which occur at the beginning of the excitation, are not accounted for in a harmonic analysis (see [Figure 4.1: Harmonic Systems \(p. 77\)](#)).

**Figure 4.1: Harmonic Systems**



Typical harmonic system.  $F_0$  and  $\Omega$  are known.  $u_0$  and  $\Phi$  are unknown (a). Transient and steady-state dynamic response of a structural system (b).

Harmonic analysis is a linear analysis. Some nonlinearities, such as plasticity, are ignored, even if they are defined. You can, however, have unsymmetric system matrices such as those encountered in a [fluid-structure interaction \(FSI\)](#) problem. For more information, see the [Acoustic Analysis Guide](#) and the [Coupled-Field Analysis Guide](#).

Harmonic analysis can also be performed on a prestressed structure, such as a violin string (assuming the harmonic stresses are much smaller than the pretension stress). See [Prestressed Full Harmonic Analysis \(p. 104\)](#) and [Linear Perturbation Analysis](#) for more information on prestressed harmonic analyses.

## 4.2. Commands Used in a Harmonic Analysis

You use the same set of commands to build a model and perform a harmonic analysis that you use to do any other type of finite element analysis. Likewise, you choose similar options from the graphical user interface (GUI) to build and solve models no matter what type of analysis you are doing.

[Sample Harmonic Analysis \(GUI Method\) \(p. 90\)](#) and [Example Harmonic Analysis \(Command or Batch Method\) \(p. 96\)](#) show a sample harmonic analysis done via the GUI and via commands, respectively.

For detailed, alphabetized descriptions of the ANSYS commands, see the [Command Reference](#).

## 4.3. Two Solution Methods

Two harmonic analysis methods are available: *full* and *mode superposition*. (A third, relatively expensive method is to do a transient dynamic analysis with the harmonic loads specified as time-history loading functions; see [Transient Dynamic Analysis \(p. 107\)](#) for details.) The ANSYS Professional program allows only the mode-superposition method. Before we study the details of how to implement each of these methods, let's explore the advantages and disadvantages of each method.

### 4.3.1. The Full Method

The *full method* is the easiest available method. It uses the *full* system matrices to calculate the harmonic response (no matrix reduction). The matrices may be symmetric or unsymmetric. The advantages of the full method are:

- It is easy to use, because you don't have to worry about choosing mode shapes.
- It uses full matrices, so no mass matrix approximation is involved.
- It allows unsymmetric matrices, which are typical of such applications as acoustics and bearing problems.
- It calculates all displacements and stresses in a single pass.
- It accepts *all* types of loads: nodal forces, imposed (nonzero) displacements, and element loads (pressures and temperatures).
- It allows effective use of solid-model loads.

A disadvantage is that this method usually is more expensive than the other method when you use the sparse solver. However, when you use the JCG solver or the ICCG solver, the full method can be very efficient in some 3-D cases where the model is bulky and well-conditioned.

### 4.3.2. The Mode-Superposition Method

The *mode-superposition method* sums factored mode shapes (eigenvectors) from a modal analysis to calculate the structure's response. Its advantages are:

- It is faster and less expensive than the full method for many problems.
- Element loads applied in the preceding modal analysis can be applied in the harmonic analysis via the **LVSCALE** command.
- It allows solutions to be clustered about the structure's natural frequencies. This results in a smoother, more accurate tracing of the response curve.
- Prestressing effects can be included.
- It accepts modal damping (damping ratio as a function of frequency).

### 4.3.3. Restrictions Common to Both Methods

Both methods are subject to certain common restrictions:

- All loads must be sinusoidally time-varying.
- All loads must have the same frequency.
- No nonlinearities are permitted.
- Transient effects are not calculated.

You can overcome any of these restrictions by performing a transient dynamic analysis, with harmonic loads expressed as time-history loading functions. [Transient Dynamic Analysis \(p. 107\)](#) describes the procedure for a transient dynamic analysis.

## 4.4. Performing a Harmonic Analysis

We will first describe how to perform a harmonic analysis using the full method, and then list the steps that are different for the mode-superposition method.

### 4.4.1. Full Harmonic Analysis

The procedure for a full harmonic analysis consists of three main steps:

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

### 4.4.2. Build the Model

See [Building the Model](#) in the *Basic Analysis Guide*. For further details, see the *Modeling and Meshing Guide*.

### 4.4.2.1. Modeling Hints

Both a linear constitutive model and mass must be defined. Material properties may be linear, isotropic or anisotropic, and constant or field-dependent. Nonlinear material properties are ignored.

Only linear behavior is valid in a harmonic analysis. Nonlinear elements, if any, will be treated as linear elements. If you include contact elements, for example, their stiffnesses are calculated based on their initial status and are never changed. (For a prestressed full harmonic analysis, the program assumes that the initial status of the contact elements is the status at the completion of the static prestress analysis.) The reported nodal forces due to contact elements (**FSUM**,,CONT and **NFORCE**,CONT) are also based on the initial configuration, which may violate equilibrium conditions.

For a full harmonic analysis, frequency-dependent elastic and damping behaviors are supported using:

- An elastic or structural damping material data table (**TB** and **TBFIELD**)
- A viscoelastic material (for more information, see [Viscoelasticity in the Mechanical APDL Material Reference](#))
- A spring/damper element (**COMBIN14**)

### 4.4.3. Apply Loads and Obtain the Solution

In this step, you define the analysis type and options, apply loads, specify load step options, and initiate the finite element solution. Details of how to do these tasks are explained below.

---

#### Note

Peak harmonic response occurs at forcing frequencies that match the natural frequencies of your structure. Before obtaining the harmonic solution, you should first determine the natural frequencies of your structure by obtaining a modal solution (as explained in [Modal Analysis](#) (p. 39)).

---

#### 4.4.3.1. Enter the ANSYS Solution Processor

**Command(s):** /SOLU

**GUI:** Main Menu> Solution

#### 4.4.3.2. Define the Analysis Type and Options

ANSYS offers these options for a harmonic analysis:

**Table 4.1: Analysis Types and Options**

Option	Command	GUI Path
New Analysis	<b>ANTYPE</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; New Analysis</b>
Analysis Type: Harmonic	<b>ANTYPE</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; New Analysis&gt; Harmonic</b>
Solution Method	<b>HROPT</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; Analysis Options</b>



Option	Command	GUI Path
Solution Listing Format	<b>HROUT</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; Analysis Options</b>
Mass Matrix Formulation	<b>LUMPM</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; Analysis Options</b>
Equation Solver	<b>EQSLV</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; Analysis Options</b>

Each of these options is explained in detail below.

- **Option: New Analysis (ANTYPE)**

Choose New Analysis. Restarts are not valid in a harmonic analysis; if you need to apply additional harmonic loads, do a new analysis each time.

- **Option: Analysis Type: Harmonic (ANTYPE)**

Choose Harmonic as the analysis type.

- **Option: Solution Method (HROPT)**

Choose one of the following solution methods:

- Full method
- Mode-superposition method

- **Option: Solution Listing Format (HROUT)**

This option determines how the harmonic displacement solution is listed in the printed output (JOBNAME.OUT). You can choose between real and imaginary parts (default), and amplitudes and phase angles.

- **Option: Mass Matrix Formulation (LUMPM)**

Use this option to specify the default formulation (which is element dependent) or lumped mass approximation. We recommend the default formulation for most applications. However, for some problems involving "skinny" structures such as slender beams or very thin shells, the lumped mass approximation often yields better results. Also, the lumped mass approximation can result in a shorter run time and lower memory requirements.

After you complete the fields on the Harmonic Analysis Options dialog box, click on **OK** to reach a second Harmonic Analysis dialog box, where you choose an equation solver.

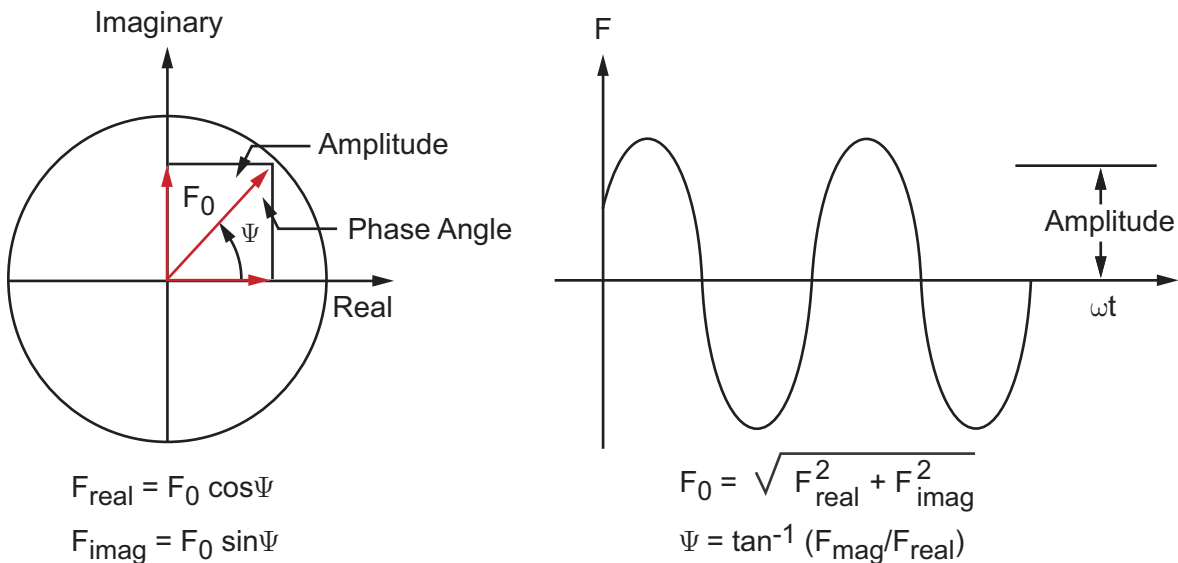
- **Option: Equation Solver (EQSLV)**

You can choose the sparse direct solver (SPARSE), the Jacobi Conjugate Gradient (JCG) solver, or the Incomplete Cholesky Conjugate Gradient (ICCG) solver. See the EQSLV command description or [Selecting a Solver](#) in the *Basic Analysis Guide* for details on selecting a solver.

### 4.4.3.3. Apply Loads on the Model

A harmonic analysis, by definition, assumes that any applied load varies harmonically (sinusoidally) with time. To completely specify a harmonic load, three pieces of information are usually required: the *amplitude*, the *phase angle*, and the *forcing frequency range* (see [Figure 4.2: Relationship Between Real/Imaginary Components and Amplitude/Phase Angle](#) (p. 82)).

**Figure 4.2: Relationship Between Real/Imaginary Components and Amplitude/Phase Angle**

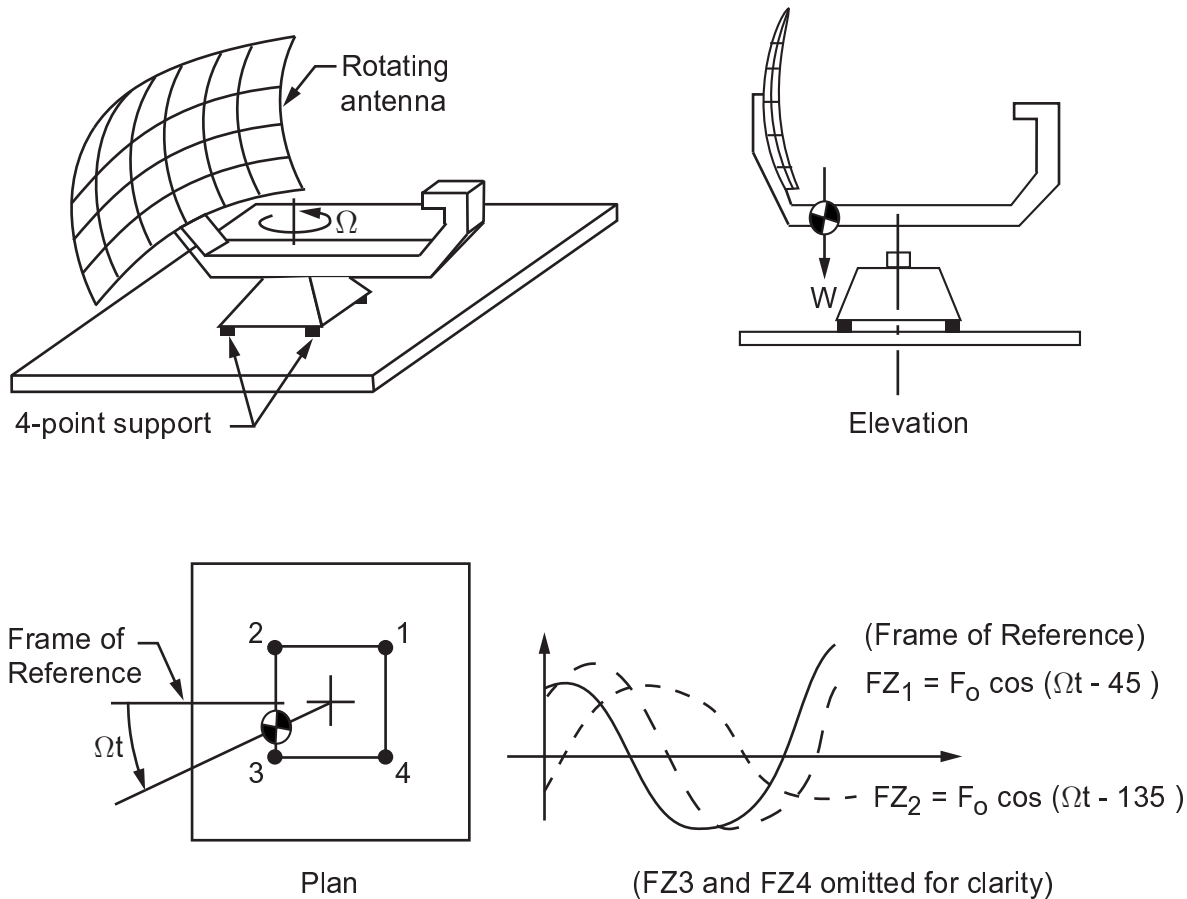


The *amplitude* is the maximum value of the load, which you specify using the commands shown in [Table 4.2: Applicable Loads in a Harmonic Analysis](#) (p. 83).

The *phase angle* is a measure of the time by which the load lags (or leads) a frame of reference. On the complex plane (see [Figure 4.2: Relationship Between Real/Imaginary Components and Amplitude/Phase Angle](#) (p. 82)), it is the angle measured from the real axis. The phase angle is required only if you have multiple loads that are out of phase with each other. For example, the unbalanced rotating antenna shown in [Figure 4.3: An Unbalanced Rotating Antenna](#) (p. 83) will produce out-of-phase vertical loads at its four support points. The phase angle cannot be specified directly; instead, you specify the real and imaginary components of the out-of-phase loads using the *VALUE* and *VALUE2* fields of the appropriate displacement and force commands. Pressures and other surface and body loads can only be specified with a phase angle of 0 (no imaginary component) with the following exceptions: nonzero imaginary components of pressures can be applied via the [SURF153](#), [SURF154](#), [SURF156](#), and [SURF159](#) elements in a full harmonic analysis, or using a mode-superposition harmonic analysis if the mode-extraction method is Block Lanczos, PCG Lanczos, Supernode, or Subspace (see the [SF](#) and [SFE](#) commands). [Figure 4.2: Relationship Between Real/Imaginary Components and Amplitude/Phase Angle](#) (p. 82) shows how to calculate the real and imaginary components.

The *forcing frequency range* is the frequency range of the harmonic load (in cycles/time). It is specified later as a load step option with the [HARFRQ](#) command.

**Figure 4.3: An Unbalanced Rotating Antenna**



An unbalanced rotating antenna will produce out-of-phase vertical loads at its four support points.

**Note**

A harmonic analysis cannot calculate the response to multiple forcing functions acting simultaneously with different frequencies (for example, two machines with different rotating speeds running at the same time). However, POST1 can superimpose multiple load cases to obtain the total response.

Table 4.2: [Applicable Loads in a Harmonic Analysis \(p. 83\)](#) summarizes the loads applicable to a harmonic analysis. Except for inertia loads, you can define loads either on the solid model (keypoints, lines, and areas) or on the finite element model (nodes and elements). For a general discussion of solid-model loads versus finite element loads, see [Loading](#) in the *Basic Analysis Guide*.

**Table 4.2: Applicable Loads in a Harmonic Analysis**

Load Type	Category	Cmd Family	GUI Path
Displacement (UX, UY, UZ, ROTX, ROTY, ROTZ)	Constraints	<b>D</b>	<b>Main Menu&gt; Solution&gt; Define Loads&gt; Apply&gt; Structural&gt; Displacement</b>

Load Type	Category	Cmd Family	GUI Path
Force, Moment (FX, FY, FZ, MX, MY, MZ)	Forces	<b>F</b>	<b>Main Menu&gt; Solution&gt; Define Loads&gt; Apply&gt; Structural&gt; Force/Moment</b>
Pressure (PRES)	Surface Loads	<b>SF</b>	<b>Main Menu&gt; Solution&gt; Define Loads&gt; Apply&gt; Structural&gt; Pressure</b>
Temperature (TEMP), Fluence (FLUE)	Body Loads	<b>BF</b>	<b>Main Menu&gt; Solution&gt; Define Loads&gt; Apply&gt; Structural&gt; Temperature</b>
Gravity, Spinning, etc.	Inertia Loads	-	<b>Main Menu&gt; Solution&gt; Define Loads&gt; Apply&gt; Structural&gt; Other</b>

In an analysis, loads can be applied, removed, operated on, or listed.

Spring preloads, initial strains, pretension loads, and thermal loads are ignored and do not generate a load.

#### 4.4.3.3.1. Applying Loads Using Commands

Table 4.3: Load Commands for a Harmonic Analysis (p. 84) lists all the commands you can use to apply loads in a harmonic analysis.

**Table 4.3: Load Commands for a Harmonic Analysis**

Load Type	Solid Model or FE	Entity	Apply	Delete	List	Operate	Apply Settings
Displacement	Solid Model	Keypoints	<b>DK</b>	<b>DKDELE</b>	<b>DKLIST</b>	<b>DTRAN</b>	-
	Solid Model	Lines	<b>DL</b>	<b>DLDELE</b>	<b>DLLIST</b>	<b>DTRAN</b>	-
	Solid Model	Areas	<b>DA</b>	<b>DADELE</b>	<b>DALIST</b>	<b>DTRAN</b>	-
	Finite Elem	Nodes	<b>D</b>	<b>DDELE</b>	<b>DLIST</b>	<b>DSCALE</b>	<b>DSYM, DCUM</b>
Force	Solid Model	Keypoints	<b>FK</b>	<b>FKDELE</b>	<b>FKLIST</b>	<b>FTRAN</b>	-
	Finite Elem	Nodes	<b>F</b>	<b>FDELE</b>	<b>FLIST</b>	<b>FSCALE</b>	<b>FCUM</b>
Pressure	Solid Model	Lines	<b>SFL</b>	<b>SFLDELE</b>	<b>SFLLIST</b>	<b>SFTRAN</b>	<b>SFGRAD</b>
	Solid Model	Areas	<b>SFA</b>	<b>SFADELE</b>	<b>SFALIST</b>	<b>SFTRAN</b>	<b>SFGRAD</b>
	Finite Elem	Nodes	<b>SF</b>	<b>SFDELE</b>	<b>SFLIST</b>	<b>SFSCALE</b>	<b>SFGRAD, SFCUM</b>
	Finite Elem	Elements	<b>SFE</b>	<b>SFEDELE</b>	<b>SFELIST</b>	<b>SFSCALE</b>	<b>SFGRAD, SFBEAM, SFFUN, SFCUM</b>

Load Type	Solid Model or FE	Entity	Apply	Delete	List	Operate	Apply Settings
Temperature, Flu- ence	Solid Model	Keypo- ints	<b>BFK</b>	<b>BFK- DELE</b>	<b>BFKLIST</b>	<b>BFTRAN</b>	-
	Solid Model	Lines	<b>BFL</b>	<b>BFLDELE</b>	<b>BFLLIST</b>	<b>BFTRAN</b>	-
	Solid Model	Areas	<b>BFA</b>	<b>BFADELE</b>	<b>BFALIST</b>	<b>BFTRAN</b>	-
	Solid Model	Volumes	<b>BFV</b>	<b>BFVDELE</b>	<b>BFVLIST</b>	<b>BFTRAN</b>	-
	Finite Elem	Nodes	<b>BF</b>	<b>BFDELE</b>	<b>BFLIST</b>	<b>BFS- CALE</b>	<b>BFCUM, BFUNIF, TUNIF</b>
	Finite Elem	Ele- ments	<b>BFE</b>	<b>BFEDELE</b>	<b>BFELIST</b>	<b>BFS- CALE</b>	<b>BFCUM</b>
Inertia	-	-	<b>ACEL, OMEGA, DO- MEGA, CGLOC, CGOMGA, DCGOMG</b>	-	-	-	-

#### 4.4.3.3.2. Applying Loads and Listing Loads Using the GUI

These steps for a harmonic analysis are the same as those for most other analyses. See [Applying Loads Using the GUI \(p. 45\)](#) and [Listing Loads \(p. 46\)](#) for more information.

#### 4.4.3.4. Specify Load Step Options

The following options are available for a harmonic analysis:

**Table 4.4: Load Step Options**

Option	Command	GUI Path
<b>General Options</b>		
Number of Harmonic Solutions	<b>NSUBST</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Time/Frequenc&gt; Freq and Substeps</b>
Stepped or Ramped Loads	<b>KBC</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Time/Frequenc&gt; Time - Time Step or Freq and Substeps</b>
<b>Dynamics Options</b>		
Forcing Frequency Range	<b>HARFRQ</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Time/Frequenc&gt; Freq and Substeps</b>
Damping	<b>ALPHAD, BETAD, DMPSTR,</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Time/Frequenc&gt; Damping</b>

Option	Command	GUI Path
	<b>MP</b> ,DMPR, <b>TB</b> ,SDAMP	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Other&gt; Change Mat Props&gt; Material Models&gt; Structural&gt; Damping</b>
	<b>MP</b> ,ALPD, <b>MP</b> ,BETD,	Not accessible from the GUI.
<b>Output Control Options</b>		
Printed Output	<b>OUTPR</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Output Ctrls&gt; Solu Printout</b>
Database and Results File Output	<b>OUTRES</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Output Ctrls&gt; DB/ Results File</b>
Extrapolation of Results	<b>ERESX</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Output Ctrls&gt; Integration Pt</b>

When specifying frequency dependent damping using **TB**,SDAMP you must specify the material property using **TB**,ELAS.

#### 4.4.3.4.1. General Options

General options include the following:

- *Number of Harmonic Solutions* (**NSUBST**)

You can request any number of harmonic solutions to be calculated. The solutions (or substeps) will be evenly spaced within the specified frequency range (**HARFRQ**). For example, if you specify 10 solutions in the range 30 to 40 Hz, the program will calculate the response at 31, 32, 33, ..., 39, and 40 Hz. No response is calculated at the lower end of the frequency range.

- *Stepped or Ramped Loads* (**KBC**)

The loads may be stepped or ramped. By default, they are ramped, that is, the load amplitude is gradually increased with each substep. By stepping the loads (**KBC**,1), the same load amplitude will be maintained for all substeps in the frequency range.

---

#### Note

Surface and body loads do *not* ramp from their previous load step values. They always ramp from zero or from the value specified via **BFUNIF**.

---

#### 4.4.3.4.2. Dynamics Options

Dynamics options include the following:

- *Forcing Frequency Range* (**HARFRQ**)

The forcing frequency range must be defined (in cycles/time) for a harmonic analysis. Within this range, you then specify the number of solutions to be calculated.

- *Damping*

Damping in some form should be specified; otherwise, the response will be infinity at the resonant frequencies. **ALPHAD** and **BETAD** result in a frequency-dependent damping ratio, whereas **DMPSTR** specifies a constant structural damping coefficient.

Damping can also be specified for individual materials using **MP,BETD**, **MP,ALPD**, and **MP,DMPR** or by using a [viscoelastic material model](#). See [Damping \(p. 2\)](#) for more information.

If no damping is specified in a direct harmonic analysis (full harmonic), the program uses zero damping by default.

- Alpha (Mass) Damping (**ALPHAD**)
- Beta (Stiffness) Damping (**BETAD**)
- Constant Structural Damping Coefficient (**DMPSTR**)
- Material Dependent (Mass) Damping Multiplier (**MP,ALPD**)
- Material Dependent (Stiffness) Damping Multiplier (**MP,BETD**)
- Constant Structural Material Damping Coefficient (**MP,DMPR**)
- Material Structural Damping Coefficient (**TB,SDAMP**)
- Viscoelastic Material Damping (**TB,PRONY**)

#### 4.4.3.4.3. Output Controls

Output control options include the following:

- *Printed Output* (**OUTPR**)

Use this option to include any results data on the output file (Jobname. OUT).

- *Database and Results File Output* (**OUTRES**)

This option controls the data on the results file (Jobname. RST).

- *Extrapolation of Results* (**ERESX**)

Use this option to review element integration point results by copying them to the nodes instead of extrapolating them (default).

#### 4.4.3.5. Save a Backup Copy of the Database to a Named File

You can then retrieve your model by reentering the ANSYS program and issuing **RESUME**.

**Command(s): SAVE**

**GUI: Utility Menu > File > Save as**

#### 4.4.3.6. Start Solution Calculations

**Command(s): SOLVE**

**GUI: Main Menu > Solution > Solve > Current LS**

#### 4.4.3.7. Repeat for Additional Load Steps

Repeat the process for any additional loads and frequency ranges (that is, for additional load steps). If you plan to do time-history postprocessing (POST26), the frequency ranges should *not* overlap from one load step to the next. Another method for multiple load steps, which allows you to store the load steps on files and then solve them at once using a macro, is described in the [Basic Analysis Guide](#).

#### 4.4.3.8. Leave SOLUTION

**Command(s): FINISH**

**GUI:** Close the Solution menu.

### 4.4.4. Review the Results

The results data for a harmonic analysis are the same as the data for a basic structural analysis with the following additions: If you defined damping in the structure, the response will be out-of-phase with the loads. All results are then complex in nature and are stored in terms of real and imaginary parts. Complex results will also be produced if out-of-phase loads were applied. See [Review the Results \(p. 19\)](#) in [Structural Static Analysis \(p. 9\)](#).

#### 4.4.4.1. Postprocessors

You can review these results using either POST26 or POST1. The normal procedure is to first use POST26 to identify critical forcing frequencies - frequencies at which the highest displacements (or stresses) occur at points of interest in the model - and to then use POST1 to postprocess the entire model at these critical forcing frequencies.

- POST1 is used to review results over the entire model at specific frequencies.
- POST26 allows you to review results at specific points in the model over the entire frequency range.

Some typical postprocessing operations for a harmonic analysis are explained below. For a complete description of all postprocessing functions, see [An Overview of Postprocessing](#) in the [Basic Analysis Guide](#).

#### 4.4.4.2. Points to Remember

The points to remember for a harmonic analysis are the same as those for most structural analyses. See [Points to Remember \(p. 19\)](#) in [Structural Static Analysis \(p. 9\)](#).

#### 4.4.4.3. Using POST26

POST26 works with tables of result item versus frequency, known as *variables*. Each *variable* is assigned a reference number, with variable number 1 reserved for frequency.

1. Define the variables using these options:  
**Command(s): NSOL, ESOL, RFORCE**



**GUI: Main Menu> TimeHist Postpro> Define Variables****Note**

The **NSOL** command is for primary data (nodal displacements), the **ESOL** command for derived data (element solution data, such as stresses), and the **RFORCE** command for reaction force data. To specify the total force, static component of the total force, damping component, or the inertia component, use the **FORCE** command.

- Graph the variables (versus frequency or any other variable). Then use **PLCPLX** to work with just the amplitude, phase angle, real part, or imaginary part.

**Command(s): PLVAR, PLCPLX**

**GUI: Main Menu> TimeHist Postpro> Graph Variables**

**Main Menu> TimeHist Postpro> Settings> Graph**

- Get a listing of the variable. To list just the extreme values, use the **EXTREM** command. Then use the **PRCPLX** command to work with amplitude and phase angle or real and imaginary part.

**Command(s): PRVAR, EXTREM, PRCPLX**

**GUI: Main Menu> TimeHist Postpro> List Variables> List Extremes**

**Main Menu> TimeHist Postpro> List Extremes**

**Main Menu> TimeHist Postpro> Settings> List**

Many other functions, such as performing math operations among variables (in complex arithmetic), moving variables into array parameters, moving array parameters into variables, etc., are available in POST26; see [The Time-History Postprocessor \(POST26\)](#) in the *Basic Analysis Guide* for details.

By reviewing the time-history results at strategic points throughout the model, you can identify the critical frequencies for further POST1 postprocessing.

**4.4.4.4. Using POST1**

- You can use the **SET** command to read in the results for the desired harmonic solution. It can read in either the real component, the imaginary component, the amplitude, or the phase.
- Display the deformed shape of the structure, contours of stresses, strains, etc., or vector plots of vector items (**PLVECT**). To obtain tabular listings of data, use **PRNSOL**, **PRESOL**, **PRRSOL**, etc.

- Option: Display Deformed Shape**

**Command(s): PLDISP**

**GUI: Main Menu> General Postproc> Plot Results> Deformed Shape**

- Option: Contour Displays**

**Command(s): PLNSOL or PLESOL**

**GUI: Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu or Element Solu**

Use these options to contour almost any result item, such as stresses (SX, SY, SZ...), strains (EPELX, EPELY, EPELZ...), and displacements (UX, UY, UZ...).

The *KUND* field on **PLNSOL** and **PLESOL** gives you the option of overlaying the undeformed shape on the display.

- Option: Vector Plots**

**Command(s): PLVECT**

**GUI: Main Menu> General Postproc> Plot Results> Vector Plot> Predefined**

Use **PLNSOL** or **PLESOL** to contour almost any result item, such as stresses (SX, SY, SZ...), strains (EPELX, EPELY, EPELZ...), and displacements (UX, UY, UZ...).

- **Option: Tabular Listings**

**Command(s):** **PRNSOL** (nodal results) **PRESOL** (element-by-element results) **PRRSOL** (reaction data) etc. **NSORT**, **ESORT**

**GUI: Main Menu> General Postproc> List Results> Nodal Solution**

**Main Menu> General Postproc> List Results> Element Solution**

**Main Menu> General Postproc> List Results> Reaction Solution**

Use the **NSORT** and **ESORT** commands to sort the data before listing them.

Many other functions, such as mapping results on to a path, transforming results to different coordinate systems, load case combinations, etc., are available in POST1; see [Solution](#) in the *Basic Analysis Guide* for details.

See the *Command Reference* for a discussion of the **ANTYPE**, **HROPT**, **HROUT**, **HARFRQ**, **DMPRAT**, **DMPSTR**, **NSUBST**, **KBC**, **NSOL**, **ESOL**, **RFORCE**, **PLCPLX**, **PLVAR**, **PRCPLX**, **PRVAR**, **PLDISP**, **PRRSOL**, and **PLNSOL** commands.

## 4.5. Sample Harmonic Analysis (GUI Method)

In this sample problem, you will determine the harmonic response of a two-mass-spring system.

### 4.5.1. Problem Description

Determine the response amplitude ( $X_i$ ) and phase angle ( $\Phi_i$ ) for each mass ( $m_i$ ) of the system shown below when excited by a harmonic force ( $F_1 \sin \Omega t$ ) acting on mass  $m_1$ .

### 4.5.2. Problem Specifications

Material properties for this problem are:

$$m_1 = m_2 = 0.5 \text{ lb-sec}^2/\text{in}$$

$$k_1 = k_2 = k_c = 200 \text{ lb/in}$$

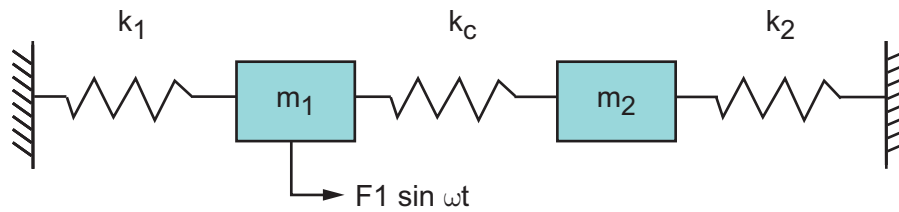
Loading for this problem is:

$$F_1 = 200 \text{ lb}$$

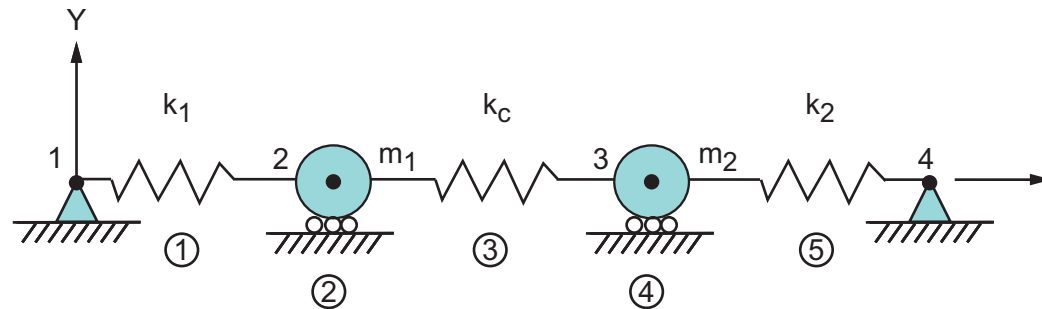
The spring lengths are arbitrarily selected and are used only to define the spring direction. Two master degrees of freedom are selected at the masses in the spring direction. A frequency range from zero to 7.5 Hz with a solution at  $7.5/30 = 0.25$  Hz intervals is chosen to give an adequate response curve. POST26 is used to get an amplitude versus frequency display.

### 4.5.3. Problem Diagram

Figure 4.4: Two-Mass-Spring-System



Problem Sketch



Representative Finite Element Model

#### 4.5.3.1. Set the Analysis Title

1. Choose menu path **Utility Menu > File > Change Title**.
2. Type the text "Harmonic Response of Two-Mass-Spring System" and click on OK.

#### 4.5.3.2. Define the Element Types

1. Choose menu path **Main Menu > Preprocessor > Element Type > Add/Edit/Delete**.
2. Click on Add. The Library of Element Types dialog box appears.
3. Scroll down the list on the left to "Combination" and select it.
4. Click once on "Spring-damper 14" in the list on the right.
5. Click on Apply.
6. Scroll up the list on the left to "Structural Mass" and select it.
7. Click once on "3D mass 21" in the list on the right.
8. Click on OK. The Library of Element Types dialog box closes.
9. Click on Close in the Element Types dialog box.

#### 4.5.3.3. Define the Real Constants

1. Choose menu path **Main Menu > Preprocessor > Real Constants**.

2. Click on Add. The Element Type for Real Constants dialog box appears.
3. Click once on Type 1 to highlight it.
4. Click on OK. The Real Constants for COMBIN14 dialog box appears.
5. Enter 200 for the spring constant (K). Click on OK.
6. Repeat steps 2-4 for Type 2, MASS21.
7. Enter .5 for mass in X direction and click on OK.
8. Click on Close to close the Real Constants dialog box.

#### **4.5.3.4. Create the Nodes**

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Nodes> In Active CS**.
2. Enter 1 for node number.
3. Enter 0, 0, 0 for the X, Y, and Z coordinates, respectively.
4. Click on Apply.
5. Enter 4 for node number.
6. Enter 1, 0, 0 for the X, Y, and Z coordinates, respectively.
7. Click on OK.
8. Choose menu path **Utility Menu> PlotCtrls> Numbering**. The Plot Numbering Controls dialog box appears.
9. Click once on "Node numbers" to turn node numbers on.
10. Click on OK.
11. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Nodes> Fill between Nds**. A picking menu appears.
12. In the ANSYS Graphics window, click once on nodes 1 and 4 (on the left and right sides of the screen). A small box appears around each node.
13. Click on OK on the picking menu. The Create Nodes Between 2 Nodes dialog box appears.
14. Click on OK to accept the default of 2 nodes to fill. Nodes 2 and 3 appear in the graphics window.

#### **4.5.3.5. Create the Spring Elements**

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Elements> Auto Numbered> Thru Nodes**. A picking menu appears.
2. In the graphics window, click once on nodes 1 and 2.
3. Click on Apply. A line appears between the selected nodes.

4. Click once on nodes 2 and 3.
5. Click on Apply. A line appears between the selected nodes.
6. Click once on nodes 3 and 4.
7. Click on OK. A line appears between the selected nodes.

#### **4.5.3.6. Create the Mass Elements**

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Elements> Elem Attributes**.
2. Enter 2 for element type number.
3. Enter 2 for real constant set number and click on OK.
4. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Elements> Auto Numbered> Thru Nodes**. A picking menu appears.
5. In the graphics window, click once on node 2.
6. Click on Apply.
7. Click once on node 3 and click on OK.

#### **4.5.3.7. Specify the Analysis Type, MDOF, and Load Step Specifications**

1. Choose menu path **Main Menu> Solution> Analysis Type> New Analysis**.
2. Click once on "Harmonic" and click on OK.
3. Choose menu path **Main Menu> Solution> Analysis Type> Analysis Options**.
4. Click once on "Full" to select the solution method.
5. Click once on "Amplitud + phase" to select the DOF printout format and click on OK.
6. Click OK in the Full Harmonic Analysis dialog box.
7. Choose menu path **Main Menu> Solution> Load Step Opts> Output Ctrl> Solu Printout**.
8. Click on "Last substep" to set the print frequency and click on OK.
9. Choose menu path **Main Menu> Solution> Load Step Opts> Time/Frequenc> Freq and Substeps**.
10. Enter 0 and 7.5 for the harmonic frequency range.
11. Enter 30 for the number of substeps.
12. Click once on "Stepped" to specify stepped boundary conditions.
13. Click on OK.

### **4.5.3.8. Define Loads and Boundary Conditions**

1. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On Nodes**. A picking menu appears.
2. Click on Pick All. The Apply U, ROT on Nodes dialog box appears.
3. In the scroll box for DOFs to be constrained, click once on "UY" to highlight it (make sure no other selections are highlighted).
4. Click on OK.
5. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On Nodes**. A picking menu appears.
6. In the graphics window, click once on nodes 1 and 4.
7. Click on OK. The Apply U, ROT on Nodes dialog box appears.
8. In the scroll box for DOFs to be constrained, click once on "UX" to highlight it and click once on "UY" to deselect it.
9. Click on OK.
10. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Force/ Moment> On Nodes**. A picking menu appears.
11. In the graphics window, click once on node 2.
12. Click on OK. The Apply F/M on Nodes dialog box appears.
13. In the scroll box for direction of force/moment, click once on "FX."
14. Enter 200 for the real part of force/moment and click on OK.

### **4.5.3.9. Solve the Model**

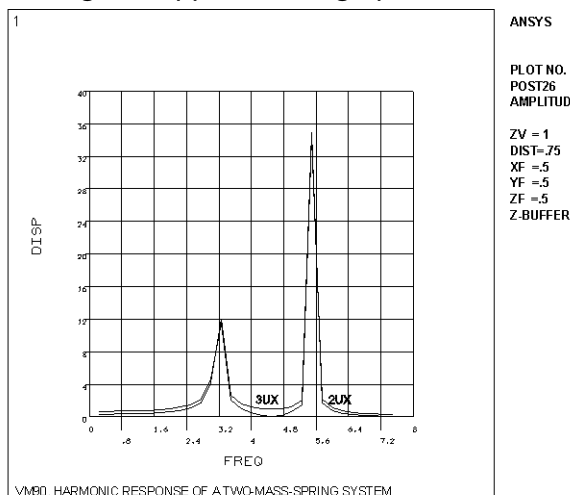
1. Choose menu path **Main Menu> Solution> Solve> Current LS**.
2. Review the information in the status window and click on Close.
3. Click on OK on the Solve Current Load Step dialog box to begin the solution.
4. When the solution is finished, a dialog box stating "Solution is done!" appears. Click on Close.

### **4.5.3.10. Review the Results**

For this sample, you will review the time-history results of nodes 2 and 3.

1. Choose menu path **Main Menu> TimeHist Postpro> Define Variables**. The Defined Time-History Variables dialog box appears.
2. Click on Add. The Add Time-History Variable dialog box appears.
3. Click on OK to accept the default of Nodal DOF result. The Define Nodal Data dialog box appears.

4. Enter 2 for reference number of variable.
5. Enter 2 for node number.
6. Enter 2UX for the user-specified label.
7. In the scroll box on the right, click once on "Translation UX" to highlight it.
8. Click on OK.
9. Click on Add in the Defined Time-History Variables dialog box. The Add Time-History Variable dialog box appears.
10. Click on OK to accept the default of Nodal DOF result. The Define Nodal Data dialog box appears.
11. Enter 3 for reference number of variable.
12. Enter 3 for node number.
13. Enter 3UX for the user-specified label.
14. In the scroll box on the right, click once on "Translation UX" to highlight it.
15. Click on OK.
16. Click on Close.
17. Choose menu path **Utility Menu> PlotCtrls> Style> Graphs**. The Graph Controls dialog box appears.
18. In the scroll box for type of grid, scroll to "X and Y lines" to select it.
19. Click on OK.
20. Choose menu path **Main Menu> TimeHist Postpro> Graph Variables**. The Graph Time-History Variables dialog box appears. Your graph should look like this:



Amplitude vs. Frequency

21. Enter 2 for 1st variable to graph.
22. Enter 3 for 2nd variable to graph.

23. Click on OK. A graph appears in the graphic window.

### 4.5.3.11. Exit ANSYS

You are now finished with this sample problem.

1. In the ANSYS Toolbar, click on Quit.
2. Choose the save option you want and click on OK.

## 4.6. Example Harmonic Analysis (Command or Batch Method)

You can perform the example harmonic analysis of a two-mass-spring system by using the following ANSYS commands instead of the GUI.

```

/prep7
et,1,combin14,,,2
et,2,mass21,,,4
r,1,200           ! spring constant = 200
r,2,.5           ! mass = 0.5
n,1
n,4,1
fill
e,1,2
e,2,3           ! spring element
e,3,4           ! spring element
type,2
real,2
e,2           ! mass element
e,3           ! mass element
finish

/solu
antype,harmic   ! harmonic analysis
hropt,full     ! full harmonic response
hrou,off       ! print results as amplitudes and phase angles
outpr,basic,1
nsubst,30      ! 30 intervals within freq. range
harfrq,,7.5   ! frequency range from 0 to 7.5 hz
kbc,1         ! step boundary condition
d,1,uy,,,4    ! constrain all 44 dof
d,1,ux,,,4,3  ! constrain nodes 1 and 4 in ux
f,2,fx,200
solve
finish

/post26
nsol,2,2,u,x,2ux   ! store ux displacements
nsol,3,3,u,x,3ux
/grid,1          ! turn grid on
/axlab,y,disp     ! y-axis label disp
!plvar,2,3       ! display variables 2 and 3
prvar,2,3
finish

```

## 4.7. Where to Find Other Examples

Several ANSYS publications, particularly the *Mechanical APDL Verification Manual*, describe additional harmonic analyses.

The *Mechanical APDL Verification Manual* consists of test case analyses demonstrating the analysis capabilities of the ANSYS family of products. While these test cases demonstrate solutions to realistic ana-



lysis problems, the *Mechanical APDL Verification Manual* does not present them as step-by-step examples with lengthy data input instructions and printouts. However, most ANSYS users who have at least limited finite element experience should be able to fill in the missing details by reviewing each test case's finite element model and input data with accompanying comments.

The *Mechanical APDL Verification Manual* includes a variety of harmonic analysis test cases:

- VM76 - Harmonic Response of a Guitar String
- VM86 - Harmonic Response of a Dynamic System
- VM87 - Equivalent Structural Damping
- VM88 - Response of an Eccentric Weight Exciter
- VM90 - Harmonic Response of a Two-Mass-Spring System
- VM176 - Frequency Response of Electrical Input Admittance for a Piezoelectric Transducer
- VM177 - Natural Frequency of a Submerged Ring
- VM183 - Harmonic Response of a Spring-Mass System
- VM203 - Dynamic Load Effect on Simply-Supported Thick Square Plate

## 4.8. Mode-Superposition Harmonic Analysis

The mode-superposition method sums factored mode shapes (obtained from a modal analysis) to calculate the harmonic response. It is the only method allowed in the ANSYS Professional program. The procedure to use the method consists of five main steps:

1. Build the model.
2. Obtain the modal solution.
3. Obtain the mode-superposition harmonic solution.
4. Expand the mode-superposition solution.
5. Review the results.

Of these, the first step is the same as described for the full method. The remaining steps are described below.

### 4.8.1. Obtain the Modal Solution

[Modal Analysis \(p. 39\)](#) describes how to obtain a modal solution. Following are some additional hints:

- The mode-extraction method should be Block Lanczos, PCG Lanczos, Supernode, Subspace, QR damped, or unsymmetric. (Other methods do not apply to mode superposition.)

If your model has damping and/or an unsymmetric stiffness or damping matrix, use the QR Damp mode-extraction method (**MODOPT,QRDAMP**) and do not request the calculation of the complex eigenmodes (*Cpxmod* = OFF on **MODOPT**).

If your model is undamped with an unsymmetric stiffness and/or mass matrix, use the unsymmetric mode-extraction method (**MODOPT,UNSYM**) and request both left and right modes (*ModType* = BOTH on **MODOPT**).

- Be sure to extract all modes that may contribute to the harmonic response.

- If you use the QR damped mode-extraction method, you must specify any **damping** (**ALPHAD**, **BETAD**, **MP**, **ALPD**, **MP**, **BETD**, or element damping including gyroscopic) that you want to include during the modal analysis. During the harmonic analysis, you can define additional damping, including a constant modal damping ratio (**DMPRAT**), constant structural material damping coefficients (**DMPSTR**, **MP**, **DMPR**), or the modal damping ratio as a function of mode (**MDAMP**). For more details about damping definition, see [Damping](#) (p. 2).
- If you need to apply harmonically varying *element loads* (pressures, temperatures, accelerations, etc.), specify them in the modal analysis. ANSYS ignores the loads for the modal solution, but calculates a load vector and writes it to the mode shape file (`Jobname.MODE`) and also writes the element load information to the `Jobname.MLV` file. You can generate multiple load vectors (see the **MODCONT** command in [Generating and Using Multiple Loads in Mode-Superposition Analyses](#) (p. 68)). The load vectors created can then be scaled and used in the harmonic solution.
- To include the contribution of higher frequency modes in the subsequent harmonic analysis, you can calculate the **residual vectors** in the modal analysis (see the **RESVEC** command in [Using the Residual-Vector Method to Improve Accuracy](#) (p. 66)).
- If the harmonic excitation is coming from the support, you can generate pseudo-static modes needed in the subsequent harmonic analysis (see the **MODCONT** command in [Enforced Motion Method for Mode-Superposition Transient and Harmonic Analyses](#) (p. 70)).
- You should expand the modes and calculate the element results to save computation time in the subsequent expansion of the harmonic results (**MXPAND**, **ALL**, **YES**, **YES**). Do not use this option if you are applying thermal loads, or if you desire to postprocess energies. Additionally, ensure that you output element nodal loads (**OUTRES**, **NLOAD**, **ALL**) if you want to postprocess total reaction forces.
- Do not change the model data (for example, nodal rotations) between the modal and harmonic analyses.
- You can select the modes of interest for the expansion. Because the expansion is performed on the selected modes only, you will save computational time in the subsequent mode superposition harmonic analysis. See [Using Mode Selection](#) (p. 71) for information about this procedure.

## 4.8.2. Obtain the Mode-Superposition Harmonic Solution

In this step, the program uses mode shapes extracted by the modal solution to calculate the harmonic response. The following requirements apply:

- The mode shape file (`Jobname.MODE`) must be available.
- If the unsymmetric eigensolver is used, the left mode shape file (`Jobname.LMODE`) must also be available (`ModType = BOTH` on the **MODOPT** command). Unlike the `Jobname.MODE` file which contains participation factors, load vectors, and other information, the `Jobname.LMODE` file only contains the left eigensolutions.
- The full file (`Jobname.FULL`) must be available if linear acceleration (**ACEL**) is present in the mode superposition analysis.
- The database must contain the same model from which the modal solution was obtained.

- The element modal load file (Jobname.MLV) must be available if load vectors were created (**MODCONT,ON**) and the element results were written on the Jobname.MODE (*MSUPkey* = YES on the **MXPAND** command) during the modal analysis.
1. Enter SOLUTION.  
**Command(s): /SOLU**  
**GUI: Main Menu > Solution**
  2. Define the analysis type and analysis options. These are the same as described for the full method, except for the following differences:
    - Choose the mode-superposition method of solution (**HROPT**).
    - Specify the modes you want to use for the solution (**HROPT**). This determines the accuracy of the harmonic solution. Generally, the number of modes specified should cover about 50 percent more than the frequency range of the harmonic loads.
    - To include the contribution of higher frequency modes, add the **residual vectors calculated in the modal analysis** (**RESVEC,ON**).
    - Optionally, cluster the solutions about the structure's natural frequencies (**HROUT**) for a smoother and more accurate tracing of the response curve.
    - Optionally, at each frequency, print a summary table that lists the contributions of each mode to the response (**HROUT**). Note, **OUTPR,NSOL** must be specified to print mode contributions at each frequency.
  3. Apply loads on the model. Harmonic loading is the same as described for the full method, except for the following restrictions:
    - Only forces, accelerations, and the load vector created in the modal analysis are valid. Use the **LVSCALE** command to apply the load vector from the modal solution. Use a zero scale factor on the **LVSCALE** command to suppress a load vector at a particular load step. Note that *ALL* loads from the modal analysis are scaled, including forces and accelerations. To avoid load duplication, delete any loads that were applied in the modal analysis.
- 
- Note**

You should apply accelerations in the modal analysis rather than in the harmonic analysis in order to obtain consistent reaction forces.
- 
- If the harmonic excitation is coming from support motion and you requested the pseudo-static modes in the modal analysis (**MODCONT** command), you can use the **DVAL** command to specify the enforced displacements and accelerations. Imposed nonzero displacements (**D** command) are ignored.
4. Specify load step options.
    - You can request any number of harmonic solutions to be calculated (**NSUBST**). The solutions (or sub-steps) will be evenly spaced within the specified frequency range (**HARFRQ**). For example, if you specify 10 solutions in the range 30 to 40 Hz, the program will calculate the response at 31, 32, 33, ..., 39, and 40 Hz. No response is calculated at the lower end of the frequency range.
    - For the cluster option, the **NSUBST** command specifies the number of solutions on each side of a natural frequency if the clustering option (**HROUT**) is chosen. The default is to calculate four solutions,

but you can specify any number of solutions from 2 through 20. (Any value over this range defaults to 10 and any value below this range defaults to 4.)

- Damping in some form should be specified; otherwise, the response will be infinity at the resonant frequencies. **ALPHAD** and **BETAD** result in a frequency-dependent damping ratio, whereas **DMPRAT** specifies a constant damping ratio to be used at all frequencies. **MDAMP** specifies a damping ratio for each individual mode. **DMPSTR** specifies a constant structural damping coefficient. See [Damping \(p. 2\)](#) for further details.
5. By default, if you used the Block Lanczos, PCG Lanczos, Supernode, Subspace, or unsymmetric option for the modal analysis (**MODOPT**,LANB; LANPCG; SNODE; SUBSP; or UNSYM), the modal coordinates (the factors to multiply each mode by) are written to the file `Jobname.RFRQ` and no output controls apply. If however you explicitly request not to write the element results to the `Jobname.MODE` file (**MX-PAND**,,,,,NO), the actual nodal displacements are written to the `.RFRQ` file. In that case, you may use a nodal component with the **OUTRES**,NSOL command to limit the displacement data written to the reduced displacement file `Jobname.RFRQ`. The expansion pass will only produce valid results for those nodes and for those elements in which all of the nodes of the elements have been written to the `.RFRQ` file. To use this option, first suppress all writing by invoking **OUTRES**,NSOL,NONE, then specify the item(s) of interest by invoking **OUTRES**,NSOL,ALL,component. Repeat the **OUTRES** command for any additional nodal components that you want to write to the `.RFRQ` file.
  6. Save a copy of the database.  
**Command(s):** **SAVE**  
**GUI:** **Utility Menu > File > Save as**
  7. Start solution calculations.  
**Command(s):** **SOLVE**  
**GUI:** **Main Menu > Solution > Solve > Current LS**
  8. Repeat steps 3 to 7 for any additional loads and frequency ranges (that is, for additional load steps). If you plan to do time-history postprocessing (POST26), the frequency ranges should *not* overlap from one load step to the next.
  9. Leave SOLUTION.  
**Command(s):** **FINISH**  
**GUI:** Close the Solution menu.

The mode-superposition harmonic solution is written to the reduced displacement file, `Jobname.RFRQ`, regardless of whether the Block Lanczos, PCG Lanczos, Supernode, Subspace, QR damped, or unsymmetric method was used for the modal solution. You will therefore need to expand the solution if you are interested in stress results.

### 4.8.3. Expand the Mode-Superposition Solution

The expansion pass starts with the harmonic solution on `Jobname.RFRQ` and calculates the displacement, stress, and force solution. These calculations are done only at frequencies and phase angles that you specify. Therefore, before you begin the expansion pass, you should review the results of the harmonic solution (using POST26) and identify the critical frequencies and phase angles.

An expansion pass is not always required. For instance, if you are interested mainly in displacements at specific points on the structure, then the displacement solution could satisfy your requirements. However, if you want to determine the stress or force solution, then you must perform an expansion pass.

### 4.8.3.1. Points to Remember

- The .RFRQ and .DB files from the harmonic solution, and the .MODE, .EMAT, .LMODE, .ESAV, and .MLV files from the modal solution must be available.
- The database must contain the same model for which the harmonic solution was calculated.

The procedure for the expansion pass follows:

### 4.8.3.2. Expanding the Modes

1. Reenter the ANSYS solution processor.

**Command(s):** /SOLU

**GUI:** Main Menu> Solution

---

#### Note

You must explicitly leave SOLUTION (using the **FINISH** command) and reenter (/SOLU) before performing the expansion pass.

---

2. Activate the expansion pass and its options. ANSYS offers these options for the expansion pass:

**Table 4.5: Expansion Pass Options**

Option	Command	GUI Path
Expansion Pass On/Off	<b>EXPASS</b>	Main Menu> Solution> Analysis Type> Expansion-Pass
No. of Solutions to Expand	<b>NUMEXP</b>	Main Menu> Solution> Load Step Opts> Expansion-Pass> Single Expand> Range of Solu's
Single Solution to Expand	<b>EXPSOL</b>	Main Menu> Solution> Load Step Opts> Expansion-Pass> Single Expand> By Time Freq
Phase Angle for Expansion	<b>HREXP</b>	Main Menu> Solution> Load Step Opts> Expansion-Pass> Single Expand> Range of Solu's
Nodal Solution Listing Format	<b>HROUT</b>	Main Menu> Solution> Analysis Type> Analysis Options

Each of these options is explained in detail below.

- **Option: Expansion Pass On/Off (EXPASS)**

Choose ON.

- **Option: Number of Solutions to Expand (NUMEXP,NUM)**

Specify the number. This number of evenly spaced solutions will be expanded over a frequency range (specified next). For example, **NUMEXP,4,1000,2000** specifies four solutions in the frequency range 1000 to 2000 (that is, expanded solutions at 1250, 1500, 1750, and 2000).

- **Option: Single Solution to Expand (EXPSOL)**

Use this option to identify a single solution for expansion if you do not need to expand multiple solutions in a range. You can specify it either by load step and substep number or by frequency. Also specify whether to calculate stresses and forces (default is to calculate both).

- **Option: Phase Angle for Expansion (HREXP)**

If multiple solutions are to be expanded over a frequency range (**NUMEXP**), we suggest that you request both the real and imaginary parts to be expanded (**HREXP,ALL**). This way, you can easily combine the two parts in POST26 to review the peak values of displacements, stresses, and other results.

If, on the other hand, a single solution is to be expanded (**EXPSOL**), you can specify the phase angle at which peak displacements occurred using **HREXP,angle**.

- **Option: Stress Calculations On/Off (NUMEXP or EXPSOL)**

You can turn off stress and force calculations if you are not interested in them. Default is to calculate stresses and forces.

- **Option: Nodal Solution Listing Format (HROUT)**

Determines how the harmonic displacement solution is listed in the printed output (Jobname.OUT). You can choose between real and imaginary parts (default), and amplitudes and phase angles.

3. Specify load step options. The only options valid for a harmonic expansion pass are output controls:

- *Printed Output (OUTPR)*

Use this option to include any results data on the printed output file (Jobname.OUT). Note that if the element results were calculated in the modal analysis, then no element output is available in the expansion pass. Use /POST1 to review element results.

- *Database and Results File Output (OUTRES)*

Use this option to control the data on the results file (Jobname.RST).

- *Extrapolation of Results (ERESX)*

Use this option to review element integration point results by copying them to the nodes instead of extrapolating them (default). Note that if the element results were calculated in the modal analysis, then this option is not applicable

---

### Note

The *FREQ* field on **OUTPR** and **OUTRES** can be only ALL or NONE.

---

4. Start expansion pass calculations.

**Command(s): SOLVE**

**GUI: Main Menu> Solution> Solve> Current LS**

5. Repeat steps 2, 3, and 4 for additional solutions to be expanded. Each expansion pass is stored as a separate load step on the results file.

6. Leave SOLUTION. You can now review results in the postprocessor.

**Command(s): FINISH**

**GUI:** Close the Solution menu.

#### 4.8.4. Review the Results of the Expanded Solution

You review results for an expansion pass in the same way that you review results for most structural analyses. See [Review the Results \(p. 19\)](#) in [Structural Static Analysis \(p. 9\)](#).

You can review these results using POST1. (If you expanded solution at several frequency points, you can also use POST26 to obtain graphs of displacement versus frequency, stress versus frequency, etc.) The only POST1 (or POST26) procedure difference between this method and the full method is that if you requested expansion at a specific phase angle (**HREXP**, angle) there is only one solution available for each frequency. Use the **SET** command to read in the results.

#### 4.8.5. Sample Input

A sample input listing for a mode-superposition harmonic analysis is shown below:

```
! Build the Model
/FILNAM,...           ! Jobname
/TITLE,...            ! Title
/PREP7                ! Enter PREP7
---
---                  ! Generate model
---
FINISH

! Obtain the Modal Solution
/SOLU                 ! Enter SOLUTION
ANTYPE,MODAL          ! Modal analysis
MODOPT,LANB           ! Block Lanczos
MXPAND,,,,YES.       ! Expand and calculate element results
D,...                 ! Constraints
SF,...                ! Element loads
SAVE
SOLVE                 ! Initiate modal solution
FINISH

! Obtain the Mode-Superposition Harmonic Solution
/SOLU                 ! Enter SOLUTION

ANTYPE,HARMIC         ! Harmonic analysis
HROPT,MSUP,...       ! Mode-superposition method; number of modes to use
HROUT,...             ! Harmonic analysis output options; cluster option
LVSCALE,...           ! Scale factor for loads from modal analysis
F,...                 ! Nodal loads
HARFRQ,...           ! Forcing frequency range
DMPRAT,...            ! Damping ratio
MDAMP,...             ! Modal damping ratios
NSUBST,...           ! Number of harmonic solutions
KBC,...              ! Ramped or stepped loads
SAVE
SOLVE                 ! Initiate solution
FINISH

! Review the Results of the Mode-Superposition Solution
/POST26
FILE,,RFRQ           ! Postprocessing file is Jobname.RFRQ
NSOL,...             ! Store nodal result as a variable
PLCPLX,...           ! Define how to plot complex variables
PLVAR,...            ! Plot variables
FINISH

! Expand the Solution (for Stress Results)
/SOLU! Re-enter SOLUTION
```



```

EXPASS,ON                ! Expansion pass
EXPSOL,...              ! Expand a single solution
HREXP,...               ! Phase angle for expanded solution
SOLVE
FINISH

! Review the Results of the Expanded Solution
/POST1
SET,...                 ! Read results for desired frequency
PLDISP,...             ! Deformed shape
PLNSOL,...             ! Contour plot of nodal results
---
FINISH

```

See the *Command Reference* for a discussion of the **ANTYPE**, **MODOPT**, **HROPT**, **HROUT**, **LVSCALE**, **F**, **HARFRQ**, **DMPRAT**, **DMPSTR**, **MDAMP**, **NSUBST**, **KBC**, **FILE**, **NSOL**, **PLCPLX**, **PLVAR**, **EXPASS**, **EXPSOL**, **HREXP**, **SET**, and **PLNSOL** commands.

## 4.9. Additional Harmonic Analysis Details

### 4.9.1. Prestressed Harmonic Analysis

A prestressed harmonic analysis calculates the dynamic response of a prestressed structure, such as a violin string. The prestress influences the stiffness of the structure through the stress-stiffening (or possibly a nonlinear tangent) matrix contribution. Response to harmonically varying loads is computed using this effective stiffness of the structure. The output stresses, therefore, will reflect the prestress effect on the structure.

The linear perturbation full harmonic procedure is the preferred method for performing a prestressed harmonic analysis. Because the [linear perturbation analysis](#) accommodates both linearly and nonlinearly prestressed cases, the prestressed effects of the structure from the previous linear or nonlinear, static or full transient analysis are included. For more information, see [Full Harmonic Analysis Based on Linear Perturbation](#) in the *Mechanical APDL Theory Reference*.

As alternatives to the linear perturbation full harmonic procedure, several other procedures for performing a prestressed harmonic analysis are described below.

#### 4.9.1.1. Prestressed Full Harmonic Analysis

The procedure described here for doing a prestressed full harmonic analysis is essentially the same as that for any other full harmonic analysis except that you first need to prestress the structure by doing a static analysis:

1. Build the model and obtain a static solution *with prestress effects turned on* (**PSTRES,ON**). The procedure to obtain the static solution is explained in [Structural Static Analysis](#) (p. 9).
2. Reenter SOLUTION and obtain the full harmonic solution, also with prestress effects turned on (reissue **PSTRES,ON**). Files `Jobname.EMAT` and `Jobname.ESAV` from the static analysis must be available. Hence, if another analysis is performed between the static and prestressed harmonic analyses, the static analysis will need to be rerun.

Because only linear behavior is valid in a harmonic analysis, nonlinear elements (if any) will be treated as linear elements. If you include contact elements, for example, their stiffnesses are calculated based on their initial status and are never changed. For a prestressed harmonic analysis, the program assumes that the initial status of the contact elements is the status at the completion of the static prestress analysis. (Note that the [linear perturbation](#) full harmonic



method provides more options for controlling the contact status compared to the method described here.)

If thermal body forces were present during the static prestress analysis, these thermal body forces must not be deleted during the full harmonic analysis or else the thermal prestress will vanish. Hence, any temperature loads used to define the thermal prestress must also be used in the full harmonic analysis as sinusoidally time-varying temperature loads. You should be aware of this limitation and exercise some judgement about whether or not to include temperature loads in their static prestress analysis.

#### **4.9.1.2. Prestressed Mode-Superposition Harmonic Analysis**

To include prestress effects in a mode-superposition analysis, you must first perform a prestressed *modal* analysis. Once prestressed modal analysis results are available, proceed as for any other mode-superposition analysis.

It is highly recommended that you perform a [linear perturbation](#) modal analysis prior to doing the downstream prestressed mode-superposition analysis. In this case, you must set both *Elcalc* and *MSUPkey* to YES on the **MXPAND** command during the linear perturbation modal analysis phases so that the downstream stress expansion pass can produce the consistent solution to the linear or nonlinear base (static or full transient) analysis. The prestressed nonlinear element history (saved variables) is accessible only in the first and second phases of the linear perturbation. The downstream MSUP or PSD analysis can only reuse this nonlinear information contained in the `Jobname.MODE` file which is generated in the linear perturbation.



---

## Chapter 5: Transient Dynamic Analysis

---

Transient dynamic analysis (sometimes called time-history analysis) is a technique used to determine the dynamic response of a structure under the action of any general time-dependent loads. You can use this type of analysis to determine the time-varying displacements, strains, stresses, and forces in a structure as it responds to any combination of static, transient, and harmonic loads. The time scale of the loading is such that the inertia or damping effects are considered to be important. If the inertia and damping effects are not important, you might be able to use a static analysis instead (see [Structural Static Analysis](#) (p. 9)).

The basic equation of motion solved by a transient dynamic analysis is

$$(M)\{\ddot{u}\} + (C)\{\dot{u}\} + (K)\{u\} = \{F(t)\}$$

where:

(M) = mass matrix

(C) = damping matrix

(K) = stiffness matrix

$\{\ddot{u}\}$  = nodal acceleration vector

$\{\dot{u}\}$  = nodal velocity vector

$\{u\}$  = nodal displacement vector

$\{F(t)\}$  = load vector

At any given time,  $t$ , these equations can be thought of as a set of "static" equilibrium equations that also take into account inertia forces  $((M)\{\ddot{u}\})$  and damping forces  $((C)\{\dot{u}\})$ . The program uses the Newmark time integration method or an improved method called HHT to solve these equations at discrete time points. The time increment between successive time points is called the *integration time step*.

The following topics are available for transient dynamic analysis:

[5.1. Preparing for a Transient Dynamic Analysis](#)

[5.2. Two Solution Methods](#)

[5.3. Performing a Full Transient Dynamic Analysis](#)

[5.4. Performing a Mode-Superposition Transient Dynamic Analysis](#)

[5.5. Performing a Prestressed Transient Dynamic Analysis](#)

[5.6. Transient Dynamic Analysis Options](#)

[5.7. Where to Find Other Examples](#)

For more information, see [Nonlinear Transient Analyses](#) (p. 200).

### 5.1. Preparing for a Transient Dynamic Analysis

A transient dynamic analysis is more involved than a static analysis because it generally requires more computer resources and more of *your* resources, in terms of the "engineering" time involved. You can save a significant amount of these resources by doing some preliminary work to understand the physics of the problem. For example, you can:

1. Analyze a simpler model first. A model of beams, masses, and springs can provide good insight into the problem at minimal cost. This simpler model may be all you need to determine the dynamic response of the structure.
2. If you are including nonlinearities, try to understand how they affect the structure's response by doing a static analysis first. In some cases, nonlinearities need not be included in the dynamic analysis.
3. Understand the dynamics of the problem. By doing a modal analysis, which calculates the natural frequencies and mode shapes, you can learn how the structure responds when those modes are excited. The natural frequencies are also useful for calculating the correct integration time step.
4. For a nonlinear problem, consider substructuring the linear portions of the model to reduce analysis costs. Substructuring is described in the [Advanced Analysis Guide](#).

## 5.2. Two Solution Methods

Two methods are available to do a transient dynamic analysis: *full* and *mode-superposition*. The ANSYS Professional program allows only the mode-superposition method. Before we study the details of how to implement each of these methods, we will examine the advantages and disadvantages of each.

### 5.2.1. Full Method

The *full method* uses the *full* system matrices to calculate the transient response (no matrix reduction). It is the more general method because it allows all types of nonlinearities to be included (plasticity, large deflections, large strain, and so on).

---

#### Note

If you do not want to include any nonlinearities, you should consider using the mode-superposition method because the full method is also the more expensive of the two.

---

The advantages of the full method are:

- It is easy to use, because you do not have to worry about choosing mode shapes.
- It allows all types of nonlinearities.
- It uses full matrices, so no mass matrix approximation is involved.
- All displacements and stresses are calculated in a single pass.
- It accepts *all* types of loads: nodal forces, imposed (nonzero) displacements (although not recommended), and element loads (pressures and temperatures) and allows tabular boundary condition specification via TABLE type array parameters.
- It allows effective use of solid-model loads.

The main disadvantage of the full method is that it is more expensive than the mode-superposition method.

For procedural information about using the full method, see [Performing a Full Transient Dynamic Analysis \(p. 109\)](#).

## 5.2.2. Mode-Superposition Method

The *mode-superposition method* sums factored mode shapes (eigenvectors) from a modal analysis to calculate the structure's response. This is the only method available in the ANSYS Professional program. Its advantages are:

- It is faster and less expensive than the full method for many problems.
- Element loads applied in the preceding modal analysis can be applied in the transient dynamic analysis via the **LVSCALE** command.
- It accepts modal damping (damping ratio as a function of mode number).

The disadvantages of the mode-superposition method are:

- The time step must remain constant throughout the transient, so automatic time stepping is not allowed.
- The only nonlinearity allowed is simple node-to-node contact (gap condition).
- It does not accept imposed (nonzero) displacements.

For procedural information about using the mode-superposition method, see [Performing a Mode-Superposition Transient Dynamic Analysis \(p. 120\)](#).

## 5.3. Performing a Full Transient Dynamic Analysis

Note - Before reading this section, you are encouraged to become familiar with the concepts presented in [Structural Static Analysis \(p. 9\)](#).

We will first describe how to do a transient dynamic analysis using the full method. We will then list the steps that are different for the mode-superposition method.

The procedure for a full transient dynamic analysis (available in the ANSYS Multiphysics, ANSYS Mechanical, and ANSYS Structural products) consists of these steps:

1. [Build the Model \(p. 110\)](#)
2. [Establish Initial Conditions \(p. 110\)](#)
3. [Set Solution Controls \(p. 113\)](#)
4. [Set Additional Solution Options \(p. 115\)](#)
5. [Apply the Loads \(p. 117\)](#)
6. [Save the Load Configuration for the Current Load Step \(p. 117\)](#)
7. [Repeat Steps 3-6 for Each Load Step \(p. 117\)](#)
8. [Save a Backup Copy of the Database \(p. 118\)](#)
9. [Start the Transient Solution \(p. 118\)](#)
10. [Exit the Solution Processor \(p. 118\)](#)

11. Review the Results (p. 118)

### 5.3.1. Build the Model

See *Building the Model* in the *Basic Analysis Guide*. For further details, see the *Modeling and Meshing Guide*.

#### 5.3.1.1. Points to Remember

Keep the following points in mind when building a model for a full transient dynamic analysis:

- You can use both linear and nonlinear elements.
- Both Young's modulus (EX) (or stiffness in some form) and density (DENS) (or mass in some form) must be defined. Material properties may be linear or nonlinear, isotropic or orthotropic, and constant or temperature-dependent.
- You can define damping using element damping, material damping and/or proportional damping ratios. For more details about damping definition, see [Damping \(p. 2\)](#).

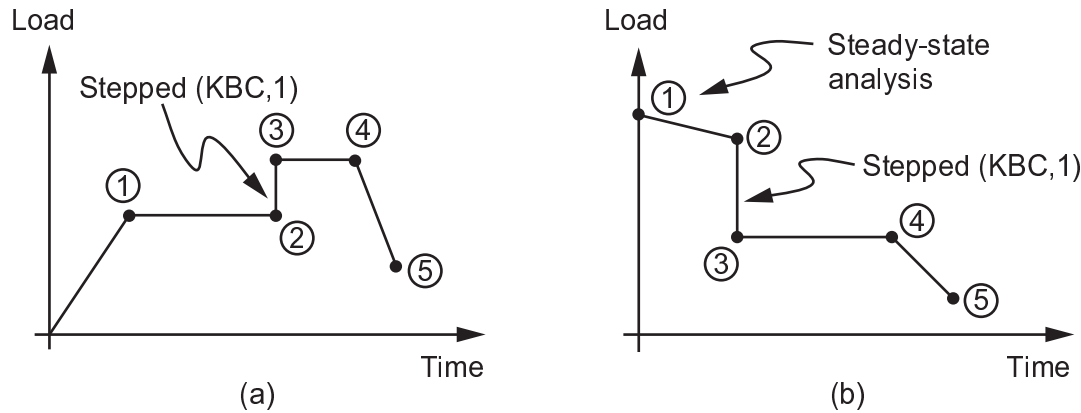
Some comments on mesh density:

- The mesh should be fine enough to resolve the highest mode shape of interest.
- Regions where stresses or strains are of interest require a relatively finer mesh than regions where only displacements are of interest.
- If you want to include nonlinearities, the mesh should be able to capture the effects of the nonlinearities. For example, plasticity requires a reasonable integration point density (and therefore a fine element mesh) in areas with high plastic deformation gradients.
- If you are interested in wave propagation effects (for example, a bar dropped exactly on its end), the mesh should be fine enough to resolve the wave. A general guideline is to have at least 20 elements per wavelength along the direction of the wave.

### 5.3.2. Establish Initial Conditions

Before you can perform a full transient dynamic analysis on a model, you need to understand how to establish initial conditions and use load steps.

A transient analysis, by definition, involves loads that are functions of time. To specify such loads, you need to divide the load-versus-time curve into suitable load steps. Each "corner" on the load-time curve may be one load step, as shown in [Figure 5.1: Examples of Load-Versus-Time Curves \(p. 111\)](#).

**Figure 5.1: Examples of Load-Versus-Time Curves**

The first load step you apply is usually to establish initial conditions. You then specify the loads and load step options for the second and subsequent transient load steps. For each load step, you need to specify both load values and time values, along with other load step options such as whether to step or ramp the loads, use automatic time stepping, and so on. You then write each load step to a file and solve all load steps together. Establishing initial conditions is described below; the remaining tasks are described later in this chapter.

The first step in applying transient loads is to establish initial conditions (that is, the condition at Time = 0). A transient dynamic analysis requires two sets of initial conditions (because the equations being solved are of second order): initial displacement ( $u_o$ ) and initial velocity ( $\dot{u}_o$ ). If no special action is taken, both  $u_o$  and  $\dot{u}_o$  are assumed to be zero. *Initial accelerations* ( $\ddot{u}_o$ ) are always assumed to be zero, but you can specify nonzero initial accelerations by applying appropriate acceleration loads over a small time interval.

**The following text describes how to apply different combinations of initial conditions:**

The term *initial displacement* as it appears in the following text can be any combination of displacement and force loads. Also, all load steps in the example input fragments that are run without applied transient effects (**TIMINT,OFF**) should be converged.

*Zero initial displacement and zero initial velocity* -- These are the default conditions, that is, if  $u_o = \dot{u}_o = 0$ , you do not need to specify anything. You may apply the loads corresponding to the first corner of the load-versus-time curve in the first load step.

*Nonzero initial displacement and/or nonzero initial velocity* -- You can set these initial conditions with the **IC** command.

**Command(s): IC**

**GUI: Main Menu > Solution > Define Loads > Apply > Initial Condit'n > Define**

**Caution**

Be careful not to define inconsistent initial conditions. For example, if you define an initial velocity at a single DOF, the initial velocity at every other DOF will be 0.0, potentially leading to conflicting initial conditions. In most cases, you will want to define initial conditions at every unconstrained DOF in your model. If these conditions are not the same at every DOF,

it is usually much easier to define initial conditions explicitly, as documented below (rather than by using the **IC** command).

See the *Command Reference* for a discussion of the **TIMINT** and **IC** commands.

*Zero initial displacement and nonzero initial velocity* - The nonzero velocity is established by applying small displacements over a small time interval on the part of the structure where velocity is to be specified. For example if  $\dot{u}_0 = 0.25$ , you can apply a displacement of 0.001 over a time interval of 0.004, as shown below.

```

...
TIMINT,OFF           ! Time integration effects off
D,ALL,UY,.001       ! Small UY displ. (assuming Y-direction velocity)
TIME,.004           ! Initial velocity = 0.001/0.004 = 0.25
LSWRITE             ! Write load data to load step file (Jobname.S01)
DDEL,ALL,UY         ! Remove imposed displacements
TIMINT,ON           ! Time integration effects on
...

```

*Nonzero initial displacement and nonzero initial velocity* - This is similar to the above case, except that the imposed displacements are actual values instead of "small" values. For example, if  $u_0 = 1.0$  and  $\dot{u}_0 = 2.5$ , you would apply a displacement of 1.0 over a time interval of 0.4:

```

...
TIMINT,OFF           ! Time integration effects off
D,ALL,UY,1.0        ! Initial displacement = 1.0
TIME,.4             ! Initial velocity = 1.0/0.4 = 2.5
LSWRITE             ! Write load data to load step file (Jobname.S01)
DDELE,ALL,UY       ! Remove imposed displacements
TIMINT,ON           ! Time integration effects on
...

```

*Nonzero initial displacement and zero initial velocity* - This requires the use of two substeps (**NSUBST,2**) with a *step* change in imposed displacements (**KBC,1**). Without the step change (or with just one substep), the imposed displacements would vary directly with time, leading to a *nonzero* initial velocity. The example below shows how to apply  $u_0 = 1.0$  and  $\dot{u}_0 = 0.0$ :

```

...
TIMINT,OFF           ! Time integration effects off for static solution
D,ALL,UY,1.0        ! Initial displacement = 1.0
TIME,.001           ! Small time interval
NSUBST,2            ! Two substeps
KBC,1               ! Stepped loads
LSWRITE             ! Write load data to load step file (Jobname.S01)
! Transient solution
TIMINT,ON           ! Time-integration effects on for transient solution
TIME,...           ! Realistic time interval
DDELE,ALL,UY       ! Remove displacement constraints
KBC,0               ! Ramped loads (if appropriate)
! Continue with normal transient solution procedures
...

```

*Nonzero initial acceleration* - This can be approximated by specifying the required acceleration (**ACEL**) over a small interval of time. For example, the commands to apply an initial acceleration of 9.81 would look like this:

```

...
ACEL,,9.81          ! Initial Y-direction acceleration
TIME,.001           ! Small time interval
NSUBST,2            ! Two substeps
KBC,1               ! Stepped loads

```



```

! The structure must be unconstrained in the initial load step, or
! else the initial acceleration specification will have no effect.
DDELE, ...           ! Remove displacement constraints (if appropriate)
LSWRITE             ! Write load data to load step file (Jobname.S01)

! Transient solution
TIME, ...           ! Realistic time interval
NSUBST, ...        ! Use appropriate time step
KBC,0              ! Ramped loads (if appropriate)
D, ...             ! Constrain structure as desired
! Continue with normal transient solution procedures
LSWRITE           ! Write load data to load step file (Jobname.S02)
...

```

See the *Command Reference* for discussions of the **ACEL**, **TIME**, **NSUBST**, **KBC**, **LSWRITE**, **DDELE**, and **KBC** commands.

### 5.3.3. Set Solution Controls

This step for a transient dynamic analysis is the same as for a basic structural analysis (see *Set Solution Controls* (p. 10) in *Structural Static Analysis* (p. 9)) with the following additions:

If you need to establish initial conditions for the full transient dynamic analysis (as described in *Establish Initial Conditions* (p. 110)), you must do so for the first load step of the analysis. You can then cycle through the Solution Controls dialog box additional times to set individual load step options for the second and subsequent load steps (as described in *Repeat Steps 3-6 for Each Load Step* (p. 117)).

#### 5.3.3.1. Access the Solution Controls Dialog Box

To access the Solution Controls dialog box, choose menu path **Main Menu> Solution> Analysis Type> Sol'n Control**. The following sections provide brief descriptions of the options that appear on each tab of the dialog box. *For details about how to set these options, select the tab that you are interested in and then click the **Help** button.* *Nonlinear Structural Analysis* (p. 193) also contains details about the nonlinear options introduced in this chapter.

#### 5.3.3.2. Using the Basic Tab

The **Basic** tab is active when you access the dialog box.

The controls that appear on the **Basic** tab provide the minimum amount of data needed for the analysis. 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 adjust the default settings for the more advanced controls. As soon as you click **OK** on any tab of the dialog box, the settings are applied to the database and the dialog box closes.

You can use the **Basic** tab to set the options listed in *Table 2.1: Basic Tab Options* (p. 11). For specific information about using the Solution Controls dialog box to set these options, access the dialog box, select the **Basic** tab, and click the **Help** button.

Special considerations for setting these options in a full transient analysis include:

- When setting **ANTYPE** and **NLGEOM**, choose **Small Displacement Transient** if you are performing a new analysis and you want to ignore large deformation effects such as large deflection, large rotation, and large strain. Choose **Large Displacement Transient** if you expect large deflections (as in the case of a long, slender bar under bending) or large strains (as in a metal-forming problem). Choose **Restart Current Analysis** if you want to restart a failed nonlinear analysis, if you have previously completed a static prestress or a full transient dynamic analysis and you want to extend the time-history, or if you wish to use the

Jobname .RSX information from a previous VT Accelerator run. Note that in a VT Accelerator run, you cannot restart a job in the middle; you can only rerun the job from the beginning with changes in the input parameters.

- When setting **AUTOTS**, remember that this load step option (which is also known as time-step optimization in a transient analysis) increases or decreases the integration time step based on the response of the structure. For most problems, we recommend that you turn on automatic time stepping, with upper and lower limits for the integration time step. These limits, specified using **DELTIM** or **NSUBST**, help to limit the range of variation of the time step; see [Automatic Time Stepping \(p. 132\)](#) for more information. The default is ON.
- **NSUBST** and **DELTIM** are load step options that specify the integration time step for a transient analysis. The integration time step is the time increment used in the time integration of the equations of motion. You can specify the time increment directly or indirectly (that is, in terms of the number of substeps). The time step size determines the accuracy of the solution: the smaller its value, the higher the accuracy. You should consider several factors in order to calculate a "good" integration time step; see [Guidelines for Integration Time Step \(p. 130\)](#) for details.
- When setting **OUTRES**, keep this caution in mind:

---

**Caution**

By default, only the last substep (time-point) is written to the results file (Jobname .RST) in a full transient dynamic analysis. To write all substeps, set the **Frequency** so that it writes all of the substeps.

---

### 5.3.3.3. Using the Transient Tab

You can use the **Transient** tab to set the options listed in [Table 5.1: Transient Tab Options \(p. 114\)](#). For specific information about using the Solution Controls dialog box to set these options, access the dialog box, select the **Transient** tab, and click the **Help** button.

**Table 5.1: Transient Tab Options**

Option	For more information about this option, see:
Specify whether time integration effects are on or off ( <b>TIMINT</b> )	<ul style="list-style-type: none"> <li>• <a href="#">Performing a Nonlinear Transient Analysis (p. 254)</a> in the <i>Structural Analysis Guide</i></li> </ul>
Specify whether to ramp the load change over the load step or to step-apply the load change ( <b>KBC</b> )	<ul style="list-style-type: none"> <li>• <a href="#">Stepped Versus Ramped Loads</a> in the <i>Basic Analysis Guide</i></li> <li>• <a href="#">Stepping or Ramping Loads</a> in the <i>Basic Analysis Guide</i></li> </ul>
Specify mass and stiffness damping ( <b>ALPHAD, BETAD</b> )	<ul style="list-style-type: none"> <li>• <a href="#">Damping (p. 2)</a> in the <i>Structural Analysis Guide</i></li> </ul>
Choose the time integration method, Newmark or HHT ( <b>TRNOPT</b> )	<ul style="list-style-type: none"> <li>• <a href="#">Transient Analysis</a> in the <i>Mechanical APDL Theory Reference</i></li> </ul>
Define integration parameters ( <b>TINTP</b> )	<ul style="list-style-type: none"> <li>• <a href="#">Mechanical APDL Theory Reference</a></li> </ul>

Special considerations for setting these options in a full transient analysis include:

- **TIMINT** is a dynamic load step option that specifies whether time integration effects are on or off. Time integration effects *must* be turned on for inertia and damping effects to be included in the analysis (otherwise a static solution is performed), so the default is to include time integration effects. This option is useful when beginning a transient analysis from an initial static solution; that is, the first load steps are solved with the time integration effects off.
- **ALPHAD** (alpha, or mass, damping) and **BETAD** (beta, or stiffness, damping) are dynamic load step options for specifying damping options. Damping in some form is present in most structures and should be included in your analysis. See [Damping Option \(p. 116\)](#) for other damping options.
- **TRNOPT** (*TINTOPT*) specifies the time integration method to be used. The default is Newmark method.
- **TINTP** is a dynamic load step option that specifies transient integration parameters. Transient integration parameters control the nature of the Newmark and HHT time integration techniques.

### 5.3.3.4. Using the Remaining Solution Controls Tabs

The options on the remaining tabs in the Solution Controls dialog box for a full transient analysis are the same as the ones you can set for a static structural analysis. See the following sections of [Structural Static Analysis \(p. 9\)](#) for a list of these options:

- [Using the Sol'n Options Tab \(p. 12\)](#)
- [Using the Nonlinear Tab \(p. 12\)](#)
- [Using the Advanced NL Tab \(p. 13\)](#). Exception: You cannot use arc-length options in a full transient analysis.

#### 5.3.3.4.1. Set Additional Solution Options

The additional solution options that you can set for a full transient analysis are mostly the same as the ones you can set for a static structural analysis. For a general description of what additional solution options are, along with descriptions of those options that are the same, see the following sections of [Structural Static Analysis \(p. 9\)](#):

- [Set Additional Solution Options \(p. 13\)](#)
- [Stress Stiffening Effects \(p. 13\)](#)
- [Newton-Raphson Option \(p. 14\)](#)
- [Creep Criteria \(p. 15\)](#)
- [Printed Output \(p. 15\)](#)
- [Extrapolation of Results \(p. 15\)](#)

Additional solution options for a full transient analysis that differ from those for a static analysis, or have different descriptions are presented in the following sections.

You may also use the **NLHIST** command to monitor results of interest in real time during solution. Before starting the solution, you can request nodal data such as displacements or reaction forces at specific nodes. You can also request element nodal data such as stresses and strains at specific elements to be

graphed. Pair-based contact data are also available. The result data are written to a file named `Jobname.nlh`.

For example, a reaction force-deflection curve could indicate when possible buckling behavior occurs. Nodal results and contact results are monitored at every converged substep while element nodal data are written as specified via the **OUTRES** setting.

You can also track results during batch runs. To execute, either access the Launcher and select **File Tracking** from the **Tools** menu, or type `nlhist150` at the command line. Use the supplied file browser to navigate to your `Jobname.nlh` file, and click on it to invoke the tracking utility. You can use this utility to read the file at any time, even after the solution is complete.

To use this option, use either of these methods:

**Command(s): NLHIST**

**GUI: Main Menu > Solution > Results Tracking**

#### 5.3.3.4.1.1. Prestress Effects

You may include prestress effects in your analysis. This requires element files from a previous static (or transient) analysis; see [Performing a Prestressed Transient Dynamic Analysis \(p. 129\)](#) for details.

**Command(s): PSTRES**

**GUI: Main Menu > Solution > Unabridged Menu > Analysis Type > Analysis Options**

#### 5.3.3.4.1.2. Damping Option

Use this load step option to include damping. Damping in some form is present in most structures and should be included in your analysis. In addition to setting **ALPHAD** and **BETAD** on the Solution Controls dialog box (as described in [Using the Transient Tab \(p. 114\)](#)), you can specify the following additional forms of damping for a full transient dynamic analysis:

- Material-dependent alpha and beta damping (**MP,ALPD** and **MP,BETD**)
- Element damping (**COMBIN14**, and so on)

To use the **MP** form of damping:

**Command(s): MP,ALPD or MP,BETD**

**GUI: Not accessible from the GUI.**

Note that constant material damping coefficient (**MP,DMPR**) is not applicable in transient analysis. See [Damping \(p. 2\)](#) for further details.

#### 5.3.3.4.1.3. Mass Matrix Formulation

Use this analysis option to specify a lumped mass matrix formulation. We recommend the default formulation for most applications. However, for some problems involving "skinny" structures, such as slender beams or very thin shells, the lumped mass approximation might provide better results. Also, the lumped mass approximation can result in a shorter run time and lower memory requirements.

To use this option:

**Command(s): LUMPM**

**GUI: Main Menu > Solution > Unabridged Menu > Analysis Type > Analysis Options**

### 5.3.4. Apply the Loads

You are now ready to apply loads for the analysis. The loads shown in [Table 2.5: Loads Applicable in a Static Analysis \(p. 16\)](#) are also applicable to a transient dynamic analysis. In addition to these, you can apply acceleration loads in a transient analysis (see [DOF Constraints](#) in the *Basic Analysis Guide* for more information).

Except for inertia loads, velocity loads, and acceleration loads, you can define loads either on the solid model (keypoints, lines, and areas) or on the finite element model (nodes and elements). In an analysis, loads can be applied, removed, operated on, or deleted. For a general discussion of solid-model loads versus finite-element loads, see [Loading](#) in the *Basic Analysis Guide*.

You can also apply time-dependent boundary conditions by defining a one-dimensional table (TABLE type array parameter). See [Applying Loads Using TABLE Type Array Parameters \(p. 17\)](#).

---

#### Note

It is recommended that you use acceleration input to define support motion when applicable (using the tabular array parameter definition on the **D,,ACC** or **ACEL** command). A displacement input (**D,,U** command) is likely to show numerical noise, because it implies a discontinuity in acceleration in the simulation. In this case, numerical damping (**GAMMA** on the **TINTP** command) can be used to improve the acceleration and force results.

---

### 5.3.5. Save the Load Configuration for the Current Load Step

As described in [Establish Initial Conditions \(p. 110\)](#), you need to apply loads and save the load configuration to a load step file for each corner of the load-versus-time curve. You may also want to have an additional load step that extends past the last time point on the curve to capture the response of the structure after the transient loading.

**Command(s):** **LSWRITE**

**GUI:** Main Menu > Solution > Load Step Opts > Write LS File

### 5.3.6. Repeat Steps 3-6 for Each Load Step

For each load step that you want to define for a full transient dynamic analysis, you need to repeat steps 3-6. That is, for each load step, reset any desired solution controls and options, apply loads, and write the load configuration to a file.

For each load step, you can reset any of these load step options: **TIMINT**, **TINTP**, **ALPHAD**, **BETAD**, **MP**, **ALPD**, **MP**, **BETD**, **TIME**, **KBC**, **NSUBST**, **DELTIM**, **AUTOTS**, **NEQIT**, **CNVTOL**, **PRED**, **LNSRCH**, **CRPLIM**, **NCNV**, **CUTCONTROL**, **OUTPR**, **OUTRES**, **ERESX**, and **RESCONTROL**.

An example load step file is shown below:

```

TIME, ...           ! Time at the end of 1st transient load step
Loads ...          ! Load values at above time
KBC, ...           ! Stepped or ramped loads
LSWRITE           ! Write load data to load step file
TIME, ...           ! Time at the end of 2nd transient load step
Loads ...          ! Load values at above time
KBC, ...           ! Stepped or ramped loads
LSWRITE           ! Write load data to load step file
TIME, ...           ! Time at the end of 3rd transient load step
Loads ...          ! Load values at above time
KBC, ...           ! Stepped or ramped loads

```

LSWRITE                   ! Write load data to load step file  
Etc.

### 5.3.7. Save a Backup Copy of the Database

Save a copy of the database to a named file. You can then retrieve your model by reentering the program and issuing **RESUME**.

**Command(s):** **SAVE**

**GUI:** Utility Menu > File > Save as

### 5.3.8. Start the Transient Solution

Use one of these methods to start the transient solution:

**Command(s):** **LSSOLVE**

**GUI:** Main Menu > Solution > Solve > From LS Files

For additional ways to create and solve multiple load steps (the array parameter method and the multiple **SOLVE** method), see [Solving Multiple Load Steps](#) in the *Basic Analysis Guide*.

### 5.3.9. Exit the Solution Processor

Use one of these methods to exit the solution processor:

**Command(s):** **FINISH**

**GUI:** Close the Solution menu.

### 5.3.10. Review the Results

You review results for a full transient analysis in the same way that you review results for most structural analyses. See [Review the Results](#) (p. 19) in [Structural Static Analysis](#) (p. 9).

#### 5.3.10.1. Postprocessors

You can review these results using either POST26, which is the time-history postprocessor, or POST1, which is the general postprocessor.

- POST26 is used to review results at specific points in the model as functions of time.
- POST1 is used to review results over the entire model at specific time points.

Some typical postprocessing operations for a transient dynamic analysis are explained below. For a complete description of all postprocessing functions, see [Postprocessors Available](#) in the *Basic Analysis Guide*.

#### 5.3.10.2. Points to Remember

The points to remember for a full transient analysis are the same as those for most structural analyses. See [Points to Remember](#) (p. 19) in [Structural Static Analysis](#) (p. 9).

#### 5.3.10.3. Using POST26

POST26 works with tables of result item versus time, known as *variables*. Each *variable* is assigned a reference number, with variable number 1 reserved for time.

1. Define the variables.

**Command(s):** **NSOL** (primary data, that is, nodal displacements) **ESOL** (derived data, that is, element solution data, such as stresses) **RFORCE** (reaction force data) **FORCE** (total force, or static, damping, or inertia component of total force) **SOLU** (time step size, number of equilibrium iterations, response frequency, and so on)

**GUI: Main Menu> TimeHist Postpro> Define Variables**

---

### Note

In the mode-superposition method, only static force is available with the **FORCE** command.

---

- Graph or list the variables. By reviewing the time-history results at strategic points throughout the model, you can identify the critical time points for further POST1 postprocessing.

**Command(s):** **PLVAR** (graph variables) **PRVAR, EXTREM** (list variables)

**GUI: Main Menu> TimeHist Postpro> Graph Variables**

**Main Menu> TimeHist Postpro> List Variables**

**Main Menu> TimeHist Postpro> List Extremes**

### 5.3.10.4. Other Capabilities

Many other postprocessing functions, such as performing math operations among variables, moving variables into array parameters, and moving array parameters into variables, are available in POST26. See [The Time-History Postprocessor \(POST26\)](#) in the *Basic Analysis Guide* for details.

### 5.3.10.5. Using POST1

- Read in model data from the database file.

**Command(s):** **RESUME**

**GUI: Utility Menu> File> Resume from**

- Read in the desired set of results. Use the **SET** command to identify the data set by load step and substep numbers or by time.

**Command(s):** **SET**

**GUI: Main Menu> General Postproc> Read Results> By Time/Freq**

- Perform the necessary POST1 operations. The typical POST1 operations that you perform for a transient dynamic analysis are the same as those that you perform for a static analysis. See [Typical Postprocessing Operations \(p. 20\)](#) for a list of these operations.

---

### Note

If you specify a time for which no results are available, the results that are stored will be a linear interpolation between the two nearest time points.

---

## 5.3.11. Sample Input for a Full Transient Dynamic Analysis

A sample input listing for a full transient analysis is shown below:

```
! Build the Model
/FILNAM,...           ! Jobname
/TITLE,...            ! Title
```



```

/PREP7                ! Enter PREP7
---
---! Generate model
---
FINISH

! Apply Loads and Obtain the Solution
/SOLU                 ! Enter SOLUTION
ANTYPE,TRANS         ! Transient analysis
TRNOPT,FULL          ! Full method
D,...                ! Constraints
F,...                ! Loads
SF,...
ALPHAD,...           ! Mass damping
BETAD,...            ! Stiffness damping
KBC,...              ! Ramped or stepped loads
TIME,...             ! Time at end of load step
AUTOTS,ON            ! Auto time stepping
DELTIM,...           ! Time step size
OUTRES,...           ! Results file data options
LSWRITE              ! Write first load step
---
---! Loads, time, etc. for 2nd load step
---
LSWRITE               ! Write 2nd load step
SAVE
LSSOLVE,1,2          ! Initiate multiple load step solution
FINISH
!
! Review the Results
/POST26
SOLU,...             ! Store solution summary data
NSOL,...             ! Store nodal result as a variable
ESOL,...             ! Store element result as a variable
RFORCE,...           ! Store reaction as a variable
PLVAR,...            ! Plot variables
PRVAR,...            ! List variables
FINISH

/POST1
SET,...              ! Read desired set of results into database
PLDISP,...           ! Deformed shape
PRRSOL,...           ! Reaction loads
PLNSOL,...           ! Contour plot of nodal results
PRERR                ! Global percent error (a measure of mesh adequacy)
---
---! Other postprocessing as desired
---
FINISH

```

See the *Command Reference* for discussions of the **ANTYPE**, **TRNOPT**, **ALPHAD**, **BETAD**, **KBC**, **TIME**, **AUTOTS**, **DELTIM**, **OUTRES**, **LSWRITE**, **LSSOLVE**, **SOLU**, **NSOL**, **ESOL**, **RFORCE**, **PLVAR**, **PRVAR**, **PLDISP**, **PRRSOL**, **PLNSOL**, and **PRERR** commands.

## 5.4. Performing a Mode-Superposition Transient Dynamic Analysis

The mode-superposition method scales the mode shapes obtained from a modal analysis and sums them to calculate the dynamic response. For more detailed information, see [Mode Superposition Method](#) in the *Mechanical APDL Theory Reference*.

This method is available in the ANSYS Multiphysics, ANSYS Mechanical, ANSYS Structural, and ANSYS Professional products. The procedure to use the method consists of five main steps:

1. Build the model.
2. Obtain the modal solution.



3. Obtain the mode-superposition transient solution.
4. Expand the mode-superposition solution.
5. Review the results.

### 5.4.1. Build the Model

Building the model for a mode-superposition transient dynamic analysis is the same as that described for the full method. See [Build the Model \(p. 110\)](#) for more information.

### 5.4.2. Obtain the Modal Solution

[Modal Analysis \(p. 39\)](#) describes how to obtain a modal solution. Following are some additional hints:

- The mode-extraction method should be Block Lanczos, PCG Lanczos, Supernode, Subspace, or QR damped. (The other methods, unsymmetric and damped, do not apply to mode-superposition.) If your model has damping and/or an unsymmetric stiffness matrix, use the QR Damp mode-extraction method (**MODOPT,QRDAMP**).
- Be sure to extract all modes that may contribute to the dynamic response.
- If you use the QR damped mode-extraction method, you must specify any damping (**ALPHAD, BETAD, MP,ALPD, MP,BETD, MP,DMPR** or element damping including gyroscopic) that you want to include in the modal analysis. During the transient analysis, you can define additional damping, including a constant modal damping ratio (**DMPRAT**) or the modal damping ratio as a function of mode (**MDAMP**). For more details about damping definitions, see [Damping \(p. 2\)](#)
- Specify displacement constraints, if any. Note that additional constraints may not be applied in the subsequent mode superposition transient analysis.
- If you need to apply element loads (pressures, temperatures, accelerations, etc.) in a transient dynamic analysis, you must specify them in the modal analysis. The loads are ignored for the modal solution, but a load vector will be calculated and written to the mode shape file (`Jobname.MODE`), and the element load information will be written to `Jobname.MLV`. You can generate multiple load vectors (see the **MODCONT** command in [Generating and Using Multiple Loads in Mode-Superposition Analyses \(p. 68\)](#)). You can then scale and use these load vectors in the transient solution.
- To include the contribution of higher frequency modes in the subsequent transient analysis, you can calculate the residual vectors in the modal analysis (see the **RESVEC** command in [Using the Residual-Vector Method to Improve Accuracy \(p. 66\)](#)).
- If the transient excitation is coming from the support, you can generate pseudo-static modes needed in the subsequent transient analysis (see the **MODCONT** command in [Enforced Motion Method for Mode-Superposition Transient and Harmonic Analyses \(p. 70\)](#)).
- You should expand the modes and calculate the element results to save computation time in the subsequent expansion of the transient results (**MXPAND,ALL,,,YES,,YES**). Do not use this option if you are applying thermal loads, or if you want to postprocess energies. The model data (for example, nodal rotations) should not be changed between the modal and transient analyses.
- You can select the modes of interest for the expansion. Because the expansion is performed on selected modes only, you will save computational time in the subsequent mode superposition transient analysis. See [Using Mode Selection \(p. 71\)](#) for information about this procedure.

### 5.4.3. Obtain the Mode-Superposition Transient Solution

In this step, the program uses mode shapes extracted by the modal solution to calculate the transient response. The following requirements apply:

- The mode shape file (`Jobname.MODE`) must be available.
- The full file (`Jobname.FULL`) must be available if linear acceleration (**ACEL**) is present in the mode superposition analysis.
- The database must contain the same model for which the modal solution was obtained.
- The element modal load file (`Jobname.MLV`) must be available if load vectors were created (**MOD-CONT,ON**) and the element results were written on the `Jobname.MODE` file (`MSUPkey = YES` on the `MPAND` command) during the modal analysis.

#### 5.4.3.1. Obtaining the Solution

The procedure to obtain the mode-superposition transient solution is described below:

1. Enter SOLUTION.  
**Command(s): /SOLU**  
**GUI: Main Menu > Solution**
2. Define the analysis type and analysis options. These are the same as the analysis options described for the full method (in [Set Solution Controls \(p. 113\)](#) and [Set Additional Solution Options \(p. 115\)](#)), except for the following differences:
  - You cannot use the Solution Controls dialog box to define analysis type and analysis options for a mode-superposition transient analysis. Instead, you must set them using the standard set of solution commands (which are listed in [Set Solution Controls \(p. 113\)](#) and [Set Additional Solution Options \(p. 115\)](#)) and the standard corresponding menu paths.
  - Restarts are available (**ANTYPE**).
  - Choose the *mode-superposition* method of solution (**TRNOPT**).
  - When you specify a mode-superposition transient analysis, a **Solution** menu appropriate for the specified analysis type appears. The **Solution** menu will be either “abridged” or “unabridged,” depending on the actions you took prior to this step in your session. The abridged menu contains only those solution options that are valid and/or recommended for mode-superposition transient analyses. If you are on the abridged **Solution** menu and you want to access other solution options (that is, solution options that are valid for you to use, but their use may not be encouraged for this type of analysis), select the **Unabridged Menu** option from the **Solution** menu. For details, see [Using Abridged Solution Menus](#) in the *Basic Analysis Guide*.
  - Specify the number of modes you want to use for the solution (**TRNOPT**). This determines the accuracy of the transient solution. At a minimum, you should use all modes that you think will contribute to the dynamic response. If you expect higher frequencies to be excited, for example, the number of modes specified should include the higher modes. The default is to use all modes calculated in the modal solution.
  - To include the contribution of higher frequency modes, add the [residual vectors calculated in the modal analysis \(RESVEC,ON\)](#).

- If you do not want to use rigid body (0 frequency) modes, use `MINMODE` on the `TRNOPT` command to skip over them.
  - Nonlinear options (`NLGEOM`, `NROPT`) are not available.
3. Define gap conditions, if any. They can only be defined between two nodal degrees of freedom (DOF) or between a nodal DOF and ground. *If you used the QR damped mode-extraction method, gap conditions are not supported.* More details about gap conditions are presented in the `GP` command description.

**Command(s):** `GP`

**GUI:** Main Menu > Solution > Dynamic Gap Cond > Define

4. Apply loads to the model. The following loading restrictions apply in a mode-superposition transient dynamic analysis:

- Only nodal forces (`F`) and accelerations applied via the `ACEL` command are available.

---

**Note**

For consistent reaction forces, apply accelerations in the modal analysis rather than in the transient analysis.

---

- A load vector created in the modal analysis can be included via the `LVSCALE` command (**Main Menu > Solution > Define Loads > Apply > Load Vector > For Mode Super**) to apply the load vector from the modal solution. Use a zero scale factor on the `LVSCALE` command to suppress a load vector at a particular load step. You can use such a load vector to apply element loads (pressures, temperatures, and so on) on the model. If you use `LVSCALE`, ensure that all nodal forces (`F`) defined in the modal analysis solution are removed in the transient analysis. Generally, you should apply nodal forces in the transient part of the analysis.
- If the transient excitation is coming from support motion and if you requested the pseudo-static modes in the modal analysis (`MODCONT` command), you can use the `DVAL` command to specify the enforced displacements and accelerations. Imposed nonzero displacements (`D` command) are ignored.

Multiple load steps are usually required to specify the load history in a transient analysis. The first load step is used to establish initial conditions, and second and subsequent load steps are used for the transient loading, as explained next.

5. Establish initial conditions. In modal superposition transient analyses, a first solution is done at `TIME = 0`. This establishes the initial condition and time step size for the entire transient analysis. Generally, the only load applicable for the first load step is initializing nodal forces. For this pseudo-static analysis, the mode-superposition method may yield poor results at `TIME = 0` if nonzero loads are applied.

The following load step options are available for the first load step:

**Table 5.2: Options for the First Load Step: Mode-Superposition Analysis**

Option	Command	GUI Path
<b>Dynamics Options</b>		

Option	Command	GUI Path
Transient Integration Parameters	<b>TINTP</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Time/Fre- quenc&gt; Time Integration</b>
Damping	<b>ALPHAD, BETAD, DM- PRAT, MDAMP</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Time/Fre- quenc&gt; Damping</b> <b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Other&gt; Change Mat Props&gt; Material Models&gt; Structural&gt; Damping</b>
<b>General Options</b>		
Integration Time Step	<b>DELTIM</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Time/Fre- quenc&gt; Time-Time Step</b>
<b>Output Control Options</b>		
Printed Output	<b>OUTPR</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Output Ctrls&gt; Solu Printout</b>

- Dynamics options include the following:

- *Transient Integration Parameters* (**TINTP**)

Transient integration parameters control the nature of the Newmark time integration technique. The default is to use the constant average acceleration scheme; see your [Mechanical APDL Theory Reference](#) for further details.

- *Damping*

Damping in some form is present in most structures and should be included in your analysis. You can specify five forms of damping in a mode-superposition transient dynamic analysis:

- Alpha (mass) damping (**ALPHAD**)
- Beta (stiffness) damping (**BETAD**)
- Constant damping ratio (**DMPRAT**)
- Modal damping (**MDAMP**)

See [Damping \(p. 2\)](#) for further details.

- The only valid general option for the first load step is *integration time step* (**DELTIM**), which is assumed to be constant throughout the transient. By default, the integration time step is assumed to be  $1/(20f)$ , where  $f$  is the highest frequency chosen for the solution. The **DELTIM** command is valid only in the first load step and is ignored in subsequent load steps.

---

### Note

If you do issue the **TIME** command in the first load step, it will be ignored. The first solution is always a static solution at  $TIME = 0$ .

---

- The output control option for the first load step is *printed output* (**OUTPR**). Use this option to control printout of the nodal solution.
6. Specify loads and load step options for the transient loading portion.
- **General options** include the following:
    - *Time Option* (**TIME**)  
This option specifies time at the end of the load step.
    - *Stepped or Ramped Loads* (**KBC**)  
This option indicates whether to ramp the load change over the load step (**KBC**) or to step-apply the load change (**KBC,1**). The default is ramped.
  - **Output control options** include the following:
    - *Printed Output* (**OUTPR**)  
Use this option to control printed output.
    - *Database and Results File Output* (**OUTRES**)  
This option controls the data on the reduced displacement file.  
  
The only valid label on these commands is NSOL (nodal solution). The default for **OUTRES** is to write the solution for every fourth time-point to the reduced displacement file (unless there are gap conditions defined, in which case the default is to write every solution). If you expanded element results during the modal analysis, then **OUTRES** is not applicable as the modal coordinates and not the displacements are written to `Jobname.RDSP`.
7. By default, if you used the Block Lanczos, PCG Lanczos, Supernode, or Subspace option for the modal analysis (**MODOPT,LANB; LANPCG; SNODE; or SUBSP**), the modal coordinates (the factors to multiply each mode by) are written to the file `Jobname.RDSP` and no output controls apply. If however you explicitly request not to write the element results to the `.MODE` file (**MXPAND,,,,,NO**), the actual nodal displacements are written to the `.RDSP` file. In that case, you may use a nodal component with the **OUTRES,NSOL** command to limit the displacement data written to the reduced displacement file `Jobname.RFRQ`. The expansion pass will only produce valid results for those nodes and for those elements in which all of the nodes of the elements have been written to the `.RFRQ` file. To use this option, first suppress all writing by invoking **OUTRES,NSOL,NONE**, then specify the item(s) of interest by invoking **OUTRES,NSOL,FREQ,component**. Repeat the **OUTRES** command for any additional nodal components that you want to write to the `.RDSP` file. Only one output frequency is allowed. (The program uses the last frequency specified by **OUTRES**.)
8. Save a backup copy of the database to a named file.  
**Command(s): SAVE**  
**GUI: Utility Menu > File > Save as**
9. Leave SOLUTION.  
**Command(s): FINISH**  
**GUI: Close the Solution menu.**

As an alternative method of resolution, you can issue the **LSWRITE** command to write each load step to a load step file (`Jobname.S01`) and then issue **LSSOLVE** to start the transient solution.

The mode-superposition transient solution is written to the reduced displacement file, `Jobname.RDSP`, regardless of whether the Block Lanczos, PCG Lanczos, Supernode, Subspace, or QR damped method was used for the modal solution. You will therefore need to expand the solution if you are interested in stress results.

### 5.4.4. Expand the Mode-Superposition Solution

The expansion pass starts with the transient solution on `jobname.RDSP` and calculates the displacement, stress, and force solution. These calculations are performed only at the time points you specify. Before you begin the expansion pass, therefore, you should review the results of the transient solution (using POST26) and identify the critical time points.

---

#### Note

An expansion pass is not always required. For example, if your primary interest is the displacement at specific points on the structure, then the displacement solution on `jobname.RDSP` could satisfy your requirements. However, if you are interested in the stress or force solution, then you must perform an expansion pass.

---

#### 5.4.4.1. Points to Remember

- The `.RDSP` and `.DB` files from the transient solution, along with the `.MODE`, `.EMAT`, `.ESAV` and `.MLV` files from the modal solution must be available.
- The database must contain the same model for which the transient solution was calculated.

The procedure for the expansion pass is explained below.

#### 5.4.4.2. Expanding the Solution

1. Reenter SOLUTION.

**Command(s):** `/SOLU`

**GUI:** Main Menu > Solution

---

#### Note

You must explicitly leave SOLUTION (using the **FINISH** command) and reenter (`/SOLU`) before performing the expansion pass.

---

2. Activate the expansion pass and its options.

**Table 5.3: Expansion Pass Options**

Option	Command	GUI Path
Expansion Pass On/Off	<b>EXPASS</b>	Main Menu > Solution > Analysis Type > Expansion Pass
No. of Solutions to be Expanded	<b>NUMEXP</b>	Main Menu > Solution > Load Step Opts > ExpansionPass > Range of Solu's
Single Solution to Expand	<b>EXPSOL</b>	Main Menu > Solution > Load Step Opts > ExpansionPass > Single Expand > By Time/Freq

- **Option: Expansion Pass On/Off (EXPASS)**

Choose ON.

- **Option: Number of Solutions to be Expanded (NUMEXP)**

Specify the number. This number of evenly spaced solutions will be expanded over the specified time range. The solutions nearest these times will be expanded. Also specify whether to calculate stresses and forces (default is to calculate both).

- **Option: Single Solution to Expand (EXPSOL)**

Use this option to identify a single solution for expansion if you do not need to expand multiple solutions in a range. You can specify it either by load step and substep number or by time. Also specify whether to calculate stresses and forces (default is to calculate both).

3. Specify load step options. The only options valid for a transient dynamic expansion pass are output controls:

- Output Controls

- *Printed Output (OUTPR)*

Use this option to include any results data on the output file (Jobname. OUT).

---

**Note**

If element results were calculated in the modal analysis, then no element output is available in the expansion pass. Use /POST1 to review the element results.

---

- *Database and Results File Output (OUTRES)*

This option controls the data on the results file (Jobname. RST).

- *Extrapolation of Results (ERESX)*

Use this option to review element integration point results by copying them to the nodes instead of extrapolating them (default).

---

**Note**

The *FREQ* field on **OUTPR** and **OUTRES** can only be ALL or NONE.

---

4. Start expansion pass calculations.

**Command(s): SOLVE**

**GUI: Main Menu> Solution> Solve> Current LS**

5. Repeat steps 2, 3, and 4 for additional solutions to be expanded. Each expansion pass is stored as a separate load step on the results file.

6. Leave SOLUTION.

**Command(s): FINISH**



**GUI:** Close the Solution window.

### 5.4.4.3. Reviewing the Results of the Expanded Solution

You review results for an expansion pass in the same way that you review results for most structural analyses. See Review the Results in "Structural Static Analysis".

You can review these results using POST1. (If you expanded solutions at several time points, you can also use POST26 to obtain graphs of stress versus time, strain versus time, and so on.) The procedure to use POST1 (or POST26) is the same as described for the full method.

### 5.4.5. Review the Results

Results consist of displacements, stresses, and reaction forces at each time-point for which the solution was expanded. You can review these results using POST26 or POST1, as explained for the full method (see [Review the Results \(p. 118\)](#)).

---

#### Note

Reaction forces and other force output (**PRRSOL**, **FSUM**, **RFORCE**, etc.) contain only the static contributions. The inertial and damping contributions are not included or available (**FORCE** command).

---

### 5.4.6. Sample Input for a Mode-Superposition Transient Dynamic Analysis

A sample input listing for a mode-superposition transient analysis is shown below:

```
! Build the Model
/FILNAM,...           ! Jobname
/TITLE,...            ! Title
/PREP7                ! Enter PREP7
---
---! Generate model
---
FINISH

! Obtain the Modal Solution
/SOLU                 ! Enter SOLUTION
ANTYPE,MODAL          ! Modal analysis
MODOPT,LANB           ! Block Lanczos
MXPAND,,,YES          ! Expand the results and calculate element results
D,...                 ! Constraints
SF,...                ! Element loads
ACEL,...
SAVE
SOLVE
FINISH

! Obtain the Mode-Superposition Transient Solution
/SOLU                 ! Reenter SOLUTION
ANTYPE,TRANS          ! Transient analysis
TRNOPT,MSUP,...       ! Mode-superposition method
LVSCALE,...           ! Scale factor for element loads
F,...                 ! Nodal Loads
MDAMP,...              ! Modal damping ratios
DELTIM,...            ! Integration time step sizes
SOLVE                 ! Solve 1st load step
---                  ! Remember: The 1st load step is
---                  !      solved statically at time=0.
---
---! Loads, etc. for 2nd load step
TIME,...              ! Time at end of second load step
```



```

KBC,...           ! Ramped or stepped loads
OUTRES,...        ! Results-file data controls
---
SOLVE             ! Solve 2nd load step (first transient load step)
FINISH

! Review results of the mode-superposition solution
/POST26           ! Enter POST26
FILE,,RDSP        ! Results file is Jobname.RDSP
SOLU,...          ! Store solution summary data
NSOL,...          ! Store nodal result as a variable
PLVAR,...         ! Plot variables
PRVAR,...         ! List variables
FINISH

! Expand the Solution
/SOLU             ! Reenter SOLUTION
EXPASS,ON         ! Expansion pass
NUMEXP,...        ! No. of solutions to expand; time range
OUTRES,...        ! Results-file data controls
SOLVE
FINISH

! Review the Results of the Expanded Solution
/POST1
SET,...           ! Read desired set of results into database
PLDISP,...        ! Deformed shape
PRRSOL,...        ! Reaction loads
PLNSOL,...        ! Contour plot of nodal results
PRERR             ! Global percent error (a measure of mesh adequacy)
---
---! Other postprocessing as desired
---
FINISH

```

See the *Command Reference* for discussions of the **ANTYPE**, **MODOPT**, **ACEL**, **TRNOPT**, **LVSCALE**, **MDAMP**, **DELTIM**, **TIME**, **KBC**, **OUTRES**, **LSSOLVE**, **FILE**, **SOLU**, **NSOL**, **PLVAR**, **PRVAR**, **EXPASS**, **NUMEXP**, **OUTRES**, **PLDISP**, **PRRSOL**, **PLNSOL**, and **PRERR** commands.

## 5.5. Performing a Prestressed Transient Dynamic Analysis

A prestressed transient dynamic analysis calculates the dynamic response of a prestressed structure, such as a heat-treated part with residual thermal stresses. Prestressed-analysis procedures vary, depending on the type of transient dynamic analysis being performed.

### 5.5.1. Prestressed Full Transient Dynamic Analysis

You can include prestressing effects in a full transient dynamic analysis by applying the prestressing loads in a preliminary static load step. (Do not remove these loads in subsequent load steps.) The procedure consists of two steps:

1. Build your model, enter SOLUTION, and define a transient analysis type (**ANTYPE**,TRANS).
  - Apply all prestressing loads.
  - Turn time integration effects off (**TIMINT**,OFF).
  - If you need to include large strain, large deflection, or stress stiffening effects, issue **NLGEOM**,ON.
  - Set time equal to some small dummy value (**TIME**).
  - Write your first load step to Jobname.S01 (**LSWRITE**).

If prestressing effects develop because of nonlinear behavior (as in the case of residual thermal stresses in a casting), several load steps might be required to complete the static prestressing phase of your analysis. In the case of geometric nonlinearities (large deformation effects), you can capture the prestressing effect by issuing **NLGEOM,ON**.

- For all subsequent load steps, turn time integration effects on (**TIMINT,ON**), and proceed using the full transient dynamic analysis procedures described previously. Once all load steps are written to files (**LSWRITE**), you can initiate the multiple load step solution (**LSSOLVE**).

---

### Note

If you intend to define initial conditions (**IC**), perform the static prestress solution as a separate solution. To activate the gyroscopic damping matrix in a prestressed transient analysis, perform a separate static solution with Coriolis effects activated (**CORIOLIS,ON,,,ON**) in a [stationary reference frame](#).

**(Main Menu> Solution> Define Loads> Apply> Initial Condit'n> Define)**

The **IC** command is valid only in the first load step.

---

## 5.5.2. Prestressed Mode-Superposition Transient Dynamic Analysis

In order to include prestress effects in a mode-superposition analysis, you must first do a prestressed *modal* analysis. See [Modal Analysis \(p. 39\)](#) for details. Once prestressed modal analysis results are available, proceed as for any other mode-superposition analysis.

## 5.6. Transient Dynamic Analysis Options

The following sections provide additional details about defining integration time step, automatic time stepping, and damping.

### 5.6.1. Guidelines for Integration Time Step

The accuracy of the transient dynamic solution depends on the integration time step: the smaller the time step, the higher the accuracy. A time step that is too large introduces an error that affects the response of the higher modes (and hence the overall response). A time step that is too small wastes computer resources. To calculate an optimum time step, adhere to the following guidelines:

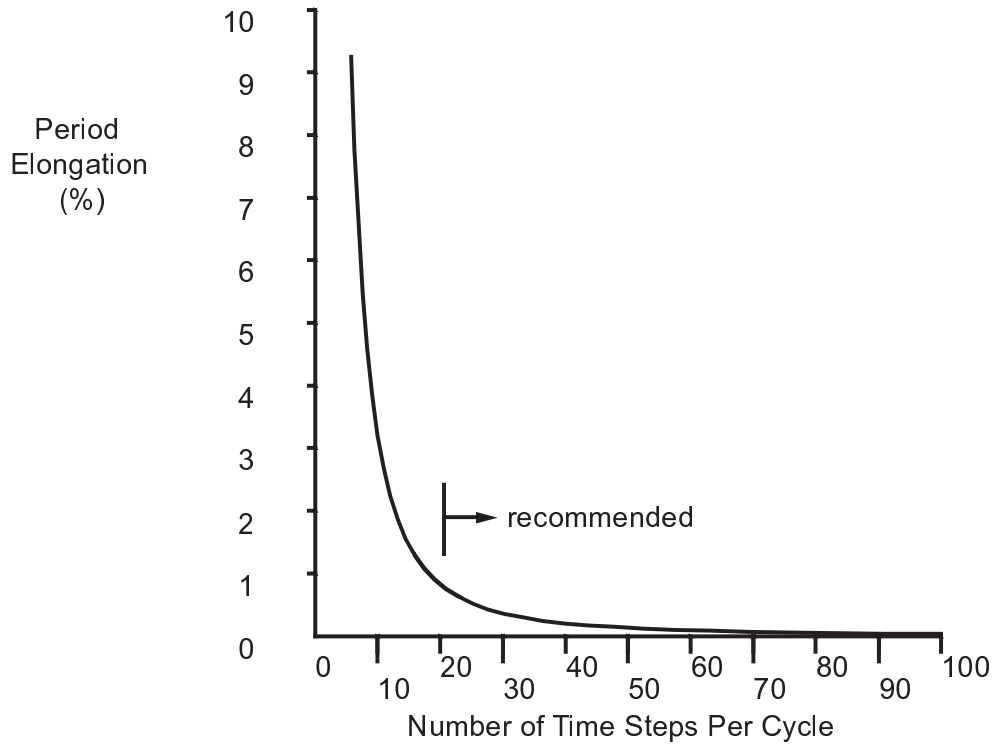
- Resolve the response frequency. The time step should be small enough to resolve the motion (response) of the structure. Since the dynamic response of a structure can be thought of as a combination of modes, the time step should be able to resolve the highest mode that contributes to the response. For the Newmark time integration scheme, it has been found that using approximately twenty points per cycle of the highest frequency of interest results in a reasonably accurate solution. That is, if  $f$  is the frequency (in cycles/time), the integration time step (ITS) is given by

$$\text{ITS} = 1/(20f)$$

Smaller ITS values may be required if acceleration results are needed.

The following figure shows the effect of ITS on the period elongation of a single-DOF spring-mass system. Notice that 20 or more points per cycle result in a period elongation of less than 1 percent.

**Figure 5.2: Effect of Integration Time Step on Period Elongation**

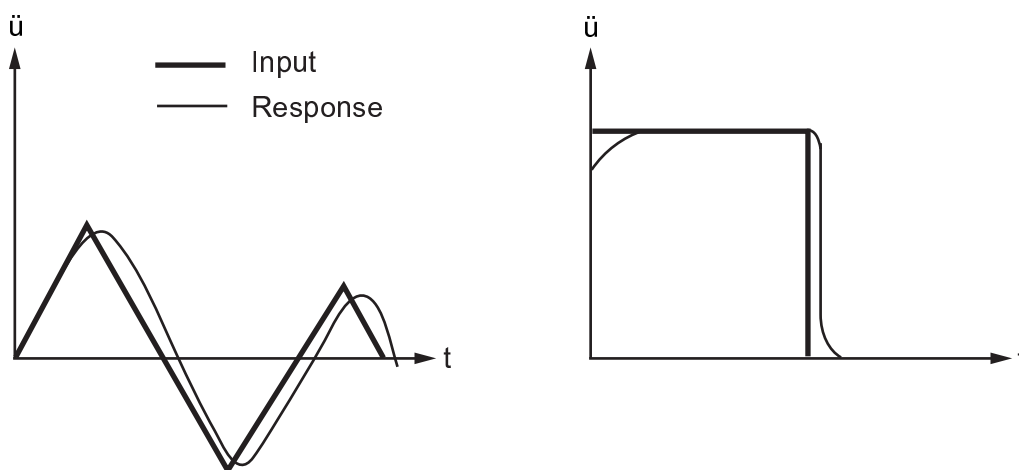


For the HHT time integration method, the same guidelines for time step should be applied. Note that if the same time step and time integration parameters are used, the HHT method will be more accurate compared to the Newmark method.

An alternative way to select time step size is to use the midstep residual criterion. When this criterion is used, the response frequency criterion is disabled by default. You have the option to enable the response frequency criterion along with the midstep residual criterion (see [item 6](#) below).

- Resolve the applied load-versus-time curve(s). The time step should be small enough to "follow" the loading function. The response tends to lag the applied loads, especially for stepped loads, as shown in [Figure 5.3: Transient Input vs. Transient Response \(p. 131\)](#). Stepped loads require a small ITS at the time of the step change so that the step change can be closely followed. ITS values as small as  $1/180f$  may be needed to follow stepped loads.

**Figure 5.3: Transient Input vs. Transient Response**



- Resolve the contact frequency. In problems involving contact (impact), the time step should be small enough to capture the momentum transfer between the two contacting surfaces. Otherwise, an apparent energy loss will occur and the impact will not be perfectly elastic. The integration time step can be determined from the contact frequency ( $f_c$ ) as:

$$\text{ITS} = 1/Nf_c \quad f_c = (1/2\pi)\sqrt{k/m}$$

where  $k$  is the gap stiffness,  $m$  is the effective mass acting at the gap, and  $N$  is the number of points per cycle. To minimize the energy loss, at least thirty points per cycle of ( $N = 30$ ) are needed. Larger values of  $N$  may be required if acceleration results are needed. For the mode-superposition method,  $N$  must be at least 7 to ensure stability.

You can use fewer than thirty points per cycle during impact if the contact period and contact mass are much less than the overall transient time and system mass, because the effect of any energy loss on the total response would be small.

- Resolve the wave propagation. If you are interested in wave propagation effects, the time step should be small enough to capture the wave as it travels through the elements. See [Build the Model \(p. 110\)](#) for a discussion of element size.
- Resolve the nonlinearities. For most nonlinear problems, a time step that satisfies the preceding guidelines is sufficient to resolve the nonlinearities. There are a few exceptions, however: if the structure tends to stiffen under the loading (for example, large deflection problems that change from bending to membrane load-carrying behavior), the higher frequency modes that are excited will have to be resolved.
- Satisfy the time step accuracy criterion. Satisfaction of the dynamics equations at the end of each time step ensures the equilibrium at these discrete points of time. The equilibrium at the intermediate time is usually not satisfied. If the time step is small enough, it can be expected that the intermediate state should not deviate too much from the equilibrium. On the other hand, if the time step is large, the intermediate state can be far from the equilibrium. The midstep residual norm provides a measure of the accuracy of the equilibrium for each time step. You can use the **MIDTOL** command to choose this criterion. See the **MIDTOL** command description for suggested tolerance values. See also [Midstep Residual for Structural Dynamic Analysis](#) in the *Mechanical APDL Theory Reference*.

After calculating the time step using the appropriate guidelines, use the minimum value for your analysis. By using automatic time stepping, you can let the program decide when to increase or decrease the time step during the solution. Automatic time stepping is discussed next.

---

### Caution

Avoid using exceedingly small time steps, especially when establishing initial conditions. Exceedingly small numbers can cause numerical difficulties. Based on a problem time scale of unity, for example, time steps smaller than  $10^{-10}$  could cause numerical difficulties.

---

## 5.6.2. Automatic Time Stepping

Automatic time stepping, also known as time step optimization, attempts to adjust the integration time step during solution based on the response frequency and on the effects of nonlinearities. The main benefit of this feature is that the total number of substeps can be reduced, resulting in computer resource savings. Also, the number of times that you might have to rerun the analysis (adjusting the time step size, nonlinearities, and so on) is greatly reduced. If nonlinearities are present, automatic time stepping

gives the added advantage of incrementing the loads appropriately and retreating to the previous converged solution (bisection) if convergence is not obtained. You can activate automatic time stepping with the **AUTOTS** command. (For more information on automatic time stepping in the context of nonlinearities, see [Nonlinear Structural Analysis](#) (p. 193).)

Although it seems like a good idea to activate automatic time stepping for all analyses, there are some cases where it may not be beneficial (and may even be harmful):

- Problems that have only localized dynamic behavior (for example, turbine blade and hub assemblies), where the low-frequency energy content of part of the system may dominate the high-frequency areas
- Problems that are constantly excited (for example, seismic loading), where the time step tends to change continually as different frequencies are excited
- Kinematics (rigid-body motion) problems, where the rigid-body contribution to the response frequency term may dominate

## 5.7. Where to Find Other Examples

Several ANSYS, Inc. publications, particularly the [Mechanical APDL Verification Manual](#), describe additional transient dynamic analyses.

The [Mechanical APDL Verification Manual](#) consists of test case analyses demonstrating the analysis capabilities of the program. While these test cases demonstrate solutions to realistic analysis problems, the [Mechanical APDL Verification Manual](#) does not present them as step-by-step examples with lengthy data input instructions and printouts. However, most users who have at least limited finite element experience should be able to fill in the missing details by reviewing each test case's finite element model and input data with accompanying comments.

The [Mechanical APDL Verification Manual](#) includes a variety of transient dynamic analysis test cases:

- VM9 - Large Lateral Deflection of Unequal Stiffness Springs
- VM40 - Large Deflection and Rotation of a Beam Pinned at One End
- VM65 - Transient Response of a Ball Impacting a Flexible Surface
- VM71 - Transient Response of a Spring, Mass, Damping System
- VM72 - Logarithmic Decrement
- VM73 - Free Vibration with Coulomb Damping
- VM74 - Transient Response to an Impulsive Excitation
- VM75 - Transient Response to a Step Excitation
- VM77 - Transient Response to a Constant Force with a Finite Rise Time
- VM79 - Transient Response of a Bilinear Spring Assembly
- VM80 - Plastic Response to a Suddenly Applied Constant Force
- VM81 - Transient Response of a Drop Container
- VM84 - Displacement Propagation along a Bar with Free Ends
- VM85 - Transient Displacements in a Suddenly Stopped Moving Bar
- VM91 - Large Rotation of a Swinging Pendulum
- VM156 - Natural Frequency of Nonlinear Spring-Mass System
- VM158 - Motion of a Bobbing Buoy
- VM179 - Dynamic Double Rotation of a Jointed Beam
- VM182 - Transient Response of a Spring-Mass System



---

## Chapter 6: Spectrum Analysis

---

A spectrum analysis is one in which the results of a modal analysis are used with a known *spectrum* to calculate displacements and stresses in the model. It is mainly used in place of a time-history analysis to determine the response of structures to random or time-dependent loading conditions such as earthquakes, wind loads, ocean wave loads, jet engine thrust, rocket motor vibrations, and so on.

The following spectrum analysis topics are available:

- 6.1. Understanding Spectrum Analysis
- 6.2. Single-Point Response Spectrum (SPRS) Analysis Process
- 6.3. Example Spectrum Analysis (GUI Method)
- 6.4. Example Spectrum Analysis (Command or Batch Method)
- 6.5. Where to Find Other Examples
- 6.6. Performing a Random Vibration (PSD) Analysis
- 6.7. DDAM Spectrum Analysis Process
- 6.8. Performing a Multi-Point Response Spectrum (MPRS) Analysis
- 6.9. Example Multi-Point Response Spectrum (MPRS) Analysis (Command or Batch Method)

### 6.1. Understanding Spectrum Analysis

The *spectrum* is a graph of spectral value versus frequency that captures the intensity and frequency content of time-history loads. Three types of spectra are available for a spectrum analysis:

- Response Spectrum
  - Single-Point Response Spectrum (SPRS)
  - Multi-Point Response Spectrum (MPRS)
- Dynamic Design Analysis Method (DDAM)
- Power Spectral Density (PSD)

SPRS is the only method available in the ANSYS Professional program.

#### 6.1.1. Response Spectrum

A *response spectrum* input represents the maximum *response* of single-DOF systems to a time-history loading function. It is a graph of response versus frequency, where the response might be displacement, velocity, acceleration, or force. Two types of response spectrum analysis are possible: single-point response spectrum and multi-point response spectrum. The output of a response spectrum analysis is the maximum response of each mode to the input spectrum. While the maximum response of each mode is known, the relative phase of each mode is unknown. To account for this, various mode combination methods are used (rather than simply summing these maximum modal responses).

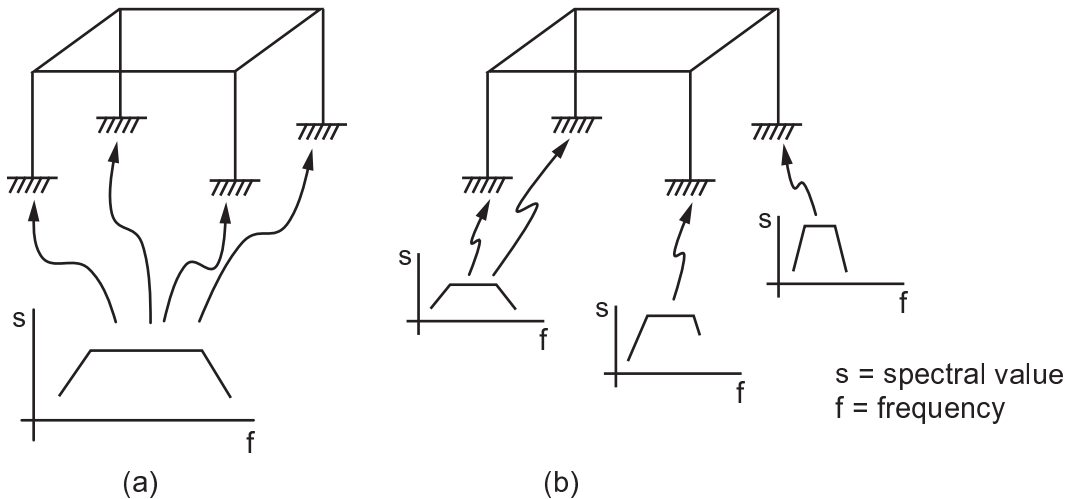
### 6.1.1.1. Single-Point Response Spectrum (SPRS)

In a *single-point response spectrum (SPRS)* analysis, you specify one response spectrum curve (or a family of curves) at a set of points in the model, such as at all supports, as shown in [Figure 6.1: Single-Point and Multi-Point Response Spectra \(p. 136\)](#) (a).

### 6.1.1.2. Multi-Point Response Spectrum (MPRS)

In a *multi-point response spectrum (MPRS)* analysis, you specify different spectrum curves at different sets of points, as shown in [Figure 6.1: Single-Point and Multi-Point Response Spectra \(p. 136\)](#) (b).

**Figure 6.1: Single-Point and Multi-Point Response Spectra**



### 6.1.2. Dynamic Design Analysis Method (DDAM)

The *Dynamic Design Analysis Method (DDAM)* is a technique used to evaluate the shock resistance of shipboard equipment. The technique is essentially a response spectrum analysis in which the spectrum is obtained from a series of empirical equations and shock design tables provided in the U.S. Naval Research Laboratory Report NRL-1396.

### 6.1.3. Power Spectral Density

Power spectral density (PSD) is a statistical measure defined as the limiting root mean-square (rms) value of a random variable. It is used in *random vibration analyses* in which the instantaneous magnitudes of the response can be specified only by probability distribution functions that show the probability of the magnitude taking a particular value. It is assumed that the dynamic input has a zero mean value and the range of values takes the form of a Gaussian or normal probability distribution.

A PSD is a graph of the PSD value versus frequency, where the PSD may be a displacement PSD, velocity PSD, acceleration PSD, or force PSD, that captures both the power or intensity of the input vibration and its frequency content. The PSD value is in  $(\text{unit})^2/\text{Hz}$ , such as  $g^2/\text{Hz}$ . Mathematically, the area under a PSD-versus-frequency curve is equal to the variance (square of the standard deviation) of the input vibration. Likewise, the output also takes on a Gaussian distribution and zero mean value. The output values of a PSD analysis are the response PSDs, with the area under the response PSD curve being the variance (the square of the standard deviation) of the response.

Similar to response spectrum analysis, a random vibration analysis may be single-point or multi-point. In a single-point random vibration analysis, you specify one PSD spectrum at a set of points in the



model. In a multi-point random vibration analysis, you specify different PSD spectra at different points in the model.

### 6.1.4. Deterministic vs. Probabilistic Analyses

Response spectrum and DDAM analyses are *deterministic* analyses because both the input to the analyses and output from the analyses are *actual* maximum values. Random vibration analysis, on the other hand, is *probabilistic* in nature, because both input and output quantities represent only the *probability* that they take on certain values.

## 6.2. Single-Point Response Spectrum (SPRS) Analysis Process

The general process for performing a single-point response spectrum analysis consists of four primary steps:

- [Step 1: Build the Model \(p. 137\)](#)
- [Step 2: Obtain the Modal Solution \(p. 137\)](#)
- [Step 3: Obtain the Spectrum Solution \(p. 138\)](#)
- [Step 4: Review the Results \(p. 143\)](#)

The process for running multiple spectrum analyses can be found in:

- [Running Multiple Spectrum Analyses \(p. 145\)](#)

If you wish to perform mode selection that includes and expands only the significant modes in a single-point response spectrum analysis, refer to [Mode Selection Based on the Mode Coefficients \(p. 74\)](#).

### 6.2.1. Step 1: Build the Model

See [Building the Model](#) in the *Basic Analysis Guide*. For further details, see the *Modeling and Meshing Guide*.

#### 6.2.1.1. Hints and Recommendations

- Only linear behavior is valid in a spectrum analysis. Nonlinear elements, if any, are treated as linear. If you include contact elements, for example, their stiffnesses are calculated based on their initial status and are never changed.
- Both Young's modulus (EX) (or stiffness in some form) and density (DENS) (or mass in some form) must be defined. Material properties can be linear, isotropic or orthotropic, and constant or temperature-dependent. Nonlinear properties, if any, are ignored.
- You can define damping using damping ratio, material damping and/or proportional damping. For more details about damping definition, see [Damping \(p. 2\)](#).

### 6.2.2. Step 2: Obtain the Modal Solution

The modal solution is required because the structure's mode shapes and frequencies must be available to calculate the spectrum solution.

The procedure for obtaining the modal solution is described in [Modal Analysis \(p. 39\)](#), but the following additional recommendations apply:

- Use the Block Lanczos, PCG Lanczos, Supernode, or Subspace method to extract the modes. (Other methods are not valid for subsequent spectrum analysis.)
- Extract a sufficient number of modes to characterize the structure's response in the frequency range of interest. The ratio of the effective mass (used in the subsequent spectrum analysis) to the total mass is printed out along with the participation factors at the end of the modal analysis. A ratio greater than 0.9 (more than 90% of the mass is included) is generally considered acceptable.
- To include the contribution of higher modes, you can either request the calculation of the residual vectors in the modal analysis (**RESVEC**) and use them in the spectrum analysis, or you can include the effect of the missing mass directly in the spectrum analysis (**MMASS** command).
- Expand all the modes (**MXPAND,ALL**). This is required if you eventually want to compute summed forces and moments in postprocessing (for example, obtaining the forces across a cut in the model using **FSUM**). You can also expand the modes after the spectrum solution if the size of the results file `Jobname.RST` is an issue (see the use of the *SIGNIF* argument in the **MXPAND** command documentation). In this case, use **MXPAND,-1** to suppress the expansion during the modal analysis. Expanding modes after a spectrum solution will only expand significant modes based on the excitation direction specified in spectrum analysis. If you included missing-mass effect (**MMASS**), then the calculation will be done only for these expanded modes. Note that you can limit the expanded mode set using the procedure in [Using Mode Selection \(p. 71\)](#).
- To include material-dependent damping in the spectrum analysis, specify it in the modal analysis.
- Constrain those degrees of freedom where you want to apply a base excitation spectrum.
- At the end of the solution, exit the Solution processor.

If you intend to perform multiple independent spectrum analyses, keep a copy of the `Jobname.MODE` file from the modal analysis. For more information, see [Running Multiple Spectrum Analyses \(p. 145\)](#).

### 6.2.3. Step 3: Obtain the Spectrum Solution

In this step, the program uses mode shapes extracted by the modal solution to calculate the SPRS solution. The following requirements and limitations apply:

- The mode shape file (`Jobname.MODE`) must be available.
- The database must contain the same model from which the modal solution was obtained.
- The `Jobname.FULL` file must be available for the participation factors calculation.
- If the missing mass calculation is activated (**MMASS** command), the element matrices file (`Jobname.EMAT`) and the results file (`Jobname.RST`) from the modal analysis must also be available.
- Spectrum reaction force with missing mass is not supported.

1. Enter SOLUTION.

**Command(s): /SOLU**

**GUI: Main Menu> Solution**

- Define the analysis type and analysis options. The program offers the following analysis options for a spectrum analysis. Not all modal analysis options and not all eigenvalue extraction techniques work with all spectrum analysis options.

**Table 6.1: Analysis Types and Options**

Option	Command	GUI Path
New Analysis	<b>ANTYPE</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; New Analysis</b>
Analysis Type: Spectrum	<b>ANTYPE</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; New Analysis&gt; Spectrum</b>
Spectrum Type: SPRS	<b>SPOPT</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; Analysis Options</b>
Number of Modes to Use for Solution	<b>SPOPT</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; Analysis Options</b>

- **Option: New Analysis [ANTYPE]**

Choose New Analysis.

- **Option: Analysis Type: Spectrum [ANTYPE]**

Choose analysis type spectrum.

- **Option: Spectrum Type: Single-point Response Spectrum [SPOPT]**

Choose Single-point Response Spectrum (SPRS).

- **Option: Number of Modes to Use for Solution [SPOPT]**

Choose enough modes to cover the frequency range spanned by the spectrum and to characterize the structure's response. The accuracy of the solution depends on the number of modes used: the larger the number, the higher the accuracy.

- Choose one of the mode combination methods.

Mechanical APDL offers six different mode combination methods for the single-point response spectrum analysis:

- Square Root of Sum of Squares (SRSS)
- Complete Quadratic Combination (CQC)
- Double Sum (DSUM)
- Grouping (GRP)
- Naval Research Laboratory Sum (NRLSUM)
- Rosenblueth (ROSE)

The NRLSUM method is typically used in the context of the Dynamic Design and Analysis Method (DDAM) spectrum analysis.

The following commands are used to invoke different methods of mode combinations:

**Command(s): SRSS, CQC, DSUM, GRP, NRLSUM, ROSE**

**GUI: Main Menu> Solution> Analysis Type> New Analysis> Spectrum**

**Main Menu> Solution> Analysis Type> Analysis Opts> Single-pt resp**

**Main Menu> Load Step Opts> Spectrum> Spectrum-Single Point-Mode Combine**

---

### Note

You must specify damping if you use the Complete Quadratic Combination method of mode combination (CQC).

---

These commands allow computation of three different types of responses:

- Displacement (label = *DISP*)

Displacement response refers to displacements, stresses, forces, etc.

- Velocity (label = *VELO*)

Velocity response refers to velocities, "stress velocities," "force velocities," etc.

- Acceleration (label = *ACEL*)

Acceleration response refers to accelerations, "stress accelerations," "force accelerations," etc.

These commands also allow you to specify the type of modal forces used in the combination. *ForceType*=STATIC (default) combines the modal static forces (i.e., stiffness multiplied by mode shape forces, both of which are stress-causing forces) while *ForceType*=TOTAL combines the summed modal static forces and inertia forces (i.e., stiffness and mass forces, both of which forces are seen by the supports).

The DSUM method also allows the input of time duration for earthquake or shock spectrum.

If the missing mass effect is included (**MMASS**), only displacement results are available (Label = DISP).

If the effect of the rigid responses is included (**RIGRESP**), the mode combination methods supported are SRSS, CQC and ROSE.

4. Specify load step options. The following options are available for single-point response spectrum analysis:

**Table 6.2: Load Step Options**

Option	Command	GUI Path
<b>Spectrum Options</b>		
Type of Response Spectrum	<b>SVTYP</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Spectrum&gt; Single Point&gt; Settings</b>
Excitation Direction	<b>SED</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Spectrum&gt; Single Point&gt; Settings</b>
Spectral-value- vs-frequency Curve	<b>FREQ, SV</b>	<b>Main Menu&gt; Solution&gt; Load Step Opts&gt; Spectrum&gt; Single Point&gt; Freq Table or Spectr Values</b>
<b>Damping (Dynamics Options)</b>		

Option	Command	GUI Path
Beta (Stiffness) Damping	<b>BETAD</b>	Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping
Alpha (Mass) Damping	<b>ALPHAD</b>	Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping
Constant Damping Ratio	<b>DMPRAT</b>	Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping
Modal Damping	<b>MDAMP</b>	Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping

### • Spectrum Options

These data include the following:

– *Type of Response Spectrum* [**SVTYP**]

The spectrum type can be displacement, velocity, acceleration, force, or PSD. All except the force spectrum represent seismic spectra; that is, they are assumed to be specified at the base. The force spectrum is specified at non-base nodes with the **F** or **FK** command, and the direction is implied by labels FX, FY, FZ. The PSD spectrum [**SVTYP**,4] is internally converted to a displacement response spectrum and is limited to flat, narrowband spectra; a more robust random vibration analysis procedure is described in [Performing a Random Vibration \(PSD\) Analysis](#) (p. 159).

– *Excitation Direction* [**SED**]

In addition, the **ROCK** command allows you to specify a rocking spectrum.

– *Spectral-Value-Versus-Frequency Curve* [**FREQ**, **SV**]

**SV** and **FREQ** commands are used to define the spectral curve with a maximum of 100 points. You can define a family of spectral curves, each curve for a different damping ratio. Use the **STAT** command to list current spectrum curve values, and the **SVPLOT** command to display the spectrum curves.

– *Missing Mass Effect* [**MMASS**]

The missing mass effect reduces the error caused when the higher modes are neglected in the analysis.

– *Rigid Responses Effect* [**RIGRESP**]

If rigid responses are included, the combination of modal responses with frequencies in the higher end of the spectrum frequency range will be more accurate.

– *Residual Vector* [**RESVEC**]

Just like the missing mass response, the residual vectors reduce the error caused when the higher modes are neglected in the analysis.

Unlike the missing mass response, the residual vector is calculated in the modal analysis and is considered as an additional mode. Hence, it is combined as such, and coupling may

exist between the residual vector and the modes depending on the mode combination method.

A frequency is associated with the residual vector. The input spectrum for this frequency should correspond to the Zero Period Acceleration value. The frequency is also used for the calculation of the velocity and acceleration solutions.

- **Damping (Dynamics Options)**

If you specify more than one form of damping, the program calculates an effective damping ratio at each frequency. The spectral value at this effective damping ratio is then calculated by log-log interpolation of the spectral curves. If no damping is specified, the spectral curve with the lowest damping is used. For more information about different forms of damping, see [Damping \(p. 2\)](#) in [Transient Dynamic Analysis \(p. 107\)](#).

The following forms of damping are available:

- *Beta (stiffness) Damping* [**BETAD**]

This option results in a frequency-dependent damping ratio.

- *Alpha (mass) Damping* [**ALPHAD**]

This option results in a frequency-dependent damping ratio.

- *Constant Damping Ratio* [**DMPRAT**]

This option specifies a constant damping ratio to be used at all frequencies.

- *Modal Damping* [**MDAMP**]

---

### Note

Material-dependent damping ratio (**MP,DMPR**) is also available, but only if specified in the modal analysis where an effective damping ratio is calculated based on the elements' strain energies. You must request that element results be calculated in the modal expansion (*Elcalc* = YES on the **MXPAND** command).

---

## 5. Start solution calculations.

**Command(s): SOLVE**

**GUI: Main Menu> Solution> Solve> Current LS**

The output from the solution includes the Response Spectrum Calculation Summary. This table, which is part of the printed output, lists the participation factors, mode coefficients (based on lowest damping ratio), and the mass distribution for each mode. To obtain the response of each mode (modal response), multiply the mode shape by the mode coefficient (based on lowest damping ratio). You do this by retrieving the mode coefficient with the **\*GET** command (Entity = MODE) and using it as a scale factor in the **SET** command.

The mode coefficients based on the actual damping are listed in the Significant Mode Coefficients (Including Damping) table.

- Repeat steps 4 and 5 for additional response spectra, if any. Note that solutions are not written to the `Jobname.rst` at this time.

The mode combination phase writes a file of POST1 commands (`Jobname.MCOM`). Read in this file in POST1 to do the mode combinations, using the results file (`Jobname.RST`) from the modal expansion pass.

The file `Jobname.MCOM` contains POST1 commands that combine the maximum modal responses by using the specified mode combination method to calculate the overall response of the structure.

The mode combination method determines how the structure's modal responses are to be combined:

- If you selected *displacement* as the response type (label = `DISP`), displacements and stresses are combined for each mode on the mode combination command.
- If you selected *velocity* as the response type (label = `VELO`), velocities and stress velocities are combined for each mode on the mode combination command.
- If you selected *acceleration* as the response type (label = `ACEL`), accelerations and stress accelerations are combined for each mode on the mode combination command.

- Leave the SOLUTION processor.

**Command(s): FINISH**

**GUI:** Close the Solution menu.

If needed, you can retrieve the frequencies, participation factors, mode coefficients and effective damping ratios with the **\*GET** command (Entity = MODE).

## 6.2.4. Step 4: Review the Results

Results from a single-point response spectrum analysis are written to the mode combination file, `Jobname.MCOM`, in the form of POST1 commands. These commands calculate the overall response of the structure by combining the maximum modal responses in some fashion (specified by one of the mode combination methods). The overall response consists of the overall displacements (or velocities or accelerations) and, if placed on the results file during the expansion pass, the overall stresses (or stress velocities or stress accelerations), strains (or strain velocities or strain accelerations), and reaction forces (or reaction force velocities or reaction force accelerations).

You can use POST1, the general postprocessor, to review the results.

---

### Note

If you want a direct combination of the derived stresses (S1, S2, S3, SEQV, SI) from the results file, issue the **SUMTYPE,PRIN** command before reading in the `Jobname.MCOM` file. With the PRIN option, component stresses are not available.

---

The command default (**SUMTYPE,COMP**) is used to directly operate only on the unaveraged element component stresses and compute the derived quantities from these. Refer to [Creating and Combining Load Cases](#) in the *Basic Analysis Guide*. Also, see the [Command Reference](#) for a description of the **SUMTYPE** command.

- Read the commands on `Jobname.MCOM`:

**Command(s): /INPUT**



**GUI: Utility Menu> File> Read Input From**

For example, issue **/INPUT** with the following arguments:

```
/INPUT,FILE,MCOM!Assumes the default jobname FILE
```

## 2. Display results:

• **Option: Display Deformed Shape**

**Command(s): PLDISP**

**GUI: Main Menu> General Postproc> Plot Results> Deformed Shape**

• **Option: Contour Displays**

**Command(s): PLNSOL or PLESOL**

**GUI: Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu or Element Solu**

Use **PLNSOL** or **PLESOL** to contour almost any result item, such as stresses (SX, SY, SZ ...), strains (EPELX, EPELY, EPELZ ...), and displacements (UX, UY, UZ ...). If you previously issued the **SUMTYPE** command, the results of the **PLNSOL** or **PLESOL** command are affected by the particular **SUMTYPE** command option (**SUMTYPE,COMP** or **SUMTYPE,PRIN**) that you selected.

Use the **PLETAB** command to contour element table data and **PLLS** to contour line element data.

Displacements, stresses, and strains are always in the element coordinate system (**RSYS,SOLU**).

Derived data, such as stresses and strains, are averaged at the nodes by the **PLNSOL** command. This averaging results in "smeared" values at nodes where elements of different materials, different shell thicknesses, or other discontinuities meet. To avoid the smearing effect, use selecting (described in [Selecting and Components](#) in the *Basic Analysis Guide*) to select elements of the same material, same shell thickness, etc. before issuing **PLNSOL**.

You can view correct membrane results for shells (**SHELL**, MID) by using **KEYOPT(8) = 2** (for **SHELL181**, **SHELL208**, **SHELL209**, **SHELL281**, and **ELBOW290**). These **KEYOPTS** write the mid-surface node results directly to the results file, and allow the membrane results to be directly operated on during squaring operations. The default method of averaging the TOP and BOT squared values to obtain a MID value can possibly yield incorrect MID values.

• **Option: Vector Displays**

**Command(s): PLVECT**

**GUI: Main Menu> General Postproc> Plot Results> Vector Plot> Predefined**

## 3. List results:

**Command(s): PRNSOL** (nodal results), **PRESOL** (element-by-element results), **PRRFOR** (reaction data), **FSUM**, **NFORCE**, **PRNLD** (nodal element forces sum)

**GUI: Main Menu> General Postproc> List Results> Nodal Solution**

**Main Menu> General Postproc> List Results> Element Solution**

**Main Menu> General Postproc> List Results> Reaction Solution**

**Note**

The element forces (**PRESOL**) and reaction forces (**PRRFOR**) are based on the type of force requested using *ForceType* on the combination commands (**SRSS**, **CQC**,



etc.). The same applies to the values retrieved using **\*GET,Par,ELEM,n,EFOR** and **\*GET,Par,NODE,n,RF**. The element nodal forces (**FSUM**, **PRNLD**, and **NFORCE** commands) are summed prior to the combination of those forces.

#### 4. Other Capabilities:

Many other postprocessing functions, such as mapping results onto a path, transforming results to different coordinate systems, and load case combinations, are available in POST1. See [The General Postprocessor \(POST1\)](#) in the *Basic Analysis Guide* for details.

### 6.2.5. Running Multiple Spectrum Analyses

When running multiple spectrum analyses, it is possible to compute the effects of multiple spectra independently, or to compute a velocity and/or an acceleration response in addition to a displacement response.

- To compute the effects of multiple spectra independently (for example, for three orthogonal directions for which you then combine in a separate step), repeat [Step 3: Obtain the Spectrum Solution \(p. 138\)](#). After the first spectrum analysis and for each subsequent analysis, you must activate the *modeReuseKey* on the **SPOPT** command so that the database and the needed files are ready for the new spectrum analysis.

*Reminder:* The existing `Jobname.MCOM` file is appended with the additional spectrum analyses.

- To compute a velocity and/or an acceleration response in addition to a displacement response, the mode combination can be performed in a separate solution phase.

In this case, you must save a copy (**/COPY**) of the `Jobname.MCOM` file from the spectrum analysis step. This ensures that the `Jobname.MCOM` files used for generating the downstream mode combinations use only the mode coefficients from that independent spectra input.

*Reminder:* The existing `Jobname.MCOM` file is also overwritten by the additional mode-combination step(s).

The procedure to run multiple spectrum analyses is as follows:

1. Enter SOLUTION.  
**Command(s): /SOLU**  
**GUI: Main Menu> Solution**
2. Define analysis type.  
**Command(s): ANTYPE**  
**GUI: Main Menu> Solution> Analysis Type> New Analysis**
  - **Option: New Analysis [ANTYPE]**  
Choose New Analysis.
  - **Option: Analysis Type [ANTYPE]**  
Choose analysis type spectrum.
3. Choose one of the mode combination methods.

The following commands are used to invoke different methods of mode combinations:

**Command(s):** **SRSS, CQC, DSUM, GRP, NRLSUM, ROSE**

**GUI:** Main Menu> Solution

Use the VELO or ACEL label on the mode-combination commands (**SRSS, CQC, DSUM, GRP, NRLSUM, ROSE**).

4. Start solution.

**Command(s):** **SOLVE**

**GUI:** Main Menu> Solution> Solve> Current LS

## 6.3. Example Spectrum Analysis (GUI Method)

In this example problem, you determine the seismic response of a beam structure. This problem is similar to *VM70* in the *Mechanical APDL Verification Manual*.

### 6.3.1. Problem Description

A simply supported beam of length  $\ell$ , mass per unit length  $m$ , and section properties shown in Problem Specifications, is subjected to a vertical motion of both supports. The motion is defined in terms of a seismic displacement response spectrum. Determine the nodal displacements, reactions forces, and the element solutions.

### 6.3.2. Problem Specifications

The following material properties are used for this problem:

$$E = 30 \times 10^6 \text{ psi}$$

$$m = 0.2 \text{ lb-sec}^2/\text{in}^2$$

The following geometric properties are used for this problem:

$$I = (1000/3) \text{ in}^4$$

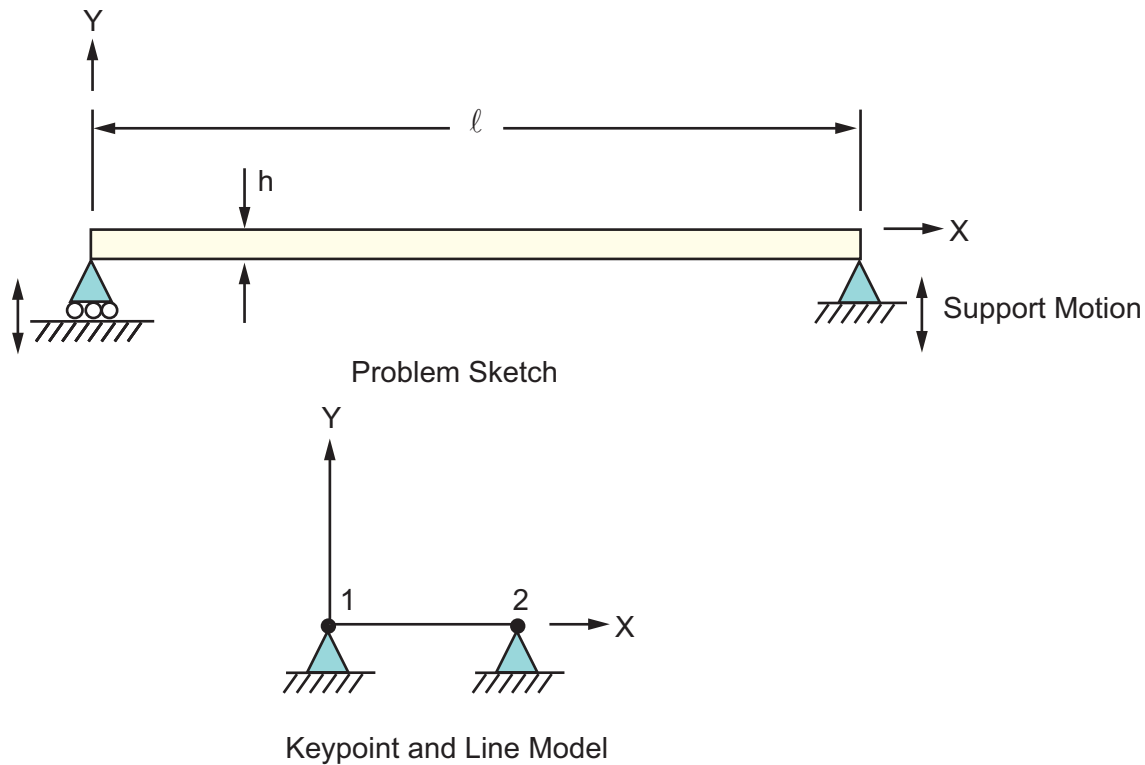
$$A = 273.9726 \text{ in}^2$$

$$\ell = 240 \text{ in}$$

$$h = 14 \text{ in}$$

### 6.3.3. Problem Sketch

Figure 6.2: Simply Supported Beam with Vertical Motion of Both Supports



Response Spectrum	
Frequency, Hz	Displacement, in.
0.1	0.44
10.0	0.44

### 6.3.4. Procedure

#### 6.3.4.1. Set the Analysis Title

1. Choose menu path **Utility Menu > File > Change Title**.
2. Type the text "Seismic Response of a Beam Structure" and click on OK.

#### 6.3.4.2. Define the Element Type

1. Choose menu path **Main Menu > Preprocessor > Element Type > Add/Edit/Delete**. The Element Types dialog box appears.
2. Click on Add. The Library of Element Types dialog box appears.
3. Scroll down the list on the left to "Structural Beam" and select it.
4. Click on "2 Node 188" in the list on the right.

5. Click on OK. The Library of Element Types dialog box closes.
6. Click on Options in the Element Types dialog box.
7. Choose Element Behavior K3 : Cubic Form. Click on OK.
8. Click on Close in the Element Types dialog box.

#### **6.3.4.3. Define the Cross-Section Area**

1. Choose menu path **Main Menu> Preprocessor> Sections> Beam> Common Sections**. The BeamTool dialog box appears.
2. Enter 71.6 for B and 3.82 for H. Click on Apply.
3. Click on Close to close the BeamTool dialog box.

#### **6.3.4.4. Define Material Properties**

1. Choose menu path **Main Menu> Preprocessor> Material Props> Material Models**. The Define Material Model Behavior dialog box appears.
2. In the Material Models Available window, click on the following options: Structural, Linear, Elastic, Isotropic. A dialog box appears.
3. Enter 30E6 for EX (Young's modulus), 0.30 for PRXY (Poisson's ratio), and then click on OK. Material Model Number 1 appears in the Material Models Defined window on the left.
4. Click on Density. A dialog box appears.
5. Enter 73E-5 for DENS (density), and click on OK.
6. Choose menu path **Material> Exit** to remove the Define Material Model Behavior dialog box.

#### **6.3.4.5. Define Keypoints and Line**

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Keypoints> In Active CS**. The Create Keypoints in Active Coordinate System dialog box appears.
2. Enter 1 for keypoint number.
3. Click on Apply to accept the default X, Y, Z coordinates of 0,0,0.
4. Enter 2 for keypoint number.
5. Enter 240,0,0 for X, Y, and Z coordinates, respectively.
6. Click on Apply.
7. Enter 3 for keypoint number.
8. Enter 0,1,0 for X, Y, and Z coordinates, respectively.
9. Click on OK.

10. Choose menu path **Utility Menu> PlotCtrls> Numbering**. The Plot Numbering Controls dialog box appears.
11. Click on "keypoint numbers" to turn keypoint numbering on.
12. Click on OK.
13. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Lines> Lines> Straight Line**. A picking menu appears.
14. Click on keypoint 1, and then on keypoint 2. A straight line appears between the two keypoints.
15. Click on OK. The picking menu closes.

#### **6.3.4.6. Set Global Element Density and Mesh Line**

1. Choose menu path **Main Menu > Preprocessor > Meshing > Size Cntrls > ManualSize > Global > Size**. The Global Element Sizes dialog box appears.
2. Enter 8 for the number of element divisions and click on OK. The Global Element Sizes dialog box closes.
3. Choose menu path **Main Menu> Preprocessor> Meshing> Mesh Attributes> All Lines**. The Line Attributes dialog box appears.
4. Click on Pick orientation keypoint(s) and click on OK. A picking menu appears.
5. In the graphic window, click once on the keypoint 3 and click on OK.
6. Choose menu path **Main Menu> Preprocessor> Meshing> Mesh> Lines**. A picking menu appears.
7. Click on Pick All. The picking menu closes.

#### **6.3.4.7. Set Boundary Conditions**

1. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On Nodes**. A picking menu appears.
2. In the graphics window, click once on the node at the left end of the beam.
3. Click on OK. The Apply U,ROT on Nodes dialog box appears.
4. In the scroll box of DOFs to be constrained, click once on "UY" to highlight it.
5. Click on OK.
6. Repeat steps 1-3 and select the node at the right end of the beam.
7. In the scroll box of DOFs to be constrained, click once on "UX." Both "UX" and "UY" should be highlighted.
8. Click on OK. The Apply U,ROT on Nodes dialog box closes.
9. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> Symmetry B.C.> On Nodes**. The Apply SYMM on Nodes box dialog appears.
10. In the scroll box for Norml symm surface is normal to, scroll to "z-axis" and click on OK.

### 6.3.4.8. Specify Analysis Type and Options

1. Choose menu path **Main Menu> Solution> Analysis Type> New Analysis**. The New Analysis dialog box appears.
2. Click on "Modal" to select it and click on OK. The New Analysis dialog box closes.
3. Choose menu path **Main Menu> Solution> Analysis Type> Analysis Options**. The Modal Analysis dialog box appears.
4. Click on "Block Lanczos" as the mode-extraction method [**MODOPT**].
5. Enter 3 for the number of modes to extract.
6. Enter 1 for the number of modes to expand [**MXPAND**].
7. Click on the Calculate elem. results dialog button [**MXPAND**] to specify YES.
8. Click on OK. The Modal Analysis dialog box closes, and the Block Lanczos Modal Analysis dialog box appears. Click OK to accept the default values.

### 6.3.4.9. Solve the Modal Analysis

1. Choose menu path **Main Menu> Solution> Solve> Current LS**. The Solve Current Load Step dialog box appears, along with a status window.
2. Carefully review the information in the status window, and then click on Close.
3. Click on OK on the Solve Current Load Step dialog box to start the solution.
4. When the solution is finished, a dialog box stating "Solution is done!" appears. Click on Close.

### 6.3.4.10. Set Up the Spectrum Analysis

1. Choose menu path **Main Menu> Solution> Analysis Type> New Analysis**. The New Analysis dialog box appears, along with a warning message that states: "Changing the analysis type is only valid within the first load step. Pressing OK will cause you to exit and reenter SOLUTION. This will reset load step count to 1." Click on CLOSE to close the warning message box.
2. Click on "Spectrum" to select it, and click on OK. The New Analysis dialog box closes.
3. Choose menu path **Main Menu> Solution> Load Step Opts> Spectrum> Single Point> Settings**. The Settings for Single-point Response Spectrum dialog box appears.
4. Select "Seismic displac" in the scroll box as the type of response spectrum.
5. Enter 0,1,0 for excitation direction into the excitation direction input windows and click on OK.

### 6.3.4.11. Define Spectrum Value vs. Frequency Table

1. Choose menu path **Main Menu> Solution> Load Step Opts> Spectrum> Single Point> Freq Table**. The Frequency Table dialog box appears.
2. Enter 0.1 for **FREQ1**, enter 10 for **FREQ2**, and click on OK.

3. Choose menu path **Main Menu> Solution> Load Step Opts> Spectrum> Single Point> Spectr Values**. The Spectrum Values - Damping Ratio dialog box appears.
4. Click on OK to accept the default of no damping. The Spectrum Values dialog box appears.
5. Enter 0.44 and 0.44 for FREQ1 and FREQ2, respectively.
6. Click on OK. The Spectrum Values dialog box closes.

#### **6.3.4.12. Select Mode Combination Method**

1. Choose menu path **Main Menu> Solution> Load Step Opts> Spectrum> Single Point> Mode Combine**. The Mode Combination Methods dialog box appears.
2. Select **SRSS** as the mode combination method.
3. Enter 0.15 for the significant threshold.
4. Select displacement for the type of output. Click OK. The Mode Combination Methods dialog box closes.

#### **6.3.4.13. Solve Spectrum Analysis**

1. Choose menu path **Main Menu> Solution> Solve> Current LS**. The Solve Current Load Step dialog box appears, along with a status window.
2. Carefully review the information in the status window, and then click on Close.
3. Click on OK on the Solve Current Load Step dialog box to start the solution.
4. When the solution is finished, a dialog box stating "Solution is done!" appears. Click on Close.

#### **6.3.4.14. Postprocessing: Print Out Nodal, Element, and Reaction Solutions**

1. Choose menu path **Main Menu > General Postproc > Results Summary**. The **SET** Command listing window appears.
2. Review the information in the listing window, and click on Close. The **SET** Command listing window closes.
3. Choose menu path **Utility Menu> File> Read Input From**. The Read File dialog box appears.
4. From the right side of the Read File dialog box, select the directory containing your results from the scroll box.
5. From the left side of the Read File dialog box, select the Jobname.MCOM file from the scroll box.
6. Click on OK. The Read File dialog box closes.
7. Issue a **PRNSOL**,DOF command.
8. Issue a **PRRSOL**,F command.

#### **6.3.4.15. Exit Mechanical APDL**

1. In the Mechanical APDL Toolbar, click on Quit.

2. Choose the save option you want and click on OK.

You have completed this example analysis.

## 6.4. Example Spectrum Analysis (Command or Batch Method)

You can perform the example spectrum analysis using the Mechanical APDL commands shown below instead of GUI choices. Items prefaced by an exclamation point (!) are comments.

### 6.4.1. Single Point Response Spectrum Analysis on a Beam Structure

```

/prep7
/title seismic response of a beam structure
et,1,beam188
keyopt,1,3,3
sectype,1,beam,rect
secdata,71.6,3.82
mp,ex,1,30e6
mp,prxy,1,0.30
mp,dens,1,73e-5
k,1
k,2,240
k,3,0,1,0          ! define orientation keypoint
l,1,2
esize,,8
latt,,,,,3        ! use orientation keypoint
lmesh,1
nselect,s,loc,x,0
dsym,symm,z
d,all,uy
nselect,s,loc,x,240
d,all,ux,,,,,uy
nselect,all
finish

/solu
antype,modal          ! mode-frequency analysis
modopt,lanb,3        ! block lanczos, extract first 3 modes
mxpand,1,,yes        ! expand first mode shape, calculate element stresses
outpr,basic,1
solve
finish

/solu
antype,spectr        ! spectrum analysis
spt,sprs            ! single point spectrum
sed,,1              ! global y-axis as spectrum direction
svtyp,3             ! seismic displacement spectrum
freq,.1,10          ! frequency points for sv vs. freq. table
sv,,.44,.44         ! spectrum values associated with frequency points
srss,0.15,dispr     ! square root of sum of squares mode combination
                    ! with signif=0.15 and displacement solution requested

solve
finish

/post1
set,list
/inp,,mcom

prnsol,dof          ! print nodal solution
prsol,f            ! print reaction solution
finish

```



## 6.4.2. Single Point Response Spectrum Analysis on a Piping Structure with Excitation along X, Y, and Z Directions

```

/prep7
et,1,pipe288,,3           ! PIPE288 using cubic shape function
et,2,mass21              ! MASS21

! *** Section and Real Constants
sectype,1,pipe
secdata,2.37500000,0.15400000,14
r,2,0.447000518e-01,0.447000518e-01,0.447000518e-01,0.0,0.0,0.0
r,3,0.447000518e-01,0.447000518e-01,0.447000518e-01,0.0,0.0,0.0
r,4,0.447000518e-01,0.447000518e-01,0.447000518e-01,0.0,0.0,0.0
r,5,0.447000518e-01,0.447000518e-01,0.447000518e-01,0.0,0.0,0.0
r,6,0.432699275e-01,0.432699275e-01,0.432699275e-01,0.0,0.0,0.0
r,7,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,8,0.432699275e-01,0.432699275e-01,0.432699275e-01,0.0,0.0,0.0
r,9,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,10,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,11,0.432699275e-01,0.432699275e-01,0.432699275e-01,0.0,0.0,0.0
r,12,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,13,0.432699275e-01,0.432699275e-01,0.432699275e-01,0.0,0.0,0.0
r,14,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,15,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0

! *** Nodes
n,1,0.0,-30.00
n,2,27.25,-30.00
n,3,27.25,-30.00,17.250
n,4,0.0,-30.00,17.250
n,5,0.0,18.625,17.250
n,6,0.0,18.625,8.625
n,7,0.0,18.625
n,8,8.625,18.625,
n,9,18.625,18.625
n,10,27.25,18.625
n,11,27.25,18.625,8.625
n,12,27.25,18.625,17.250
n,13,18.625,18.625,17.250
n,14,8.625,18.625,17.250
n,15,0.0,-80.00
n,16,27.25,-80.00
n,17,27.25,-80.00,17.250
n,18,0.0,-80.00,17.25

! *** Mesh of Straight Pipe: Tangent Elements
mat,1
type,1
secnum,1
en,1,15,1
en,2,1,7
en,3,7,6
en,4,6,5
en,5,5,4
en,6,4,18
en,7,16,2
en,8,2,10
en,9,10,11
en,10,11,12
en,11,12,3
en,12,3,17
en,13,12,13
en,14,13,14
en,15,14,5
en,16,7,8
en,17,8,9
en,18,9,10

! *** Mass Elements

```

```
mat,1
type,2
real,2
en,19,1
real,3
en,20,2
real,4
en,21,3
real,5
en,22,4
real,6
en,23,5
real,7
en,24,6
real,8
en,25,7
real,9
en,26,8
real,10
en,27,9
real,11
en,28,10
real,12
en,29,11
real,13
en,30,12
real,14
en,31,13
real,15
en,32,14

! *** Component for Support Nodes
nset,s,node,,15
nset,a,node,,16
nset,a,node,,17
nset,a,node,,18
cm,fixedsu,node
allsel,all

! *** Material
mp,ex,1,27899996.8
mp,nuxy,1,0.3
mp,dens,1,2.587991718e-10

! *** Constraints
d,15,all,0
d,16,all,0
d,17,all,0
d,18,all,0
allsel,all
finish

! *** Modal Analysis
! *****
/solution
antype,modal
modopt,lanb,5
mxpand,,,yes ! request the element results
solve
save ! save the database
finish

! ** Spectrum Analysis
! *****

/solution
antype,spectr
spopt,sprs,,,
dmpmat,0.02
grp,0.001
svtyp,2
```

```

sed,1                ! X-direction base excitation
freq,3.1,4,5,5.8,7.1,8.8,11,14.1,17.2
freq,35,40,588
sv,0.02,400,871,871,700,1188,1188,440,775,775
sv,0.02,380,348.6,145
solve
sed,0

sed,,1              ! Y-direction base excitation
freq
freq,3.1,4,5,5.8,7.1,8.8,11,14.1,17.2
freq,35,40,588
sv,0.02,266.7,580.7,580.7,466.7,792,792,293.3333,516.7,516.7
sv,0.02,253.3,232.4,96.7
solve
sed,,0

sed,,,1            ! Z-direction base excitation
freq
freq,3.1,4,5,5.8,7.1,8.8,11,14.1,17.2
freq,35,40,588
sv,0.02,400,871,871,700,1188,1188,440,775,775
sv,0.02,380,348.6,145
solve
sed,,,0
fini

/post1
/input,,mcom
prdi
finish

```

### 6.4.3. Single Point Response Spectrum Analysis on a Piping Structure with Excitation along X, Y, and Z Directions Separately by Reusing the Existing Mode File

```

/prep7
et,1,pipe288,,,3          ! PIPE288 using cubic shape function
et,2,mass21                ! MASS21

! *** Section and Real Constants
sectype,1,pipe
secdata,2.37500000,0.15400000,14
r,2,0.447000518e-01,0.447000518e-01,0.447000518e-01,0.0,0.0,0.0
r,3,0.447000518e-01,0.447000518e-01,0.447000518e-01,0.0,0.0,0.0
r,4,0.447000518e-01,0.447000518e-01,0.447000518e-01,0.0,0.0,0.0
r,5,0.447000518e-01,0.447000518e-01,0.447000518e-01,0.0,0.0,0.0
r,6,0.432699275e-01,0.432699275e-01,0.432699275e-01,0.0,0.0,0.0
r,7,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,8,0.432699275e-01,0.432699275e-01,0.432699275e-01,0.0,0.0,0.0
r,9,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,10,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,11,0.432699275e-01,0.432699275e-01,0.432699275e-01,0.0,0.0,0.0
r,12,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,13,0.432699275e-01,0.432699275e-01,0.432699275e-01,0.0,0.0,0.0
r,14,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0
r,15,0.893995859e-02,0.893995859e-02,0.893995859e-02,0.0,0.0,0.0

! *** Nodes
n,1,0.0,-30.00
n,2,27.25,-30.00
n,3,27.25,-30.00,17.250
n,4,0.0,-30.00,17.250
n,5,0.0,18.625,17.250
n,6,0.0,18.625,8.625
n,7,0.0,18.625
n,8,8.625,18.625,

```

```
n,9,18.625,18.625
n,10,27.25,18.625
n,11,27.25,18.625,8.625
n,12,27.25,18.625,17.250
n,13,18.625,18.625,17.250
n,14,8.625,18.625,17.250
n,15,0.0,-80.00
n,16,27.25,-80.00
n,17,27.25,-80.00,17.250
n,18,0.0,-80.00,17.25
```

```
! *** Mesh of Straight Pipe: Tangent Elements
```

```
mat,1
type,1
secnum,1
en,1,15,1
en,2,1,7
en,3,7,6
en,4,6,5
en,5,5,4
en,6,4,18
en,7,16,2
en,8,2,10
en,9,10,11
en,10,11,12
en,11,12,3
en,12,3,17
en,13,12,13
en,14,13,14
en,15,14,5
en,16,7,8
en,17,8,9
en,18,9,10
```

```
! *** Mass Elements
```

```
mat,1
type,2
real,2
en,19,1
real,3
en,20,2
real,4
en,21,3
real,5
en,22,4
real,6
en,23,5
real,7
en,24,6
real,8
en,25,7
real,9
en,26,8
real,10
en,27,9
real,11
en,28,10
real,12
en,29,11
real,13
en,30,12
real,14
en,31,13
real,15
en,32,14
```

```
! *** Component for Support Nodes
```

```
nselect,s,node,,15
nselect,a,node,,16
nselect,a,node,,17
nselect,a,node,,18
```

```
cm,fixedsu,node
allsel,all

! *** Material
mp,ex,1,27899996.8
mp,nuxy,1,0.3
mp,dens,1,2.587991718e-10

! *** Constraints
d,15,all,0
d,16,all,0
d,17,all,0
d,18,all,0
allsel,all
finish

! *** Modal Analysis
! *****
/solution
antype,modal
modopt,lanb,5
mxpand,,,yes ! request the element results
solve
save ! save the database for future reuse of MODE file
finish

/clear,nostart ! clear the database to start a new analysis

/com,***** X direction *****

/prep7
resume,,db ! resume the database from the modal analysis
finish

/solution
antype,spectr
spopt,sprs,,yes ! reuse the existing mode file
dmpmat,0.02
grp,0.001
svtyp,2

sed,1 ! X-direction base excitation
freq,3.1,4,5,5.8,7.1,8.8,11,14.1,17.2
freq,35,40,588
sv,0.02,400,871,871,700,1188,1188,440,775,775
sv,0.02,380,348.6,145
solve
sed,0
finish

/post1
/input,,mcom
prdi
finish
/copy,,mcom,,xdir,mcom
/clear,nostart

/com,***** Y direction *****

/prep7
resume,,db ! resume the database from the modal analysis
finish
```

```
/solution
antype,spectr
spopt,sprs,,,yes      ! reuse the existing mode file
dmpmat,0.02
grp,0.001
svtyp,2

sed,,1                ! Y-direction base excitation
freq
freq,3.1,4,5,5.8,7.1,8.8,11,14.1,17.2
freq,35,40,588
sv,0.02,266.7,580.7,580.7,466.7,792,792,293.3333,516.7,516.7
sv,0.02,253.3,232.4,96.7
solve
sed,,0
finish

/post1
/input,,mcom
prdi
finish
/copy,,mcom,,ydir,mcom
/clear,nostart

/com,***** Z direction *****

/prep7
resume,,db            ! resume the database from the modal analysis
finish

/solution
antype,spectr
spopt,sprs,,,yes      ! reuse the existing mode file
dmpmat,0.02
grp,0.001
svtyp,2

sed,,1                ! Z-direction excitation
freq
freq,3.1,4,5,5.8,7.1,8.8,11,14.1,17.2
freq,35,40,588
sv,0.02,400,871,871,700,1188,1188,440,775,775
sv,0.02,380,348.6,145
solve
sed,,0
finish

/post1
/input,,mcom
prdi
finish
/copy,,mcom,,zdir,mcom
```

## 6.5. Where to Find Other Examples

Several ANSYS publications, particularly the *Mechanical APDL Verification Manual*, describe additional spectrum analyses.

The *Mechanical APDL Verification Manual* consists of test case analyses demonstrating the analysis capabilities of the program. While these test cases demonstrate solutions to realistic analysis problems, the *Mechanical APDL Verification Manual* does not present them as step-by-step examples with lengthy data

input instructions and printouts. However, most ANSYS users who have at least limited finite element experience should be able to fill in the missing details by reviewing each test case's finite element model and input data with accompanying comments.

The *Mechanical APDL Verification Manual* includes a variety of spectrum analysis test cases:

- VM19 - Random Vibration Analysis of a Deep Simply-Supported Beam
- VM68 - PSD Response of a Two DOF Spring-Mass System
- VM69 - Seismic Response
- VM70 - Seismic Response of a Beam Structure
- VM203 - Dynamic Load Effect on Simply-Supported Thick Square Plate
- VM212 - DDAM Analysis of Foundation System (2-DOF System)

See the *Command Reference* for a discussion of the **ANTYPE**, **MODOPT**, **D**, **EXPASS**, **MXPAND**, **SPOPT**, **SVTYP**, **SED**, **FREQ**, **SV**, **SRSS**, **CQC**, **DSUM**, **GRP**, **NRLSUM**, **ROSE**, **MMASS**, **RIGRESP** and **DMPRAT** commands.

## 6.6. Performing a Random Vibration (PSD) Analysis

The procedure for a PSD analysis consists of the following steps:

1. Build the model.
2. Obtain the modal solution.
3. Obtain the spectrum solution.
4. Combine the modes.
5. Review the results.

Of these, the first two steps are the same as described for a single-point response spectrum analysis. The procedure for the remaining three steps is explained below. Random vibration analysis is not available in the ANSYS Professional program.

### 6.6.1. Obtain the PSD Solution

In this step, the program uses mode shapes extracted by the modal solution to calculate the PSD response. The following requirements apply:

- The mode shape file (`Jobname.MODE`) must be available.
- The database must contain the same model from which the modal solution was obtained.
- The `Jobname.FULL`, `.ESAV`, and `.EMAT` files must be available for the participation factors calculation.
- The results file (`Jobname.RST`) must be available for writing the static solutions (base excitation) and other solution types.
- The element modal load file (**Jobname.MLV**) must be available if load vectors were created (**MOD-CONT,ON**) and the element results were written on the **Jobname.MODE** file (`MSUPKEY = YES` on the **MXPAND** command) during the modal analysis.

1. Enter SOLUTION.

**Command(s): /SOLU**

**GUI: Main Menu> Solution**

2. Define the analysis type and analysis options:

- For spectrum type [**SPOPT**], choose Power Spectral Density (PSD).
- If you are interested in element results and reaction forces, specify YES for *ElcAlc* on the **SPOPT** command. Element results and reaction forces caused by the spectrum are calculated only if they were also requested during the modal expansion pass. Note that you must have asked for element results during the modal analysis as well (**MXPAND**)

3. Specify load step options. The following options are available for a random vibration analysis:

• **Spectrum Data**

- Type of PSD

**Command(s): PSDUNIT**

**GUI: Main Menu> Solution> Load Step Opts> Spectrum> PSD> Settings**

The PSD type can be displacement, velocity, force, pressure, or acceleration. Whether it is a base excitation or a nodal excitation is specified in Steps 4 and 5. If a pressure PSD is to be applied, the pressures should be applied in the modal analysis itself.

- PSD-versus-frequency table

Define a piecewise-linear (in log-log scale) PSD versus frequency table. Since a curve-fitting polynomial is used for the closed-form integration of the curve, you should graph the input, which is overlaid with the fitted curve to ensure a good fit. If the fit is not good, you should add one or more intermediate points to the table until you obtain a good fit. For a good fit, the PSD values between consecutive points should not change by more than an order of magnitude.

**Command(s): PSDFRQ, PSDVAL, PSDGRAPH**

**GUI: Main Menu> Solution> Load Step Opts> Spectrum> PSD> PSD vs Freq**

**Main Menu> Solution> Load Step Opts> Spectrum> PSD> Graph PSD Tables**

**PSDFRQ** and **PSDVAL** are used to define the PSD-versus-frequency table. Step 6 describes how to apply additional PSD excitations (if any). The maximum number of tables is 200.

You can issue **STAT** to list PSD tables and issue **PSDGRAPH** to graph them.

• **Damping (Dynamics Options)**

The following forms of damping are available: **ALPHAD**, **BETAD**, and **MDAMP** result in a frequency-dependent damping ratio, whereas **DMPRAT** specifies a constant damping ratio to be used at all



frequencies. If you specify more than one form of damping, Mechanical APDL calculates an effective damping ratio at each frequency.

---

### Note

If no damping is specified in a PSD analysis, a default **DMPRAT** of 1 percent is used.

---

### Note

Material-dependent damping ratio [**MP**,**DMPR**] is also available but only if specified in the modal analysis where an effective damping ratio is calculated based on the elements' strain energies.

---

- Alpha (Mass) Damping  
**Command(s): ALPHAD**  
**GUI: Main Menu > Solution > Load Step Opts > Time/Frequenc > Damping**
- Beta (Stiffness) Damping  
**Command(s): BETAD**  
**GUI: Main Menu > Solution > Load Step Opts > Time/Frequenc > Damping**
- Constant Damping Ratio  
**Command(s): DMPRAT**  
**GUI: Main Menu > Solution > Load Step Opts > Time/Frequenc > Damping**
- Frequency-Dependent Damping Ratio  
**Command(s): MDAMP**  
**GUI: Main Menu > Solution > Load Step Opts > Time/Frequenc > Damping**

- **Residual Vector [RESVEC]**

The residual vectors reduce the error caused when the higher modes are neglected in the analysis.

The remaining steps are specific to a random vibration analysis:

#### 4. Apply the PSD excitation.

- For base excitation, use the *UX*, *UY*, *UZ* labels and the *ROTX*, *ROTY*, *ROTZ* labels on the **D** (or **DK**, or **DL**, or **DA**) command. A value of 0.0 (or blank) can be used to remove a specification. Values other than 1.0 scale the participation factors.
- For uniform base motion using the **SED** command, specify *SEDX*, *SEDY*, or *SEDZ*. A value of 0.0 (or blank) removes a specification.
- For nodal excitation, use the *FX*, *FY*, *FZ* labels on the **F** (or **FK**) command. A value of 0.0 (or blank) can be used to remove a specification. Values other than 1.0 scale the participation factors.

- For pressure PSD excitation (where the pressure distribution was provided in the modal analysis), bring in the load vectors from the modal analysis (**LVSCALE**). You can use the scale factor to scale the participation factors.

---

### Note

You can apply base excitations only at nodes that were constrained in the modal analysis. If you applied the constraints using solid model constraints (**DK**), you must use the same solid model commands in defining the PSD excitation. Any loads applied during the preceding modal analysis must be removed by deleting or zeroing them.

---

**Command(s):** **D** (or **DK**, or **DL**, or **DA**) or **SED** for base excitation **F** (or **FK**) for nodal excitation **LVSCALE** for pressure PSD

**GUI: Main Menu> Solution> Define Loads> Apply> Structural> Spectrum> Base PSD Excit> On Nodes**

5. Begin participation factor calculations for the above PSD excitation.

Use the *TBLNO* field to indicate which PSD table to use, and *Excit* to specify whether the calculations are for a base or nodal excitation.

**Command(s):** **PFACT**

**GUI: Main Menu> Solution> Load Step Opts> Spectrum> PSD> Calculate PF**

6. If you need to apply multiple PSD excitations on the same model, repeat steps 3, 4, and 5 for each additional PSD table. Then define, as necessary, the degree of correlation between the excitations, using any of the following commands:

**Command(s):** **COVAL** for cospectral values, **QDVAL** for quadspectral values, **PSDSPL** for a spatial relationship, **PSDWAV** for a wave propagation relationship, **PSDGRAPH** to graph the data overlaid with the fitted curve

**GUI: Main Menu> Solution> Load Step Opts> Spectrum> PSD> Correlation**

**Main Menu> Solution> Load Step Opts> Spectrum> PSD> Graph Tables**

When you use the **PSDSPL** or **PSDWAV** command, you must use *SPATIAL* or *WAVE*, respectively, for *Parcor* on the **PFACT** command. **PSDSPL** and **PSDWAV** relationships might be quite CPU intensive for multi-point base excitations. Nodal excitation and base excitation input must be consistent when using **PSDWAV** and **PSDSPL** (for example, *FY* cannot be applied to one node and *FZ* be applied to another). The **PSDSPL** and **PSDWAV** commands are not available for a pressure PSD analysis.

7. Specify the output controls.

The only valid output control command for this analysis is **PSDRES**, which specifies the amount and form of output written to the results file. Up to three sets of solution quantities can be calculated: displacement solution, velocity solution, or acceleration solution. Each of these can be relative to the base or absolute.

**Command(s):** **PSDRES**

**GUI: Main Menu> Solution> Load Step Opts> Spectrum> PSD> Calc Controls**

Table 6.3: Solution Items Available in a PSD Analysis (p. 163) shows a summary of the possible solution sets. To limit the amount of data written to the results file, use **OUTRES** at the mode expansion step.

**Table 6.3: Solution Items Available in a PSD Analysis**

Solution	Items	Form
Displacement Solution (label <i>DISP</i> on <b>PSDRES</b> )	Displacements, stresses, strains, forces	Relative, absolute, or neither
Velocity Solution (label <i>VELO</i> on <b>PSDRES</b> )	Velocities, stress velocities, force velocities, etc.	Relative, absolute, or neither
Acceleration Solution (label <i>ACEL</i> on <b>PSDRES</b> )	Accelerations, stress accl's, force accl's, etc.	Relative, absolute, or neither

8. Start solution calculations.

**Command(s): SOLVE**

**GUI: Main Menu> Solution> Solve> Current LS**

9. Leave the SOLUTION processor.

**Command(s): FINISH**

**GUI: Close the Solution menu.**

## 6.6.2. Combine the Modes

The modes can be combined in a separate solution phase. A maximum of 10000 modes can be combined. The procedure is as follows:

1. Enter Solution.

**Command(s): /SOLU**

**GUI: Main Menu> Solution**

2. Define analysis type.

- **Option: New Analysis [ANTYPE]**

Choose New Analysis.

- **Option: Analysis Type: Spectrum [ANTYPE]**

Choose analysis type spectrum.

3. Only the PSD mode combination method is valid in a random vibration analysis. This method triggers calculation of the one-sigma ( $1\sigma$ , the standard deviation of the response, see [Review the Results \(p. 164\)](#) below) displacements, stresses, etc., in the structure. If you do not issue the **PSDCOM** command, the program does not calculate the one-sigma response of the structure. You can also specify the type of modal forces to be used in the combination. *ForceType*=STATIC (default) combines the modal static forces (i.e., stiffness multiplied by mode shape forces, both of which are stress-causing forces) while *ForceType*=TOTAL combines the summed modal static forces and inertia forces (i.e., stiffness and mass forces, both of which forces are seen by the supports).

**Command(s): PSDCOM**

**GUI: Main Menu> Solution> Load Step Opts> Spectrum> PSD> Mode Combin**

The *SIGNIF* and *COMODE* fields on the PSD mode combination method [**PSDCOM**] offer options to reduce the number of modes to be combined (see the description of **PSDCOM** command). If you want to exercise these options, it is prudent to print the modal covariance matrices in [Obtain the PSD Solution \(p. 159\)](#) to first investigate the relative contributions of the modes toward the final solution.

4. Start the solution.

**Command(s): SOLVE**

**GUI: Main Menu> Solution> Solve> Current LS**

5. Leave the SOLUTION processor.

**Command(s): FINISH**

**GUI: Close the Solution menu.**

---

### Note

You can run multiple PSD analyses without performing the modal analysis each time. To do so, you must activate *modeReuseKey* on the **SPOPT** command after the first PSD analysis and for each subsequent one so that the database and necessary files are ready for the new analysis.

---

## 6.6.3. Review the Results

Results from a random vibration analysis are written to the structural results file, *Jobname*.RST. They consist of the following quantities:

1. Expanded mode shapes from the modal analysis
2. Static solution for base excitation [**PFACT**,,BASE]
3. The following output, if mode combinations are requested [**PSDCOM**] and based on the **PSDRES** setting:
  - 1  $\sigma$  displacement solution (displacements, stresses, strains, and forces)
  - 1  $\sigma$  velocity solution (velocities, stress velocities, strain velocities, and force velocities)
  - 1  $\sigma$  acceleration solution (accelerations, stress accelerations, strain accelerations, and force accelerations)

1  $\sigma$  is the standard deviation of the response; that is, for any output value the expectation is that this value will not be exceeded 68.3% of the time.

Only component displacement, force, stress, and strain values are 1  $\sigma$  values and follow a Gaussian or normal distribution. Combined values (e.g. USUM, SI, SEQV, S1, etc.), or component values transformed into another coordinate system are not statistically meaningful, and they should be avoided. The exception is the SEQV value, for which a special algorithm is used to compute its value, such that multiplying it by 3 (the "3  $\sigma$ " rule) yields a good approximation to its upper bound, see [Equivalent Stress Mean Square Response](#) in the *Mechanical APDL Theory Reference*. SEQV is not computed for beam and pipe elements.

You can review these results in POST1, the general postprocessor, and then calculate response PSDs in POST26, the time-history postprocessor.

Postprocessing operations read your data from the results file. Only the solution data you SAVE will be available if you resume the database after a SOLVE.

### 6.6.3.1. Reviewing the Results in POST1

To review results in POST1, you first need to understand how the results data are organized on the results file. Table 6.4: Organization of Results Data from a PSD Analysis (p. 165) shows the organization.

#### Note

Load step 2 is left blank if you specify only nodal PSD excitation. Also, if you suppress the displacement, velocity, or acceleration solution using the **PSDRES** command, the corresponding load step is left blank. Also, the superelement displacement file (.DSUB) is not written for load steps 3, 4, or 5 in a PSD analysis.

**Table 6.4: Organization of Results Data from a PSD Analysis**

Load Step	Substep	Contents
1	1	Expanded modal solution for 1st mode
	2	Expanded modal solution for 2nd mode
	3	Expanded modal solution for 3rd mode
	Etc.	Etc.
2 (Base excit. only)	1	Unit static solution for PSD table 1
	2	Unit static solution for PSD table 2
	Etc.	Etc.
3	1	1 sigma displacement solution
4	1	1 sigma velocity solution (if requested)
5	1	1 sigma acceleration solution (if requested)

#### 6.6.3.1.1. Read the Desired Set of Results into the Database

For example, to read in the 1  $\sigma$  displacement solution, issue the command:

```
SET,3,1
```

You may use Fact on the **SET** command to multiply the result values to obtain, for example, the 2  $\sigma$  values using Fact=2 (the response will be less than these 2  $\sigma$  values 95.4% of the time), or use Fact=3 for the 3  $\sigma$  values (99.7% of the time).

**Command(s): SET**

**GUI: Main Menu > General Postproc > Read Results > First Set**

### 6.6.3.1.2. Display the Results

Use the same options available for the SPRS analysis.

---

#### Note

Nodal averaging performed by the **PLNSOL** command may not be appropriate in a random vibration analysis because the result values are not actual values but standard deviations. Instead, consider using the **PLESOL** command to display unaveraged element results.

---

#### Note

Displacements, stresses, and strains are always in the solution nodal or element coordinate system (**RSYS,SOLU**).

---

### 6.6.3.2. Calculating Response PSDs in POST26

You can calculate and display response PSDs for any results quantity available on the results file (displacements, velocities, and/or accelerations) if the `Jobname.RST` and `Jobname.PSD` files are available. If you are postprocessing in a new session, the `Jobname.DB` file corresponding to the PSD analysis solve must be available for resume.

The procedure to calculate the response PSD is as follows:

1. Enter POST26, the time-history postprocessor.

**Command(s):** **/POST26**

**GUI:** Main Menu > TimeHist PostPro

2. Store the frequency vector. *NPTS* is the number of frequency points to be added on either side of natural frequencies in order to "smooth" the frequency vector (defaults to 5). The frequency vector is stored as variable 1.

**Command(s):** **STORE,PSD,NPTS**

**GUI:** Main Menu > TimeHist Postpro > Store Data

3. Define the variables in which the result items of interest (displacements, stresses, reaction forces, etc.) are to be stored.

**Command(s):** **NSOL, ESOL,** and/or **RFORCE**

**GUI:** Main Menu > TimeHist Postpro > Define Variables

4. Calculate the response PSD and store it in the desired variable. The **PLVAR** command can then be used to plot the response PSD.

**Command(s):** **RPSD**

**GUI:** Main Menu > TimeHist Postpro > Calc Resp PSD

5. You can integrate the response PSD to obtain the variance and take its square root to obtain its 1  $\sigma$  value. For example:

```
RPSD,4,3,,3,2      ! variable 4 is the relative accel RPSD of var 3
INT1,5,4,1         ! variable 5 is the integral of the RPSD
```

```
*GET,VARIANCE,VARI,5,EXTREME,VLAST      ! get the integral value
STDDEV=SQRT(VARIANCE)                   ! convert to standard deviation (1-sigma)
```

### Note

This value will correspond to the POST1  $1\sigma$  values. POST26 sums all the modes by default for the response PSD (see *SIGNIF* on the **RPSD** command), whereas POST1 only sums the significant modes (*SIGNIF* equals 0.0001 by default for the **PSDCOM** command). You can use this comparison to verify that the significance factor on the **PSDCOM** command is small enough and that the curve fitting for the input PSD curve was adequate.

### 6.6.3.3. Calculating Covariance in POST26

You can compute the covariance between two quantities available on the results file (displacements, velocities, and/or accelerations), if the Jobname .RST and Jobname .PSD files are available.

The procedure to calculate the covariance between two quantities is as follows:

1. Enter POST26, the time-history postprocessor.  
**Command(s): /POST26**  
**GUI: Main Menu> TimeHist PostPro**
2. Define the variables in which the result items of interest (displacements, stresses, reaction forces, etc.) are to be stored.  
**Command(s): NSOL, ESOL, and/or RFORCE**  
**GUI: Main Menu> TimeHist Postpro> Define Variables**
3. Calculate the contributions of each response component (relative or absolute response) and store them in the desired variable. The **PLVAR** command can then be used to plot the modal contributions (relative response) followed by the contributions of pseudo-static and mixed part responses to the total covariance.  
**Command(s): CVAR**  
**GUI: Main Menu> TimeHist Postpro> Calc Covariance**
4. Obtain the covariance.  
**Command(s): \*GET,NameVARI,n,EXTREM,CVAR**  
**GUI: Utility Menu> Parameters> Get Scalar Data**

### 6.6.4. Sample Input

A sample input listing for a random vibration (PSD) analysis is shown below:

```
! Build the Model
/FILNAM,                               ! Jobname
/TITLE,                                 ! Title
/PREP7                                  ! Enter PREP7
...
...                                     ! Generate model
...
FINISH
!
! Obtain the Modal Solution
/SOLU                                   ! Enter SOLUTION
ANTYPE,MODAL                            ! Modal analysis
MODOPT,LANB                             ! Block Lanczos method
```

```

MXPAND, ...           ! Number of modes to expand, ...
D, ...               ! Constraints
SAVE
SOLVE                ! Initiates solution
FINISH
!
! Obtain the Spectrum Solution
/SOLU! Reenter SOLUTION
ANTYPE,SPECTR        ! Spectrum analysis
SPOPT,PSD, ...       ! Power Spectral Density; No. of modes;
                    ! Stress calcs. on/off
PSDUNIT, ...         ! Type of spectrum
PSDFRQ, ...          ! Frequency pts. (for spectrum values vs.
                    ! frequency tables)
PSDVAL, ...          ! Spectrum values
DMPRAT, ...          ! Damping ratio
D,0                  ! Base excitation
PFACT, ...           ! Calculate participation factors
PSDRES, ...          ! Output controls
SAVE
SOLVE
FINISH
!
! Combine modes using PSD method
/SOLU                ! Re-enter SOLUTION
ANTYPE,SPECTR        ! Spectrum analysis
PSDCOM,SIGNIF,COMODE ! PSD mode combinations with significance factor and
                    ! option for selecting a subset of modes for
                    ! combination

SOLVE
FINISH
!
! Review the Results
/POST1               ! Enter POST1
SET, ...             ! Read results from appropriate load step, substep
...! Postprocess as desired
...! (PLDISP; PLNSOL; NSORT; PRNSOL; etc.)
...
FINISH
!
! Calculate Response PSD
/POST26              ! Enter POST26
STORE,PSD            ! Store frequency vector (variable 1)
NSOL,2,...           ! Define variable 2 (nodal data)
RPSD,3,2,...         ! Calculate response PSD (variable 3)
PLVAR,3              ! Plot the response PSD
...
! Calculate Covariance
RESET                ! Reset all POST26 specifications to initial defaults.
NSOL,2               ! Define variable 2 (nodal data).
NSOL,3               ! Define variable 3 (nodal data).
CVAR,4,2,3,1,1      ! Calculate covariance between displacement
                    ! at nodes 2 and 3.
*GET,CVAR23U,VARI,4,EXREME,CVAR ! Obtain covariance.
FINISH

```

See the *Command Reference* for a discussion of the **ANTYPE**, **MODOPT**, **D**, **MXPAND**, **SPOPT**, **PSDUNIT**, **PSDFRQ**, **PSDVAL**, **DMPRAT**, **PFACT**, **PSDCOM**, **SUMTYPE**, and **PSDRES** commands.

## 6.7. DDAM Spectrum Analysis Process

The general process for performing a DDAM spectrum analysis consists of four primary steps:

- [Step 1: Build the Model \(p. 137\)](#)
- [Step 2: Obtain the Modal Solution \(p. 137\)](#)
- [Step 3: Obtain the Spectrum Solution \(p. 169\)](#)



- [Step 4: Review the Results \(p. 172\)](#)

If you wish to perform mode selection that includes and expands only the significant modes in a DDAM spectrum analysis, refer to [Mode Selection Based on DDAM Procedure \(p. 75\)](#).

The first two steps for a DDAM analysis are the same steps described for a modal analysis (see [Step 1: Build the Model \(p. 137\)](#) and [Step 2: Obtain the Modal Solution \(p. 137\)](#)). The procedure for the remaining steps is explained below:

### 6.7.1. Step 3: Obtain the Spectrum Solution

In this step, the program uses mode shapes extracted by the modal solution to calculate the DDAM solution. The following requirements apply:

- The mode shape file (Jobname .MODE) must be available.
  - The database must contain the same model from which the modal solution was obtained.
  - The Jobname .FULL file must be available for the participation factors calculation.
1. Enter SOLUTION.  
**Command(s): /SOLU**  
**GUI: Main Menu> Solution**
  2. Define the analysis type and analysis options. Mechanical APDL offers the following analysis options for a spectrum analysis. Not all modal analysis options and eigenvalue extraction techniques work with all spectrum analysis options.

**Table 6.5: Analysis Types and Options**

Option	Com- mand	GUI Path
New Analysis	<b>ANTYPE</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; New Analysis</b>
Analysis Type: Spectrum	<b>ANTYPE</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; New Analysis&gt; Spectrum</b>
Spectrum Type: DDAM	<b>SPOPT</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; Analysis Options</b>
Number of Modes to Use for Solution	<b>SPOPT</b>	<b>Main Menu&gt; Solution&gt; Analysis Type&gt; Analysis Options</b>

- **Option: New Analysis [ANTYPE]**  
Choose New Analysis.
- **Option: Analysis Type: Spectrum [ANTYPE]**  
Choose analysis type spectrum.
- **Option: Spectrum Type: Single-point Response Spectrum [SPOPT]**  
Choose Dynamic Design Analysis Method (DDAM).
- **Option: Number of Modes to Use for Solution [SPOPT]**

Choose a sufficient number of modes to characterize the structure's response and cover the frequency range spanned by the spectrum. The accuracy of the solution depends on the number of modes used. The larger the number of modes used, the higher the accuracy will be.

3. Specify load step options.

- The global excitation direction is specified using the **SED** command. Only excitation specified using **SED** is supported in a DDAM analysis.
- Use the **ADDAM** and **VDDAM** commands to specify the spectrum values and types. Based on the coefficients specified in the **ADDAM** and **VDDAM** commands, the program computes the mode coefficients according to the empirical equations given in the *Mechanical APDL Theory Reference*.
- No damping needs to be specified for solution because it is implied by the **ADDAM** and **VDDAM** commands. If damping is specified, it is used for mode combinations, but ignored for solution.

- Spectrum Options:

The following options are included:

- *Residual Vector* [**RESVEC**]

The residual vectors reduce the error caused when higher modes are neglected in the analysis.

Missing Mass and Rigid Response are not supported in a DDAM analysis.

4. Combine the modes. A maximum of 10,000 modes can be combined.

Choose one of the following mode combination methods offered for the single-point response spectrum analysis:

- Square Root of Sum of Squares (SRSS)
- Complete Quadratic Combination (CQC)
- Double Sum (DSUM)
- Grouping (GRP)
- Naval Research Laboratory Sum (NRLSUM)
- Rosenblueth (ROSE)

The NRLSUM method is typically used in the context of the Dynamic Design and Analysis Method (DDAM) spectrum analysis.

The *Closely Spaced Modes (CSM) Method* is also available using the **NRLSUM,,,CSM** command. It uses the NRL summation to process any out-of-phase modal responses with closed form treatment (NAVSEA [411]).

The following commands are used to invoke different methods of mode combinations:

**Command(s): SRSS, CQC, DSUM, GRP, NRLSUM, ROSE**

**GUI: Main Menu> Solution> Analysis Type> New Analysis> Spectrum**

**Main Menu> Solution> Analysis Type> Analysis Opts> D.D.A.M**

**Main Menu> Load Step Opts> Spectrum> Mode Combine**

These commands allow computation of three different types of responses:

- Displacement (label = *DISP*)

Displacement response refers to displacements, stresses, forces, etc.

- Velocity (label = *VELO*)

Velocity response refers to velocities, stress velocities, force velocities, etc.

- Acceleration (label = *ACEL*)

Acceleration response refers to accelerations, stress accelerations, force accelerations, etc.

These commands can also be used to specify the type of modal forces used in the combination. *ForceType*=*STATIC* (default) combines the modal static forces (i.e., stiffness multiplied by mode shape forces, both of which are stress-causing forces). *ForceType*=*TOTAL* combines the summed modal static forces and inertia forces (i.e., stiffness and mass forces, both of which forces are seen by the supports).

The **DSUM** method also allows the input of time duration for earthquake or shock spectrum.

The mode combination phase writes a file of POST1 commands (*Jobname.MCOM*). Read in *Jobname.MCOM* in POST1 to do the mode combinations using the results file (*Jobname.RST*) from the modal expansion pass.

The file *Jobname.MCOM* contains POST1 commands that combine the maximum modal responses by using the specified mode combination method to calculate the overall response of the structure.

The mode combination method determines how the structure's modal responses are to be combined:

- If *displacement* is selected as the response type (label = *DISP*), displacements and stresses are combined for each mode on the mode combination command.
- If *velocity* is selected as the response type (label = *VELO*), velocities and stress velocities are combined for each mode on the mode combination command.
- If *acceleration* is selected as the response type (label = *ACEL*), accelerations and stress accelerations are combined for each mode on the mode combination command.

---

### Note

A DDAM spectrum analysis may alternately be performed in two separate solves: first, the spectrum solution, and then the mode combination. The damping needed by the mode combination must be specified before the spectrum solve.

---

## 5. Start solution.

**Command(s): SOLVE**

**GUI: Main Menu > Solution > Solve > Current LS**

The output from the solution includes the Response Spectrum Calculation Summary. This table, which is part of the printed output, lists the participation factors, mode coefficients (based on lowest damping ratio), and the weight distribution for each mode. To obtain the response of each mode (modal response), multiply the mode shape by the mode coefficient (based on lowest damping ratio). You do this by retrieving the mode coefficient with the **\*GET** command (Entity = MODE) and using it as a scale factor in the **SET** command.

If needed, you can retrieve the frequencies, participation factors, mode coefficients and effective damping ratios with the **\*GET** command (Entity = MODE).

In addition to the above, the output from the solution also includes a DDAM Response Spectrum Calculation Summary. This table, which is part of the printed output, lists the shock design value and spectrum loading type for each mode.

6. Repeat [Step 3](#) to excite the structure along different directions if necessary. Note that solutions are not written to the `Jobname.RST` file at this time.

7. Leave the SOLUTION processor.

**Command(s): FINISH**

**GUI:** Close the Solution menu.

---

### Note

To compute the effect of multiple spectra independently, see [Running Multiple Spectrum Analyses](#) (p. 145).

---

## 6.7.2. Step 4: Review the Results

Results from a DDAM analysis are written to the mode combination file `Jobname.MCOM` in the form of POST1 commands. These commands calculate the overall response of the structure by combining the maximum modal responses in some fashion (specified by one of the mode combination methods). The overall response consists of the overall displacements (or velocities or accelerations). If element result calculations were requested in the expansion pass and stored in the results file, the overall response also includes the overall stresses (or stress velocities or stress accelerations), strains (or strain velocities or strain accelerations), and reaction forces (or reaction force velocities or reaction force accelerations).

You can use POST1, the general postprocessor, to review the results. See [Step 4: Review the Results](#) (p. 143) for details about the post-processing.

## 6.7.3. Sample Input

A sample input listing for a dynamics design analysis method (DDAM) spectrum analysis is shown below:

```
! Build the Model
/FILNAM,                ! Jobname
/TITLE,                  ! Title
/PREP7                   ! Enter PREP7
...
...                      ! Generate model
...
FINISH
!
! Obtain the Modal Solution
/SOLU                    ! Enter SOLUTION
ANTYPE, MODAL           ! Modal analysis
MODOPT, LANB            ! Block Lanczos method
```

```

MXPAND, ...           ! Number of modes to expand, ...
D, ...               ! Constraints
SOLVE                ! Initiates solution
SAVE
FINISH
!
! Obtain the Spectrum Solution
/SOLU                ! Reenter SOLUTION
ANTYPE, SPECTR       ! Spectrum analysis
SPOPT, DDAM, ...     ! Dynamic design analysis method

ADDAM, ...           ! Acceleration spectrum computation constants
VDDAM, ...           ! Velocity spectrum computation constants

DMPRAT, ...         ! Specify damping ratio required if CSM method is used
SED, ...             ! Specify excitation direction of shock spectrum
NRLS,,, CSM,        ! NRLSUM mode combinations with Closely Spaced Modes Method (CSM)

SOLVE                ! Initiates solution
FINISH
!

! Review the Results
/POST1               ! Enter the database results postprocessor
/INPUT,,MCOM         ! Read mode combination file
...! Postprocess as desired
...! (PRNSOL; PRRSOL; PRDI; PRNSOL; etc.)
...
FINISH

```

See the [Command Reference](#) for a discussion of the **ANTYPE**, **MODOPT**, **D**, **MXPAND**, **SPOPT**, **ADDAM**, **VDDAM**, **DMPRAT**, **SED**, and **NRLSUM** commands.

## 6.8. Performing a Multi-Point Response Spectrum (MPRS) Analysis

The procedure for a MPRS analysis consists of six steps:

1. Build the model.
2. Obtain the modal solution.
3. Obtain the spectrum solution.
4. Combine the modes.
5. Review the results.

The first two steps for an MPRS analysis are the same as the steps described for a modal analysis. The procedure for the remaining three steps is explained below.

---

### Note

MPRS analysis is not available in ANSYS Professional.

---

### 6.8.1. Step 4: Obtain the Spectrum Solution

In this step, the program uses mode shapes extracted by the modal solution to calculate the MPRS solution. The following requirements apply:

- The mode shape file (Jobname.MODE) must be available.

- The database must contain the same model from which the modal solution was obtained.
- The `Jobname.FULL`, `.ESAV`, and `.EMAT` files must be available for the participation factors calculation.
- The results file (`Jobname.RST`) must be available for writing the static solutions (base excitation) and missing mass solutions.

1. Enter SOLUTION.

**Command(s): /SOLU**

**GUI: Main Menu> Solution**

2. Define the analysis type and analysis options. For spectrum type (**SPOPT**), choose Multi-Point Response Spectrum (MPRS).

3. Specify the damping (Dynamics Options).

The following forms of damping are available: **BETAD**, **ALPHAD** and **MDAMP**. These commands result in a frequency-dependent damping ratio, whereas **DMPRAT** specifies a constant damping ratio to be used at all frequencies. If you specify more than one form of damping, ANSYS calculates an effective damping ratio at each frequency.

---

**Note**

Material-dependent damping ratio (**MP,DMPR**) is also available but only if specified in the modal analysis where an effective damping ratio is calculated based on the elements' strain energies.

---

- Beta (Stiffness) Damping

**Command(s): BETAD**

**GUI: Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping**

- Alpha (Mass) Damping

**Command(s): ALPHAD**

**GUI: Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping**

- Constant Damping Ratio

**Command(s): DMPRAT**

**GUI: Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping**

- Frequency-Dependent Damping Ratio

**Command(s): MDAMP**

**GUI: Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping**

4. Specify load step options. The following options are available:

- Spectrum Data

– Type of input spectrum

**Command(s): SPUNIT**

**GUI: This command can not be accessed from a menu.**

- The input spectrum type can be displacement, velocity, force, pressure, or acceleration. The type of excitation (base excitation or a nodal excitation) is specified in steps 4 and 5 of this procedure. If a pressure spectrum is to be applied, the pressures should be applied in the modal analysis.
- Spectrum value-versus-frequency table

Define the points of each spectrum curve. You can define a family of spectrum curves; each curve is associated with a damping ratio.

**Command(s):** **SPFREQ**, **SPVAL**, **SPDAMP**, **SPGRAPH**

**GUI:** These commands can not be accessed from a menu.

You can issue **STAT** to list the tables and **SPGRAPH** to display them.

- Missing mass and rigid responses

- Missing Mass Effect [**MMASS**]

The missing mass effect reduces the error caused when the higher modes are neglected in the analysis.

- Rigid Responses Effect [**RIGRESP**]

If rigid responses are included, the combination of modal responses with frequencies in the higher end of the spectrum frequency range will be more accurate.

- Residual Vector [**RESVEC**]

Just like the missing mass response, the residual vectors reduce the error caused when the higher modes are neglected in the analysis.

Unlike the missing mass response, the residual vector is calculated in the modal analysis and is considered as an additional mode. Hence, it is combined as such, and coupling may exist between the residual vector and the modes depending on the mode combination method.

A frequency is associated with the residual vector. The input spectrum for this frequency should correspond to the Zero Period Acceleration value. The frequency is also used for the calculation of the velocity and acceleration solutions.

## 5. Apply the excitation.

- For base excitation, use the *UX*, *UY*, *UZ* and the *ROTX*, *ROTY*, *ROTZ* labels on the **D** (or **DK**, or **DL**, or **DA**) command. A value of 0.0 (or blank) removes a specification. Values other than 1.0 scale the participation factors.
- For uniform base motion using the **SED** command, specify *SEDX*, *SEDY*, or *SEDZ*. A value of 0.0 (or blank) removes a specification.
- For nodal excitation, use the *FX*, *FY*, *FZ* on the **F** (or **FK**) command. A value of 0.0 (or blank) removes a specification. Values other than 1.0 scale the participation factors.

- For pressure excitation (where the pressure distribution was provided in the modal analysis), bring in the load vectors from the modal analysis (**LVSCALE**). You can use the scale factor to scale the participation factors.

---

### Note

You can apply base excitations only at nodes that were constrained in the modal analysis. If you applied the constraints using solid model constraints (**DK**), you must use the same solid model commands in defining the MPRS excitation. Any loads applied during the preceding modal analysis must be removed by deleting or zeroing them.

---

**Command(s):** **D** (or **DK**, or, **DL**, or **DA**), or **SED** for base excitation **F** (or **FK**) for nodal excitation **LVSCALE** for pressure excitation

**GUI: Main Menu> Solution> Define Loads> Apply> Structural> Spectrum> Base PSD Excit> On Nodes**

6. Begin participation factor calculations for the above MPRS excitation.

Use the *TBLNO* field to indicate which spectrum table to use, and *Excit* to specify whether the calculations are for a base or nodal excitation.

**Command(s):** **PFACT**

**GUI: Main Menu> Solution> Load Step Opts> Spectrum> PSD> Calculate PF**

If you need to apply multiple MPRS excitations on the same model, repeat steps 4, 5, and 6 above for each additional spectrum table.

## 6.8.2. Step 5: Combine the Modes

This step is the same as [step 3](#) described in [Single-Point Response Spectrum \(SPRS\) Analysis Process \(p. 137\)](#); however, the absolute sum method (*AbsSumKey*=yes on the **SRSS** command) acts as an additional combination method.

---

### Note

You can run multiple MPRS analyses without performing the modal analysis each time. To do so, you must activate the *modeReuseKey* on the **SPOPT** command after the first MPRS analysis and for each subsequent one so that the database as well as the necessary files are ready for the new analysis.

---

## 6.8.3. Step 6: Review the Results

This step is the same as [step 6](#) described in [Single-Point Response Spectrum \(SPRS\) Analysis Process \(p. 137\)](#).

Intermediate results from a MPRS analysis are written to the structural results file, *Jobname*.RST. They consist of the following quantities:

1. Expanded mode shapes from the modal analysis (loadstep 1)
2. Static solutions for base excitation (**PFACT**„BASE) (loadstep 2)
3. Missing mass responses if requested (**MMASS**) (loadstep 3)



## 6.9. Example Multi-Point Response Spectrum (MPRS) Analysis (Command or Batch Method)

In this example problem, you determine the seismic response of a three-beam frame using Mechanical APDL commands.

### 6.9.1. Problem Description

A three-beam frame is subjected to vertical motion of both supports. The motion is defined in terms of seismic acceleration response spectra. A multi-point response spectrum analysis is performed to determine the nodal displacements.

### 6.9.2. Problem Specifications

The following material properties are used for this problem:

Young's modulus =  $1e7$  psi

Density =  $3e-4$  lb/in<sup>3</sup>

The following geometric properties are used for this problem:

Cross-sectional area =  $.1$  in<sup>2</sup>

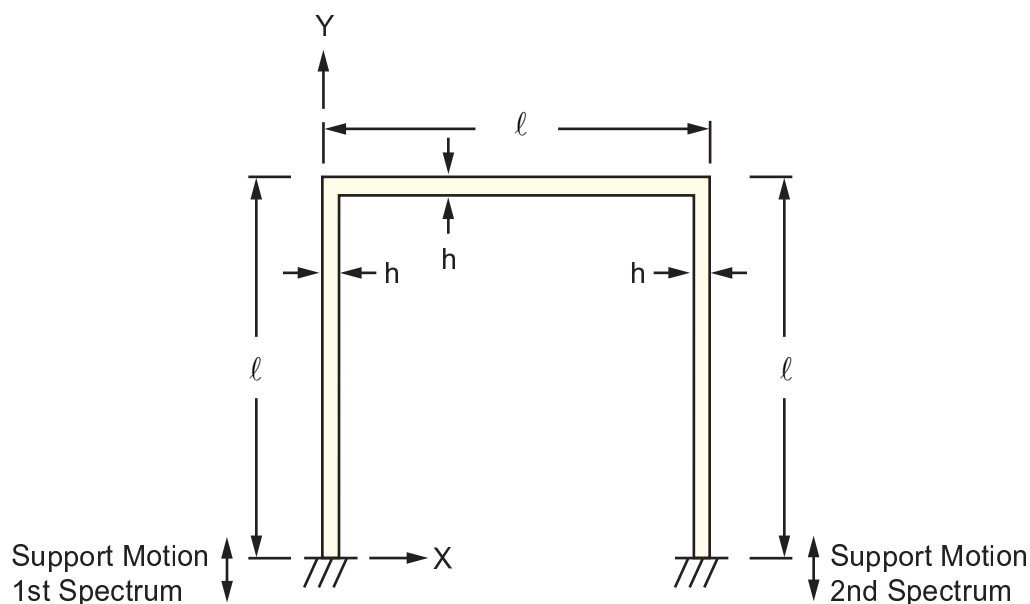
Area moment of inertia =  $.001$  in<sup>4</sup>

Beam height =  $.1$  in

Beam length =  $100$  in

### 6.9.3. Problem Sketch

Figure 6.3: Three-Beam Frame



### 6.9.4. Command Listing

Items prefaced by an exclamation point (!) are comments.

```
/prep7
et,1,188
keyopt,1,3,3
sectype,1,beam,rect
secdata,0.1,1
!r,1,.1,.001,.1
mp,ex,1,1e7
mp,nuxy,1,.3
mp,dens,1,.0003
k,1
k,2, ,100
k,3,100,100
k,4,100
l,1,2
l,2,3
l,3,4
esize,,10
lmesh,all
d,all,uz,0
d,all,rotx,0
d,all,roty,0
d,node(0,0,0),all
d,node(100,0,0),all
fini

/solu
antype,modal
modop,lanb,2 ! Lanczos eigensolver, request 2 modes
mxpand,2 ! Expand 2 modes
solve
fini

/solu
antype,spectrum ! Spectrum analysis
spopt,mprs ! Multi-point response (use all extracted modes by default)

!***** Spectrum #1 *****
spunit,1,accg ! Define the type of 1st spectrum (acceleration)
spfreq,1,1.0,100.0 ! Define the frequency range [1,100]Hz of 1st spectrum
spval,1,,1.0,1.0 ! Define acceleration values of 1st spectrum
d,node(0,0,0),uy,1.0 ! Define constraint
pfact,1 ! Calculate participation factors of 1st spectrum
! (base excitation by default)

!***** Spectrum #2 *****
spunit,2,accg ! Define the type of 2nd spectrum (acceleration)
spfreq,2,1.0,100.0 ! Define the frequency range [1,100]Hz of 2nd spectrum
spval,2,,0.8,0.8 ! Define acceleration values of 2nd spectrum
d,node(0,0,0),uy,0 ! Remove previous constraint
d,node(100,0,0),uy,1.0 ! Define new constraint
pfact,2 ! Calculate participation factors of 2nd spectrum
! (base excitation by default)
srss ! Combine using SRSS (displacement solution by default)
solve
fini

/post1
/inp,,mcom ! Input the mode combination file to perform the
! combination of displacement solutions
prns,u,y ! Printout displacement uy
finish
```

---

## Chapter 7: Buckling Analysis

---

Buckling analysis is a technique used to determine *buckling loads* (critical loads at which a structure becomes unstable) and *buckled mode shapes* (the characteristic shape associated with a structure's buckled response).

The following buckling analysis topics are available:

- 7.1. Types of Buckling Analyses
- 7.2. Commands Used in a Buckling Analysis
- 7.3. Performing a Nonlinear Buckling Analysis
- 7.4. Performing a Post-Buckling Analysis
- 7.5. Procedure for Eigenvalue Buckling Analysis
- 7.6. Sample Buckling Analysis (GUI Method)
- 7.7. Sample Buckling Analysis (Command or Batch Method)
- 7.8. Where to Find Other Examples

### 7.1. Types of Buckling Analyses

Two techniques are available in the ANSYS Multiphysics, ANSYS Mechanical, ANSYS Structural, and ANSYS Professional programs for predicting the buckling load and buckling mode shape of a structure: *non-linear* buckling analysis, and *eigenvalue* (or linear) buckling analysis. Because the two methods can yield dramatically different results, it is necessary to first understand the differences between them.

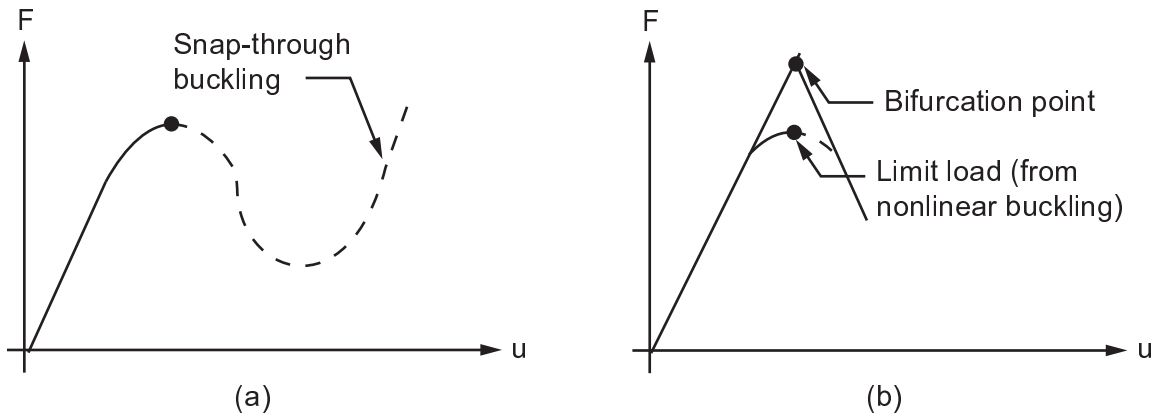
#### 7.1.1. Nonlinear Buckling Analysis

*Nonlinear buckling analysis* is usually the more accurate approach and is therefore recommended for design or evaluation of actual structures. This technique employs a nonlinear static analysis with gradually increasing loads to seek the load level at which your structure becomes unstable, as depicted in [Figure 7.1: Buckling Curves \(p. 180\)](#) (a).

Using the nonlinear technique, your model can include features such as initial imperfections, plastic behavior, gaps, and large-deflection response. In addition, using deflection-controlled loading, you can even track the post-buckled performance of your structure (which can be useful in cases where the structure buckles into a stable configuration, such as "snap-through" buckling of a shallow dome).

#### 7.1.2. Eigenvalue Buckling Analysis

*Eigenvalue buckling analysis* predicts the theoretical buckling strength (the bifurcation point) of an ideal linear elastic structure. (See [Figure 7.1: Buckling Curves \(p. 180\)](#) (b).) This method corresponds to the textbook approach to elastic buckling analysis: for instance, an eigenvalue buckling analysis of a column will match the classical Euler solution. However, imperfections and nonlinearities prevent most real-world structures from achieving their theoretical elastic buckling strength. Thus, eigenvalue buckling analysis often yields unconservative results, and should generally not be used in actual day-to-day engineering analyses.

**Figure 7.1: Buckling Curves**

(a) Nonlinear load-deflection curve, (b) Linear (Eigenvalue) buckling curve

## 7.2. Commands Used in a Buckling Analysis

You use the same set of commands to build a model and perform a buckling analysis that you use to do any other type of finite element analysis. Likewise, you choose similar options from the graphical user interface (GUI) to build and solve models no matter what type of analysis you are doing.

[Sample Buckling Analysis \(GUI Method\) \(p. 186\)](#) and [Sample Buckling Analysis \(Command or Batch Method\) \(p. 191\)](#) show you how to perform an example eigenvalue buckling analysis via the GUI or via commands, respectively. For detailed, alphabetized descriptions of the ANSYS commands, see the [Command Reference](#).

## 7.3. Performing a Nonlinear Buckling Analysis

A nonlinear buckling analysis is a static analysis with large deflection active (**NLGEOM,ON**), extended to a point where the structure reaches its limit load or maximum load. Other nonlinearities such as plasticity may be included in the analysis. The procedure for a static analysis is described in [Structural Static Analysis \(p. 9\)](#), and nonlinearities are described in [Nonlinear Structural Analysis \(p. 193\)](#).

### 7.3.1. Applying Load Increments

The basic approach in a nonlinear buckling analysis is to constantly increment the applied loads until the solution begins to diverge. Be sure to use a sufficiently fine load increment as your loads approach the expected critical buckling load. If the load increment is too coarse, the buckling load predicted may not be accurate. Turn on bisection and automatic time stepping (**AUTOTS,ON**) to help avoid this problem.

### 7.3.2. Automatic Time Stepping

With automatic time stepping on, the program *automatically* seeks out the buckling load. If automatic time stepping is ON in a static analysis having ramped loading and the solution does not converge at a given load, the program bisects the load step increment and attempts a new solution at a smaller load. In a buckling analysis, each such convergence failure is typically accompanied by a "negative pivot" message indicating that the attempted load equals or exceeds the buckling load. You can usually ignore these messages if the program successfully obtains a converged solution at the next, reduced load. The program normally converges to the limiting load as the process of bisection and resolution continues.

to the point at which the minimum time step increment (specified by **DELTIM** or **NSUBST**) is achieved. The minimum time step will directly affect the precision of your results.

### 7.3.3. Unconverged Solution

An unconverged solution does not necessarily mean that the structure has reached its maximum load. It could also be caused by numerical instability, which might be corrected by refining your modeling technique. Track the load-deflection history of your structure's response to decide whether an unconverged load step represents actual structural buckling, or whether it reflects some other problem. Perform a preliminary analysis using the arc-length method (**ARCLEN**) to predict an approximate value of buckling load. Compare this approximate value to the more precise value calculated using bisection to help determine if the structure has indeed reached its maximum load. You can also use the arc-length method itself to obtain a precise buckling load, but this method requires you to adjust the arc-length radius by trial-and-error in a series of manually directed reanalyses.

### 7.3.4. Hints and Tips for Performing a Nonlinear Buckling Analysis

If the loading on the structure is perfectly in-plane (that is, membrane or axial stresses only), the out-of-plane deflections necessary to initiate buckling will not develop, and the analysis will fail to predict buckling behavior. To overcome this problem, apply a small out-of-plane perturbation, such as a modest temporary force or specified displacement, to begin the buckling response. (A preliminary eigenvalue buckling analysis of your structure may be useful as a predictor of the buckling mode shape, allowing you to choose appropriate locations for applying perturbations to stimulate the desired buckling response.) The imperfection (perturbation) induced should match the location and size of that in the real structure. The failure load is very sensitive to these parameters.

Consider these additional hints and tips as you perform a nonlinear buckling analysis:

- Forces (and displacements) maintain their original orientation, but surface loads will "follow" the changing geometry of the structure as it deflects. Therefore, be sure to apply the proper type of loads.
- Carry your stability analysis through to the point of identifying the critical load in order to calculate the structure's factor of safety with respect to nonlinear buckling. Merely establishing the fact that a structure is stable at a given load level is generally insufficient for most design practice; you will usually be required to provide a specified safety factor, which can only be determined by establishing the actual limit load.
- For those elements that support the consistent tangent stiffness matrix, activate the consistent tangent stiffness matrix (**KEYOPT(2) = 1** and **NLGEOM,ON**) to enhance the convergence behavior of your nonlinear buckling analyses and improve the accuracy of your results. This element **KEYOPT** must be defined before the first load step of the solution and cannot be changed once the solution has started.
- Many other elements (such as **BEAM188**, **BEAM189**, **SHELL181**, **REINF264**, **SHELL281**, and **ELBOW290**) provide consistent tangent stiffness matrix with **NLGEOM,ON**.

## 7.4. Performing a Post-Buckling Analysis

A post-buckling analysis is a continuation of a [nonlinear buckling analysis](#). After a load reaches its buckling value, the load value may remain unchanged or it may decrease, while the deformation continues to increase. For some problems, after a certain amount of deformation, the structure may start to take more loading to keep deformation increasing, and a second buckling can occur. The cycle may even repeat several times.

Because the post-buckling stage is unstable, special techniques must be used. [Nonlinear stabilization](#) can help with local and global buckling, and the [arc-length method](#) is useful for global buckling. For more information, see [Unstable Structures \(p. 258\)](#).

[Nonlinear stabilization analysis](#) is more straightforward for a post-buckling analysis. Because the buckling load is unknown at the beginning of an analysis, you can do perform a nonlinear analysis as usual using [automatic time stepping](#). When the buckling load is reached or a convergence problem occurs, you can activate stabilization during a multiframe restart and continue the analysis. If the deformation becomes stable later, you can deactivate stabilization until the next buckling occurs.

If only local buckling exists, the total load could still increase when buckling occurs because the total loading is distributed differently. For these cases, nonlinear stabilization is the only applicable technique.

Because nonlinear stabilization cannot detect the negative slope of a load-vs.-displacement curve, it may yield less accurate results for history-dependent materials, and the maximum loads (buckling loads) may not be obvious. For such cases, use the arc-length method.

## 7.5. Procedure for Eigenvalue Buckling Analysis

Again, remember that eigenvalue buckling analysis generally yields *unconservative* results, and should usually not be used for design of actual structures. If you decide that eigenvalue buckling analysis is appropriate for your application, follow this procedure:

1. Build the model.
2. Obtain the static solution.
3. Obtain the eigenvalue buckling solution.
4. Review the results.

### 7.5.1. Build the Model

See [Building the Model](#) in the *Basic Analysis Guide*. For further details, see the *Modeling and Meshing Guide*.

#### 7.5.1.1. Points to Remember

- Only linear behavior is valid. Nonlinear elements, if any, are treated as linear. If you include contact elements, for example, their stiffnesses are calculated based on their initial status and are never changed. The program assumes that the initial status of the contact elements is the status at the completion of the static prestress analysis.
- Young's modulus (EX) (or stiffness in some form) must be defined. Material properties may be linear, isotropic or orthotropic, and constant or temperature-dependent. Nonlinear properties, if any, are ignored.

### 7.5.2. Obtain the Static Solution

The procedure to obtain a static solution is the same as described in [Structural Static Analysis \(p. 9\)](#), with the following exceptions:

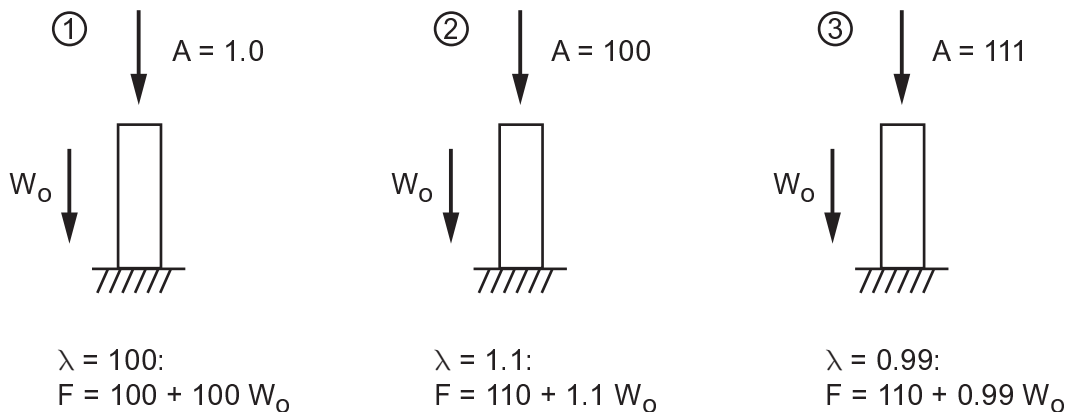
- Prestress effects (**PSTRES**) must be activated. Eigenvalue buckling analysis requires the stress stiffness matrix to be calculated.

- Unit loads are usually sufficient (that is, actual load values need not be specified). The eigenvalues calculated by the buckling analysis represent buckling load factors. Therefore, if a unit load is specified, the load factors represent the buckling loads. *All* loads are scaled. (Also, the maximum permissible eigenvalue is 1,000,000 - you must use larger applied loads if your eigenvalue exceeds this limit.)
- It is possible that different buckling loads may be predicted from seemingly equivalent pressure and force loads in an eigenvalue buckling analysis. The difference can be attributed to the fact that pressure is considered as a “follower” load. The force on the surface depends on the prescribed pressure magnitude and also on the surface orientation. Forces are not considered as follower loads. As with any numerical analysis, it is recommended to use the type of loading which best models the in-service component. See [Pressure Load Stiffness](#) of the *Mechanical APDL Theory Reference* for more details.
- Note that eigenvalues represent scaling factors for *all* loads. If certain loads are constant (for example, self-weight gravity loads) while other loads are variable (for example, externally applied loads), you need to ensure that the stress stiffness matrix from the constant loads is not factored by the eigenvalue solution.

One strategy that you can use to achieve this end is to iterate on the eigensolution, adjusting the variable loads until the eigenvalue becomes 1.0 (or nearly 1.0, within some convergence tolerance).

Consider, for example, a pole having a self-weight  $W_0$ , which supports an externally-applied load,  $A$ . To determine the limiting value of  $A$  in an eigenvalue buckling solution, you could solve repeatedly, using different values of  $A$ , until by iteration you find an eigenvalue acceptably close to 1.0.

**Figure 7.2: Adjusting Variable Loads to Find an Eigenvalue of 1.0**



- You can apply a nonzero constraint in the prestressing pass as the static load. The eigenvalues found in the buckling solution will be the load factors applied to these nonzero constraint values. However, the mode shapes will have a zero value at these degrees of freedom (and not the nonzero value specified).
- At the end of the solution, leave SOLUTION (**FINISH**).

### 7.5.3. Obtain the Eigenvalue Buckling Solution

This step requires file `Jobname.ESAV` from the static analysis. Also, the database must contain the model data (issue **RESUME** if necessary). Follow the steps below to obtain the eigenvalue buckling solution.

1. Enter the ANSYS solution processor.

**Command(s):** /SOLU

**GUI:** Main Menu> Solution

2. Specify the analysis type.

**Command(s):** ANTYPE, BUCKLE

**GUI:** Main Menu> Solution> Analysis Type> New Analysis

---

**Note**

Restarts are not valid in an eigenvalue buckling analysis.

---

---

**Note**

When you specify an eigenvalue buckling analysis, a **Solution** menu that is appropriate for buckling analyses appears. The **Solution** menu will be either "abridged" or "unabridged", depending on the actions you took prior to this step in your ANSYS session. The abridged menu contains only those solution options that are valid and/or recommended for buckling analyses. If you are on the abridged **Solution** menu and you want to access other solution options (that is, solution options that are valid for you to use, but their use may not be encouraged for this type of analysis), select the **Unabridged Menu** option from the **Solution** menu. For details, see [Using Abridged Solution Menus](#) in the *Basic Analysis Guide*.

---

3. Specify analysis options.

**Command(s):** BUCOPT, *Method*, NMODE, SHIFT, LDMULTE, RangeKey

**GUI:** Main Menu> Solution> Analysis Type> Analysis Options

Regardless of whether you use the command or GUI method, you can specify values for these options:

- For *Method*, specify the eigenvalue extraction method. The methods available for buckling are Block Lanczos and Subspace Iteration. Both methods use the full system matrices. See [Eigenvalue and Eigenvector Extraction](#) in the *Mechanical APDL Theory Reference* for more information on these two methods.
- For *NMODE*, specify the number of buckling modes (i.e., eigenvalues or load multipliers) to be extracted. This argument defaults to one, which is usually sufficient for eigenvalue buckling. We recommend that you request an additional few modes beyond what is needed in order to enhance the accuracy of the final solution.
- For *SHIFT*, specify the initial shift point about which the buckling modes are calculated (defaults to 0.0). When *RangeKey* is set to RANGE, *SHIFT* is the lower end of the load multiplier range of interest. Modifying the shift point can be helpful when numerical problems are encountered.
- For *LDMULTE*, specify the boundary of the load multiplier range of interest (defaults to  $\infty$ ). When *RangeKey* is set to CENTER, *LDMULTE* is used to determine the lower and upper ends of the load multiplier range of interest. When *RangeKey* is set to RANGE, the *LDMULTE* value is the upper end of the load multiplier range of interest.



- For *RangeKey*, specify either CENTER or RANGE. When *RangeKey* = CENTER, the program computes *NMODE* buckling modes centered around *SHIFT* in the range of  $(-LDMULTE, +LDMULTE)$ . When *RangeKey* = RANGE, the program computes *NMODE* buckling modes in the range of  $(SHIFT, LDMULTE)$ .

Specify expansion pass options.

**Command(s):** **MPAND**,*NMODE*,*Elcalc*

**GUI:** Main Menu> Solution> Load Step Opts> ExpansionPass> Expand Modes

Regardless of whether you use the command or GUI method, the following options are required for the expansion pass:

- For *NMODE*, specify the number of modes to expand. This argument defaults to the total number of modes that were extracted.
  - For *Elcalc*, indicate whether you want ANSYS to calculate stresses. "Stresses" in an eigenvalue analysis do not represent actual stresses, but give you an idea of the relative stress or force distribution for each mode. By default, no stresses are calculated.
4. Specify load step options.

The only load step options valid for eigenvalue buckling are output controls.

Database and Results File Output:

**Command(s):** **OUTRES**

**GUI:** Main Menu> Solution> Load Step Opts> Output Ctrl> DB/Results File

Output File:

**Command(s):** **OUTPR**,*NSOL*,*ALL*

**GUI:** Main Menu> Solution> Load Step Opts> OutputCtrls> Solu Printout

---

### Note

The *FREQ* field on **OUTPR OUTRES** can be only ALL or NONE, meaning that the data are written for *all* modes or *no* modes. For example, you cannot write information for every other mode.

---

5. Save a backup copy of the database to a named file.

**Command(s):** **SAVE**

**GUI:** Utility Menu> File> Save As

6. Start solution calculations.

**Command(s):** **SOLVE**

**GUI:** Main Menu> Solution> Solve> Current LS

The output from the solution mainly consists of the eigenvalues, which are printed as part of the printed output (*Jobname. OUT*). The eigenvalues represent the buckling load factors; if unit loads were applied in the static analysis, they *are* the buckling loads. No buckling mode shapes are written to the database or the results file, so you cannot postprocess the results yet. To do this, you need to expand the solution (explained next).

Sometimes you may see both positive and negative eigenvalues calculated. Negative eigenvalues indicate that buckling occurs when the loads are applied in an opposite sense.

- Exit the SOLUTION processor.

**Command(s): FINISH**

**GUI:** Close the Solution menu.

### 7.5.4. Review the Results

Results from a buckling expansion pass are written to the structural results file, `Jobname.RST`. They consist of buckling load factors, buckling mode shapes, and relative stress distributions. You can review them in POST1, the general postprocessor.

---

#### Note

To review results in POST1, the database must contain the same model for which the buckling solution was calculated (issue **RESUME** if necessary). Also, the results file (`Jobname.RST`) from the expansion pass must be available.

---

- List all buckling load factors.

**Command(s): SET,LIST**

**GUI:** Main Menu> General Postproc> Results Summary

- Read in data for the desired mode to display buckling mode shapes. (Each mode is stored on the results file as a separate substep.)

**Command(s): SET,SBSTEP**

**GUI:** Main Menu> General Postproc> Read Results> *load step*

- Display the mode shape.

**Command(s): PLDISP**

**GUI:** Main Menu> General Postproc> Plot Results> Deformed Shape

- Contour the relative stress distributions.

**Command(s): PLNSOL** or **PLESOL**

**GUI:** Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solution

**Main Menu> General Postproc> Plot Results> Contour Plot> Element Solution**

See the *Command Reference* for a discussion of the **ANTYPE**, **PSTRES**, **D**, **F**, **SF**, **BUCOPT**, **EXPASS**, **MX-PAND**, **OUTRES**, **SET**, **PLDISP**, and **PLNSOL** commands.

## 7.6. Sample Buckling Analysis (GUI Method)

In this sample problem, you will analyze the buckling of a bar with hinged ends.

### 7.6.1. Problem Description

Determine the critical buckling load of an axially loaded long slender bar of length  $\ell$  with hinged ends. The bar has a cross-sectional height  $h$ , and area  $A$ . Only the upper half of the bar is modeled because of symmetry. The boundary conditions become free-fixed for the half-symmetry model. The moment of inertia of the bar is calculated as  $I = Ah^2/12 = 0.0052083 \text{ in}^4$ .

### 7.6.2. Problem Specifications

The model is assumed to act only in the X-Y plane.

The following material properties are used:

$$E = 30 \times 10^6 \text{ psi}$$

The following geometric properties are used:

$$l = 200 \text{ in}$$

$$A = 0.25 \text{ in}^2$$

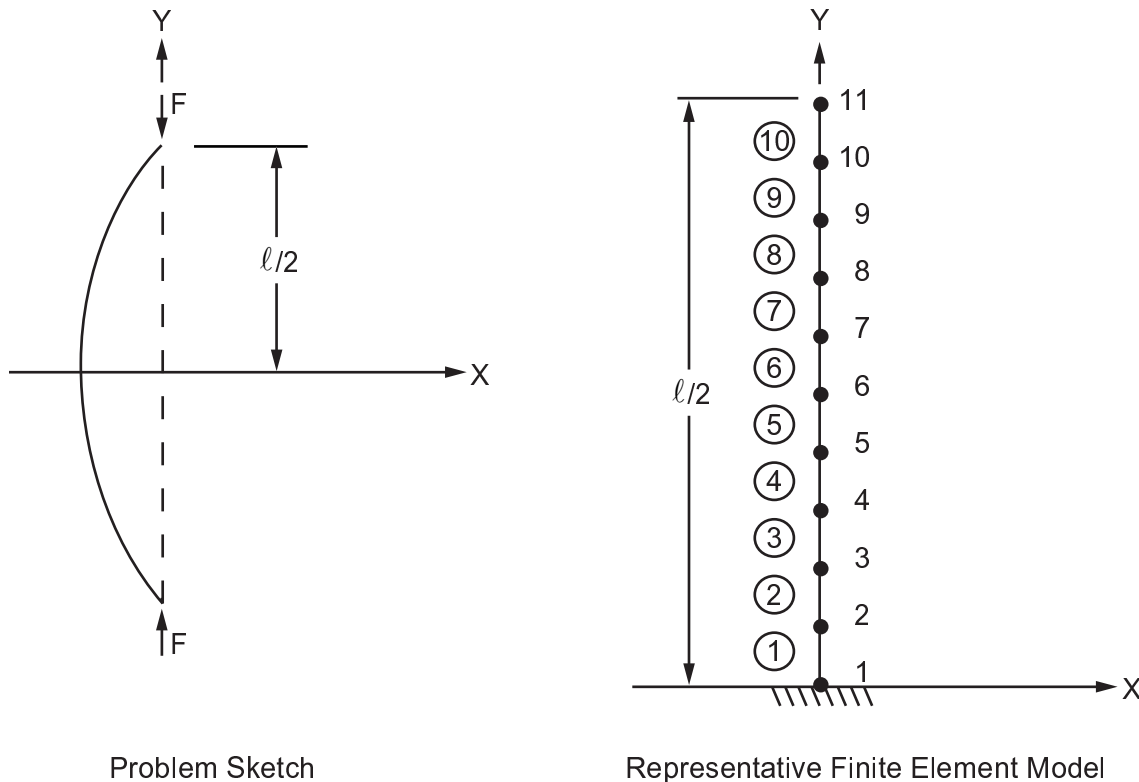
$$h = 0.5 \text{ in}$$

Loading is as follows:

$$F = 1 \text{ lb.}$$

### 7.6.3. Problem Sketch

**Figure 7.3: Bar with Hinged Ends**



#### 7.6.3.1. Set the Analysis Title

After you enter the ANSYS program, follow these steps to set the title.

1. Choose menu path **Utility Menu > File > Change Title**.
2. Enter the text "Buckling of a Bar with Hinged Ends" and click on OK.

#### 7.6.3.2. Define the Element Type

In this step, you define **BEAM188** as the element type.

1. Choose menu path **Main Menu> Preprocessor> Element Type> Add/Edit/Delete**. The Element Types dialog box appears.
2. Click Add. The Library of Element Types dialog box appears.
3. In the scroll box on the left, click on "Structural Beam" to select it.
4. In the scroll box on the right, click on "2 Node 188" to select it.
5. Click OK. The Library of Element Types dialog box closes.
6. Click Options in the Element Types dialog box.
7. Choose Element Behavior K3 : Cubic Form. Click OK.
8. Click Close in the Element Types dialog box.

### **7.6.3.3. Define the Real Constants and Material Properties**

1. Choose menu path **Main Menu> Preprocessor> Sections> Beam> Common Sections**. The BeamTool dialog box appears.
2. Enter .5 for B and .5 for H.
3. Click OK.
4. Choose menu path **Main Menu> Preprocessor> Material Props> Material Models**. The Define Material Model Behavior dialog box appears.
5. In the Material Models Available window, double-click on the following options: Structural, Linear, Elastic, Isotropic. A dialog box appears.
6. Enter 30e6 for EX (Young's modulus), and click on OK. Material Model Number 1 appears in the Material Models Defined window on the left.
7. Choose menu path **Material> Exit** to remove the Define Material Model Behavior dialog box.

### **7.6.3.4. Define Nodes and Elements**

1. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Nodes> In Active CS**. The Create Nodes in Active Coordinate System dialog box appears.
2. Enter 1 for node number.
3. Click Apply. Node location defaults to 0,0,0.
4. Enter 11 for node number.
5. Enter 0,100,0 for the X, Y, Z coordinates.

- Click OK. The two nodes appear in the ANSYS Graphics window.

---

### Note

The triad, by default, hides the node number for node 1. To turn the triad off, choose menu path **Utility Menu > PlotCtrls > Window Controls > Window Options** and select the "Not Shown" option for Location of triad. Then click OK to close the dialog box.

---

- Choose menu path **Main Menu > Preprocessor > Modeling > Create > Nodes > Fill between Nds**. The Fill between Nds picking menu appears.
- Click on node 1, then 11, and click on OK. The Create Nodes Between 2 Nodes dialog box appears.
- Click OK to accept the settings (fill between nodes 1 and 11, and number of nodes to fill 9).
- Choose menu path **Main Menu > Preprocessor > Modeling > Create > Elements > Auto Numbered > Thru Nodes**. The Elements from Nodes picking menu appears.
- Click on nodes 1 and 2, then click on OK.
- Choose menu path **Main Menu > Preprocessor > Modeling > Copy > Elements > Auto Numbered**. The Copy Elms Auto-Num picking menu appears.
- Click Pick All. The Copy Elements (Automatically-Numbered) dialog box appears.
- Enter 10 for total number of copies and enter 1 for node number increment.
- Click OK. The remaining elements appear in the ANSYS Graphics window.

### 7.6.3.5. Define the Boundary Conditions

- Choose menu path **Main Menu > Solution > Unabridged Menu > Analysis Type > New Analysis**. The New Analysis dialog box appears.
- Click OK to accept the default of "Static."
- Choose menu path **Main Menu > Solution > Analysis Type > Analysis Options**. The Static or Steady-State Analysis dialog box appears.
- In the scroll box for stress stiffness or prestress, scroll to "Prestress ON" to select it.
- Click OK.
- Choose menu path **Main Menu > Solution > Define Loads > Apply > Structural > Displacement > On Nodes**. The Apply U,ROT on Nodes picking menu appears.
- Click on node 1 in the ANSYS Graphics window, then click on OK in the picking menu. The Apply U,ROT on Nodes dialog box appears.
- Click on "UY" and "ROTZ" to select them, and click on OK.
- Choose menu path **Main Menu > Solution > Define Loads > Apply > Structural > Displacements > On Nodes**. The Apply U,ROT on Nodes picking menu appears.

10. Click on node 11 in the ANSYS Graphics window, then click on OK in the picking menu. The Apply U,ROT on Nodes dialog box appears.
11. Click on "UX" to select it, and click on OK.
12. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Force/Moment> On Nodes**. The Apply F/M on Nodes picking menu appears.
13. Click on node 11, then click OK. The Apply F/M on Nodes dialog box appears.
14. In the scroll box for Direction of force/mom, scroll to "FY" to select it.
15. Enter -1 for the force/moment value, and click on OK. The force symbol appears in the ANSYS Graphics window.
16. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> Symmetry B.C.> On Nodes**. The Apply SYMM on Nodes box dialog appears.
17. In the scroll box for Norml symm surface is normal to, scroll to "z-axis" and click on OK.

### **7.6.3.6. Solve the Static Analysis**

1. Choose menu path **Main Menu> Solution> Solve> Current LS**.
2. Carefully review the information in the status window, and click on Close.
3. Click OK in the Solve Current Load Step dialog box to begin the solution.
4. Click Close in the Information window when the solution is finished.

### **7.6.3.7. Solve the Buckling Analysis**

1. Choose menu path **Main Menu> Solution> Analysis Type> New Analysis**.

---

#### **Note**

Click Close in the Warning window if the following warning appears: Changing the analysis type is only valid within the first load step. Pressing OK will cause you to exit and reenter SOLUTION. This will reset the load step count to 1.

---

2. In the New Analysis dialog box, click the "Eigen Buckling" option on, then click on OK.
3. Choose menu path **Main Menu> Solution> Analysis Type> Analysis Options**. The Eigenvalue Buckling Options dialog box appears.
4. Enter 1 for number of modes to extract.
5. Click OK.
6. Choose menu path **Main Menu> Solution> Load Step Opts> ExpansionPass> Single Expand> Expand Modes**.
7. Enter 1 for number of modes to expand, and click on OK.

8. Choose menu path **Main Menu> Solution> Solve> Current LS**.
9. Carefully review the information in the status window, and click on Close.
10. Click OK in the Solve Current Load Step dialog box to begin the solution.
11. Click Close in the Information window when the solution is finished.

### 7.6.3.8. Review the Results

1. Choose menu path **Main Menu> General Postproc> Read Results> First Set**.
2. Choose menu path **Main Menu> General Postproc> Plot Results> Deformed Shape**. The Plot Deformed Shape dialog box appears.
3. Click the "Def + undeformed" option on. Click OK. The deformed and undeformed shapes appear in the ANSYS graphics window.

### 7.6.3.9. Exit ANSYS

1. In the ANSYS Toolbar, click on Quit.
2. Choose the save option you want and click on OK.

## 7.7. Sample Buckling Analysis (Command or Batch Method)

You can perform the example buckling analysis of a bar with hinged ends using the ANSYS commands shown below instead of GUI choices. Items prefaced by an exclamation point (!) are comments.

```

/prep7

et,1,188
keyopt,1,3,3

sectype,1,beam,rect
secdata,0.5,0.5

mp,ex,1,30e6
mp,prxy,1,0.3

n,1,0,0,0
n,11,0,100,0
fill,1,11,9,,1,1,1,
e,1,2
egen,10,1,1
finish

/solu
nset,s,node,,1
d,all,uy,,,,,rotz
allsel
nset,s,node,,11
d,all,ux
f,all,fy,-1
allsel
dsym,symm,z
rescontrol,linear,all,1
solve
finish

/solu
antype,static,restart,,,perturb

```

```
perturb,buckle,,allkeep
solve,elform

bucopt,lanb,3
mexpand,3
solve
finish

/post1
file,,rstp
set,list
!pdisp,1           ! uncomment to plot displacement
finish
```

## 7.8. Where to Find Other Examples

Several ANSYS publications, particularly the *Mechanical APDL Verification Manual*, describe additional buckling analyses.

The *Mechanical APDL Verification Manual* consists of test case analyses demonstrating the analysis capabilities of the ANSYS program. While these test cases demonstrate solutions to realistic analysis problems, the *Mechanical APDL Verification Manual* does not present them as step-by-step examples with lengthy data input instructions and printouts. However, most ANSYS users who have at least limited finite element experience should be able to fill in the missing details by reviewing each test case's finite element model and input data with accompanying comments.

The *Mechanical APDL Verification Manual* contains a variety of buckling analysis test cases:

- VM17 - Snap-Through Buckling of a Hinged Shell
- VM127 - Buckling of a Bar with Hinged Ends (Line Elements)
- VM128 - Buckling of a Bar with Hinged Ends (Area Elements)



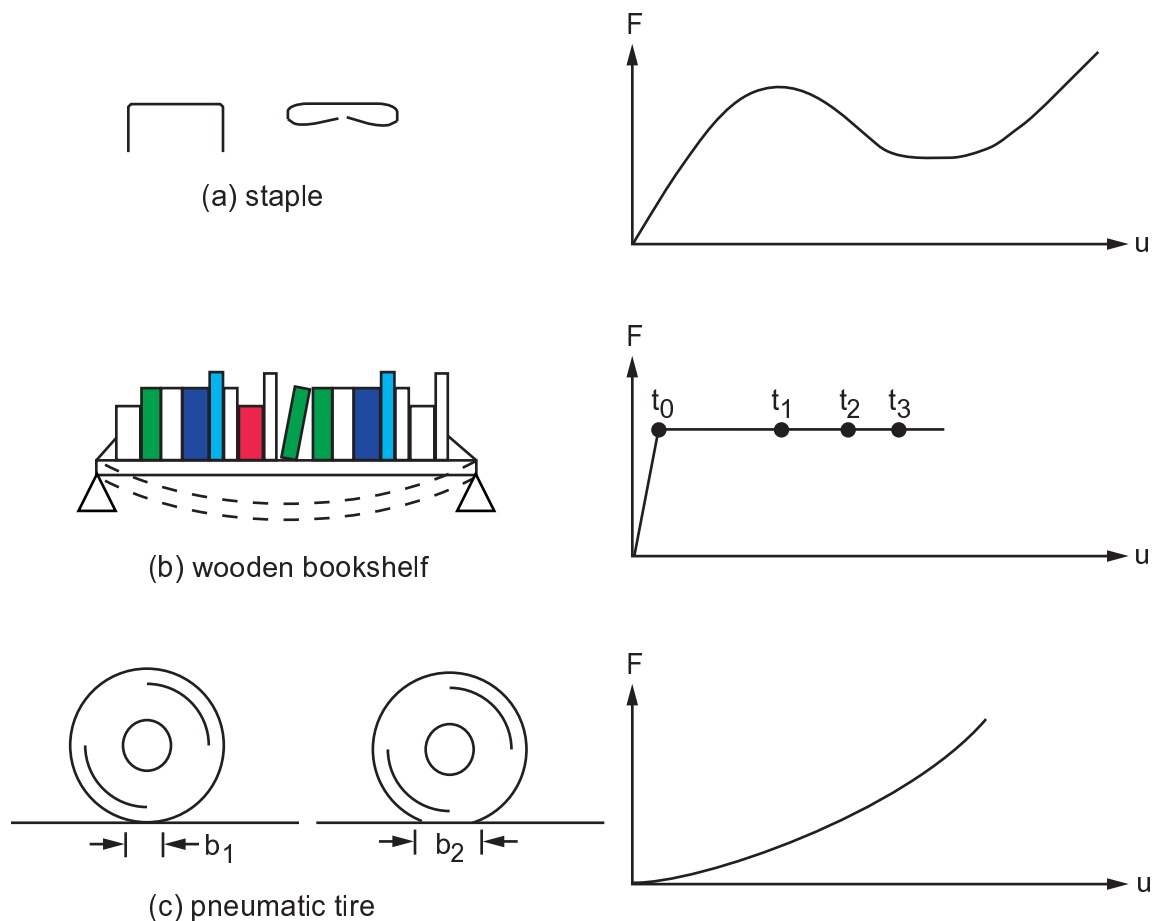
---

## Chapter 8: Nonlinear Structural Analysis

---

Structural nonlinearities occur on a routine basis. For example, whenever you staple two pieces of paper together, the metal staples are permanently bent into a different shape, as shown in Figure 8.1: Common Examples of Nonlinear Structural Behavior (p. 193) (a). If you heavily load a wooden shelf, it sags more and more as time passes, as shown in Figure (b). As weight is added to a car or truck, the contact surfaces between its pneumatic tires and the underlying pavement change in response to the added load, as shown in Figure (c). If you were to plot the load-deflection curve for each example, you would discover that they exhibit the fundamental characteristic of nonlinear structural behavior: a changing structural stiffness.

**Figure 8.1: Common Examples of Nonlinear Structural Behavior**



The following nonlinear structural analysis topics are available:

- 8.1. Causes of Nonlinear Behavior
- 8.2. Understanding Nonlinear Analyses
- 8.3. Using Geometric Nonlinearities
- 8.4. Modeling Material Nonlinearities
- 8.5. Running a Nonlinear Analysis
- 8.6. Performing a Nonlinear Static Analysis

- 8.7. Performing a Nonlinear Transient Analysis
- 8.8. Example Input for a Nonlinear Transient Analysis
- 8.9. Restarts
- 8.10. Using Nonlinear (Changing-Status) Elements
- 8.11. Unstable Structures
- 8.12. Guidelines for Nonlinear Analysis
- 8.13. Example Nonlinear Analysis (GUI Method)
- 8.14. Example Nonlinear Analysis (Command or Batch Method)
- 8.15. Where to Find Other Examples

## 8.1. Causes of Nonlinear Behavior

Nonlinear structural behavior arises from a number of causes, which can be grouped into these principal categories:

- Changing status
- Geometric nonlinearities
- Material nonlinearities

### 8.1.1. Changing Status (Including Contact)

Many common structural features exhibit nonlinear behavior that is *status-dependent*. For example, a tension-only cable is either slack or taut; a roller support is either in contact or not in contact. Status changes might be directly related to load (as in the case of the cable), or they might be determined by some external cause.

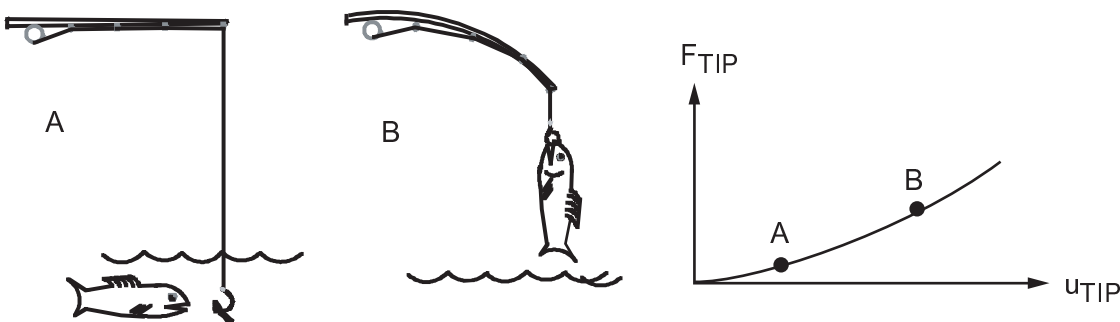
Situations in which *contact* occurs are common to many different nonlinear applications. Contact forms a distinctive and important subset to the category of changing-status nonlinearities.

See the [Contact Technology Guide](#) for detailed information about performing contact analyses.

### 8.1.2. Geometric Nonlinearities

If a structure experiences large deformations, its changing geometric configuration can cause the structure to respond nonlinearly. An example would be the fishing rod shown in [Figure 8.2: A Fishing Rod Demonstrates Geometric Nonlinearity](#) (p. 194). Geometric nonlinearity is characterized by "large" displacements and/or rotations.

**Figure 8.2: A Fishing Rod Demonstrates Geometric Nonlinearity**



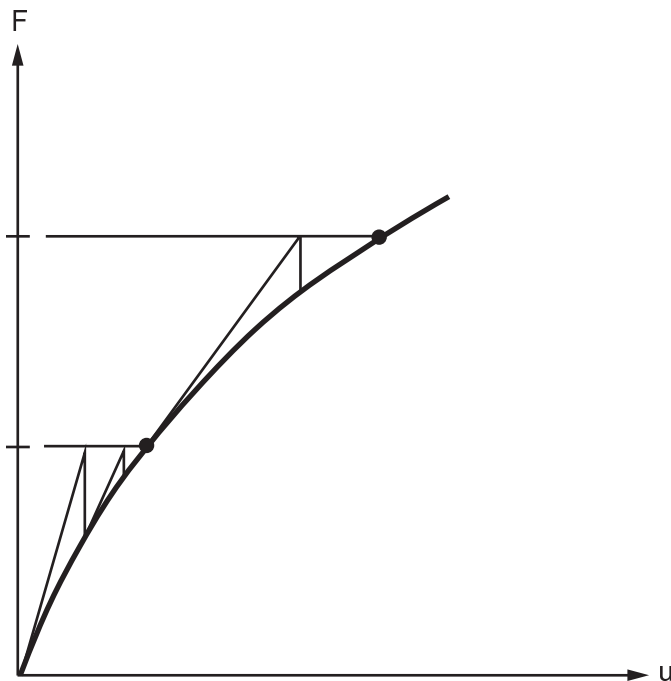
### 8.1.3. Material Nonlinearities

Nonlinear stress-strain relationships are a common cause of nonlinear structural behavior. Many factors can influence a material's stress-strain properties, including load history (as in elastoplastic response), environmental conditions (such as temperature), and the amount of time that a load is applied (as in creep response).

## 8.2. Understanding Nonlinear Analyses

The program uses the Newton-Raphson approach to solve nonlinear problems. The load is subdivided into a series of load increments which can be applied over several load steps. [Figure 8.3: Newton-Raphson Approach \(p. 195\)](#)he following figure illustrates the use of Newton-Raphson equilibrium iterations in a single-degree-of-freedom nonlinear analysis.

**Figure 8.3: Newton-Raphson Approach**



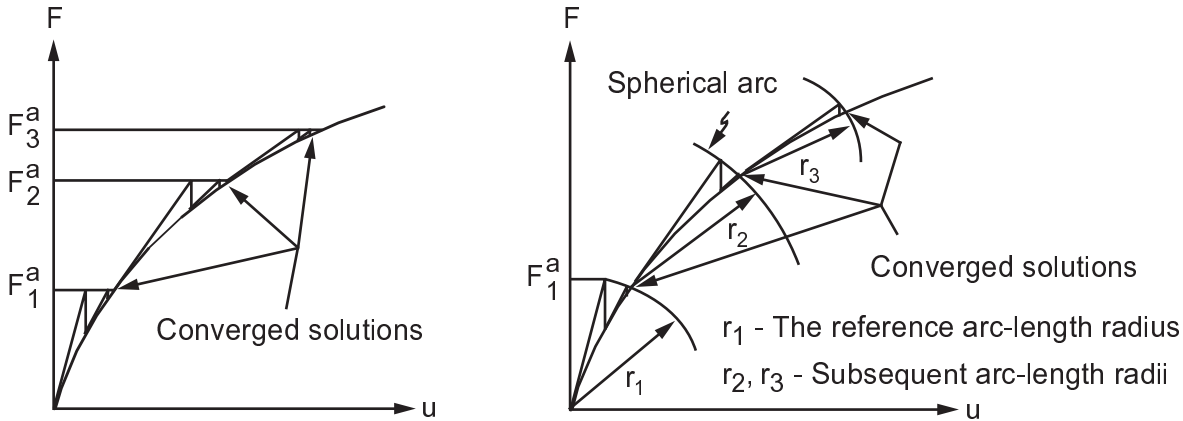
Before each solution, the Newton-Raphson method evaluates the out-of-balance load vector, which is the difference between the restoring forces (the loads corresponding to the element stresses) and the applied loads. The program then performs a linear solution, using the out-of-balance loads, and checks for convergence. If convergence criteria are not satisfied, the out-of-balance load vector is reevaluated, the stiffness matrix is updated, and a new solution is obtained. This iterative procedure continues until the problem converges.

A number of convergence-enhancement and recovery features, such as line search, automatic load stepping, and bisection, can be activated to help the problem to converge. If convergence cannot be achieved, then the program attempts to solve with a smaller load increment.

In some nonlinear static analyses, if you use the Newton-Raphson method alone, the tangent stiffness matrix may become singular (or non-unique), causing severe convergence difficulties. Such occurrences include nonlinear buckling analyses in which the structure either collapses completely or "snaps through" to another stable configuration. For such situations, you can activate an alternative iteration scheme, the *arc-length method*, to help avoid bifurcation points and track unloading.

The arc-length method causes the Newton-Raphson equilibrium iterations to converge along an *arc*, thereby often preventing divergence, even when the slope of the load vs. deflection curve becomes zero or negative. This iteration method is represented schematically in [Figure 8.4: Traditional Newton-Raphson Method vs. Arc-Length Method \(p. 196\)](#).

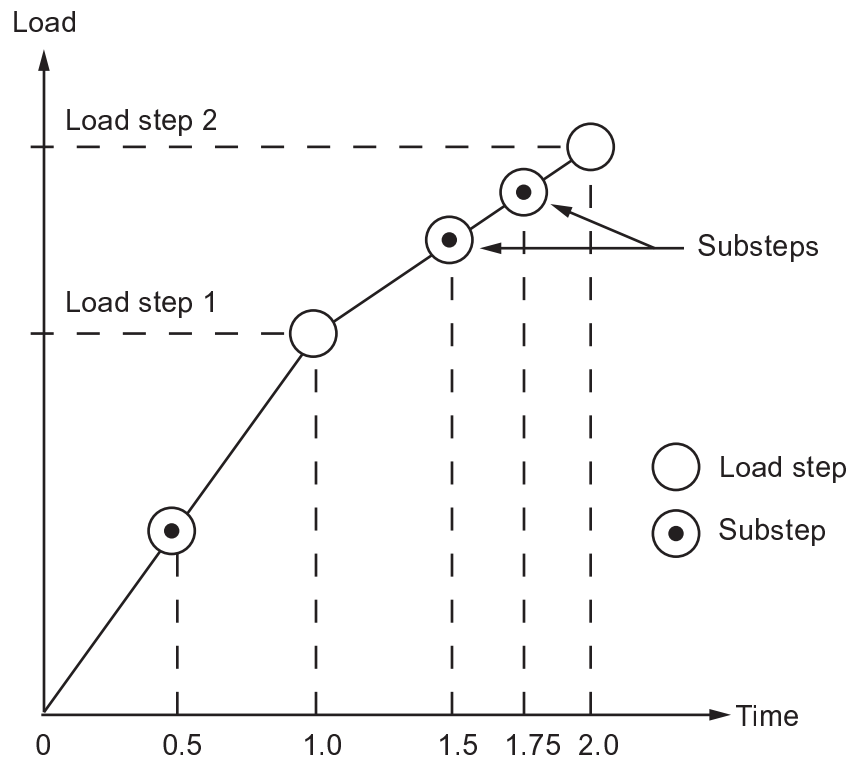
**Figure 8.4: Traditional Newton-Raphson Method vs. Arc-Length Method**



To summarize, a nonlinear analysis is organized into three levels of operation:

- The "top" level consists of the *load steps* that you define explicitly over a "time" span (see the discussion of "time" in [Loading](#) in the *Basic Analysis Guide*). Loads are assumed to vary linearly within load steps (for static analyses).
- Within each load step, you can direct the program to perform several solutions (*substeps* or *time steps*) to apply the load gradually.
- At each substep, the program performs a number of *equilibrium iterations* to obtain a converged solution.

[Figure 8.5: Load Steps, Substeps, and Time \(p. 197\)](#) illustrates a typical load history for a nonlinear analysis. Also see the discussion of load steps, substeps, and equilibrium iterations in [Loading](#) in the *Basic Analysis Guide*.

**Figure 8.5: Load Steps, Substeps, and Time**

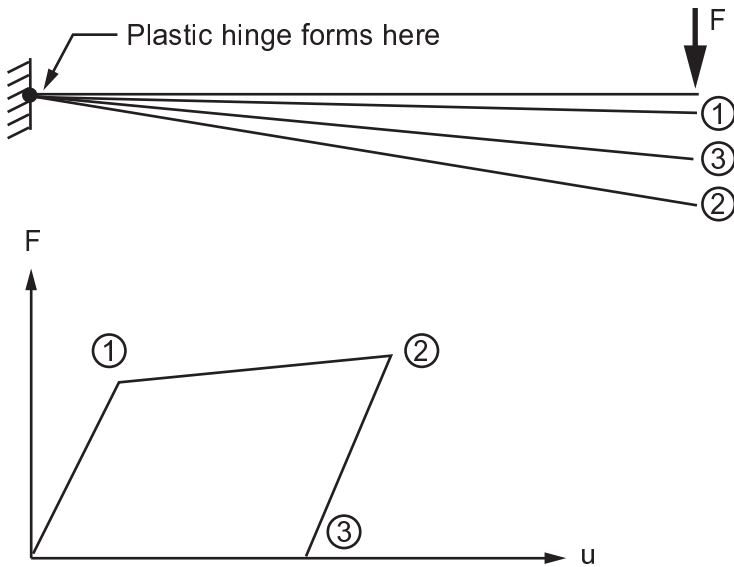
The program gives you a number of choices when you designate convergence criteria: you can base convergence checking on forces, moments, displacements, or rotations, or on any combination of these items. Additionally, each item can have a different convergence tolerance value. For multiple-degree-of-freedom problems, you also have a choice of convergence norms.

You should almost always employ a force-based (and, when applicable, moment-based) convergence tolerance. Displacement-based (and, when applicable, rotation-based) convergence checking can be *added*, if desired, but should usually not be used alone.

### 8.2.1. Conservative vs. Nonconservative Behavior; Path Dependency

If all energy put into a system by external loads is recovered when the loads are removed, the system is said to be *conservative*. If energy is dissipated by the system (such as by plastic deformation or sliding friction), the system is said to be *nonconservative*. An example of a nonconservative system is shown in [Figure 8.6: Nonconservative \(Path-Dependent\) Behavior \(p. 198\)](#).

An analysis of a conservative system is path independent: loads can usually be applied in any order and in any number of increments without affecting the end results. Conversely, an analysis of a nonconservative system is path dependent: the actual load-response history of the system must be followed closely to obtain accurate results. An analysis can also be path dependent if more than one solution could be valid for a given load level (as in a snap-through analysis). Path dependent problems usually require that loads be applied slowly (that is, using many substeps) to the final load value.

**Figure 8.6: Nonconservative (Path-Dependent) Behavior**

### 8.2.2. Substeps

When using multiple substeps, you need to achieve a balance between accuracy and economy: more substeps (that is, small time step sizes) usually result in better accuracy, but at a cost of increased run times. The program provides automatic time stepping designed for this purpose.

*Automatic time stepping* adjusts the time step size as needed, gaining a better balance between accuracy and economy. Automatic time stepping activates the *bisection* feature.

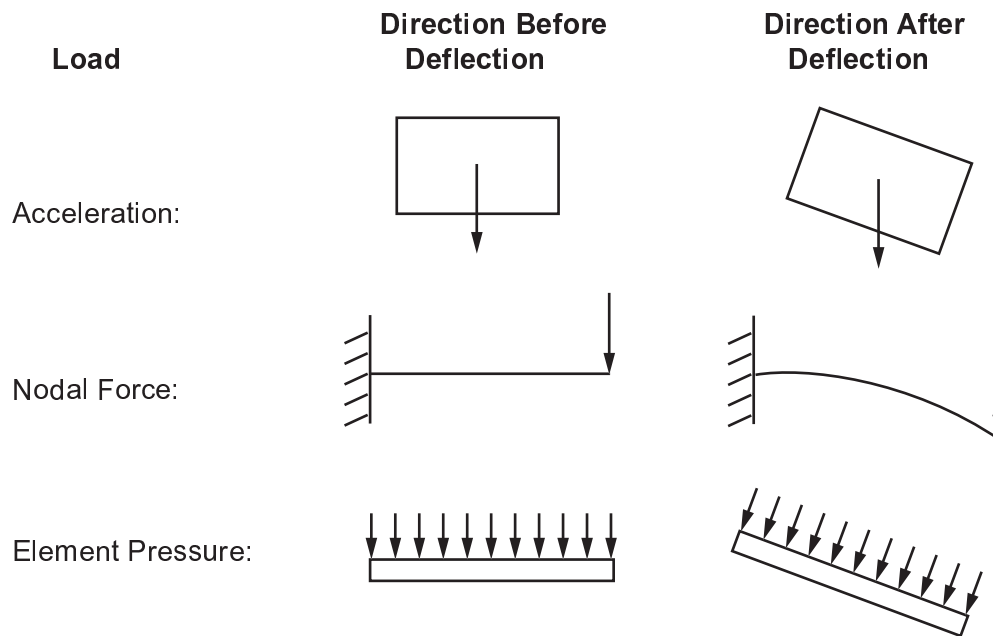
Bisection provides a means of automatically recovering from a convergence failure. This feature cuts a time step size in half whenever equilibrium iterations fail to converge and automatically restart from the last converged substep. If the halved time step again fails to converge, bisection again cuts the time step size and restart, continuing the process until convergence is achieved or until the minimum time step size (specified by you) is reached.

### 8.2.3. Load Direction in a Large-Deflection Analysis

Consider what happens to loads when your structure experiences large deflections. In many instances, the loads applied to your system maintain constant direction no matter how the structure deflects. In other cases, forces change direction, "following" the elements as they undergo large rotations.

The program can model both situations, depending on the type of load applied. *Accelerations* and *concentrated forces* maintain their original orientation, regardless of the element orientation. *Pressure loads* always act normal to the deflected element surface, and can be used to model "following" forces. [Figure 8.7: Load Directions Before and After Deflection \(p. 199\)](#) illustrates constant-direction and following forces.

Nodal coordinate system orientations are not updated in a large deflection analysis. Calculated displacements are therefore output in the original directions.

**Figure 8.7: Load Directions Before and After Deflection**

### 8.2.4. Rotations in a Large-Deflection Analysis

Applying an imposed rotation (**D** command) in a load step is done in a "rotation vector" form, where the magnitude and rotation direction are given by the values of the ROTX, ROTY, and ROTZ components on the **D** command(s). For step loading (**KBC,1**), the imposed rotation values are applied at the beginning of the load step. For ramped loading (**KBC,0**), the values are linearly ramped over the load step.

For compound rotations imposed over multiple load steps, each set of rotations is applied sequentially to the previous deformed configuration. For example, rotating a body about a nodal x-axis first then rotating it about a nodal y-axis is done simply as:

```

D,node,ROTX,value
D,node,ROTY,0.0
D,node,ROTZ,0.0
SOLVE
D,node,ROTY,value
D,node,ROTX,value
D,node,ROTZ,0.0
SOLVE

```

This simplifies the specification of compound motion such as a robotic arm. See examples presented in [Appendix A \(p. 483\)](#) to find how different ways of specifying imposed rotations affect the final configuration.

The sequence in which the rotations are applied determine how the constrained node rotates from its initial configuration to its final configuration. The examples presented in [Appendix A \(p. 483\)](#) demonstrate the effect of the order of imposed rotations on the final configuration.

The rotation components of the output displacements (ROTX, ROTY, and ROTZ) are the sum of all incremental rotations. They generally cannot be interpreted as a pseudovector or rotation vector.

## 8.2.5. Nonlinear Transient Analyses

The procedure for analyzing nonlinear transient behavior is similar to that used for nonlinear static behavior: you apply the load in incremental steps, and the program performs equilibrium iterations at each step. The main difference between the static and transient procedures is that time-integration effects can be activated in the transient analysis. Thus, "time" always represents actual chronology in a transient analysis. The automatic time stepping and bisection feature is also applicable for transient analyses.

## 8.3. Using Geometric Nonlinearities

Small deflection and small strain analyses assume that displacements are small enough that the resulting stiffness changes are insignificant.

In contrast, *large strain* analyses account for the stiffness changes that result from changes in an element's shape and orientation. By issuing **NLGEOM,ON** (GUI path **Main Menu**> **Solution**> **Analysis Type**> **Sol'n Control ( : Basic Tab)** or **Main Menu**> **Solution**> **Unabridged Menu**> **Analysis Type**> **Analysis Options**), you activate large strain effects in those element types that support this feature. The large strain feature is available in most of the solid elements (including all of the large strain elements), as well as in most of the shell and beam elements. Large strain effects are not available in the ANSYS Professional program; however, large deflection effects (**NLGEOM** command) are supported for shell and beam elements in ANSYS Professional, if indicated as such in the *Element Reference*.

The large strain procedure places no theoretical limit on the total rotation or strain experienced by an element. Certain element types are subject to practical limitations on total strain, as described below. The procedure requires that strain *increments* be restricted to maintain accuracy, however, so the total load should be broken into smaller steps.

### 8.3.1. Stress-Strain

In large strain solutions, all stress-strain input and results are in terms of *true* stress and *true* (or *logarithmic*) strain. (In one dimension, true strain would be expressed as  $\epsilon = \ln(\ell / \ell_0)$ . For small-strain regions of response, true strain and engineering strain are essentially identical.) To convert strain from small (engineering) strain to logarithmic strain, use  $\epsilon_{\ln} = \ln(1 + \epsilon_{\text{eng}})$ . To convert from engineering stress to true stress, use  $\sigma_{\text{true}} = \sigma_{\text{eng}}(1 + \epsilon_{\text{eng}})$ . (This stress conversion is valid only for incompressible plasticity stress-strain data.)

#### 8.3.1.1. Large Deflections with Small Strain

This feature is available in all beam and most shell elements, as well as in a number of the nonlinear elements. Issue **NLGEOM,ON** (**Main Menu**> **Solution**> **Analysis Type**> **Sol'n Control ( : Basic Tab)** or **Main Menu**> **Solution**> **Unabridged Menu**> **Analysis Type**> **Analysis Options**) to activate large deflection effects for those elements that are designed for small strain analysis types that support this feature.

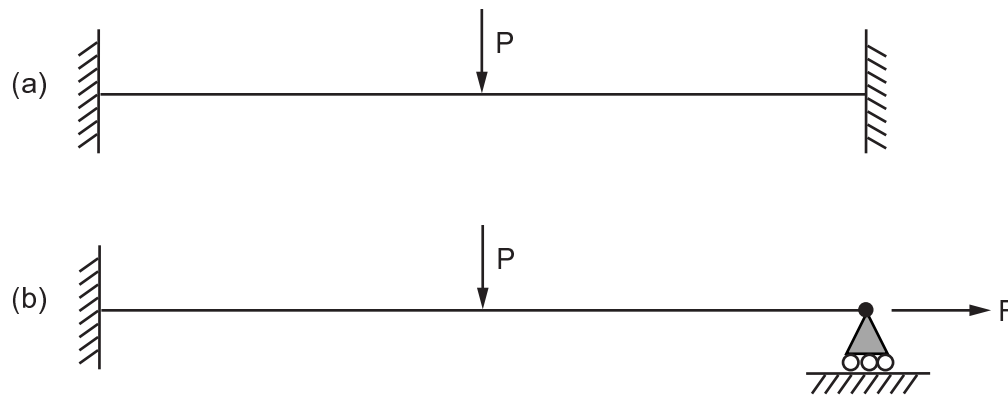
### 8.3.2. Stress Stiffening

The out-of-plane stiffness of a structure can be significantly affected by the state of in-plane stress in that structure. This coupling between in-plane stress and transverse stiffness, known as *stress stiffening*, is most pronounced in thin, highly stressed structures, such as cables or membranes. A drumhead, which gains lateral stiffness as it is tightened, would be a common example of a stress-stiffened structure.



Even though stress stiffening theory assumes that an element's rotations and strains are small, in some structural systems (such as in [Figure 8.8: Stress-Stiffened Beams \(p. 201\) \(a\)](#)), the stiffening stress is only obtainable by performing a large deflection analysis. In other systems (such as in [Figure 8.8: Stress-Stiffened Beams \(p. 201\) \(b\)](#)), the stiffening stress is obtainable using small deflection, or linear, theory.

**Figure 8.8: Stress-Stiffened Beams**



To use stress stiffening in the second category of systems, you must issue **PSTRES,ON** (GUI path **Main Menu**> **Solution**> **Unabridged Menu**> **Analysis Type**> **Analysis Options**) in your first load step.

Large strain and large deflection procedures include initial stress effects as a subset of their theory. For most elements, initial stiffness effects are automatically included when large-deformation effects are activated (**NLGEOM,ON**) (GUI path **Main Menu**> **Solution**> **Analysis Type**> **Sol'n Control ( : Basic Tab**) or **Main Menu**> **Solution**> **Unabridged Menu**> **Analysis Type**> **Analysis Options**).

## 8.4. Modeling Material Nonlinearities

A number of material-related factors can cause your structure's stiffness to change during the course of an analysis. Nonlinear stress-strain relationships of *plastic*, *multilinear elastic*, and *hyperelastic* materials cause a structure's stiffness to change at different load levels (and, typically, at different temperatures). *Creep*, *viscoplasticity*, and *viscoelasticity* give rise to nonlinearities that can be time-, rate-, temperature-, and stress-related. *Swelling* induces strains that can be a function of temperature, time, neutron flux level (or some analogous quantity), and stress. Any of these types of material properties can be incorporated into your analysis if you use appropriate element types. Nonlinear constitutive models (**TB**) are not applicable for the ANSYS Professional program.

The following topics related to modeling material nonlinearities are available:

[8.4.1. Nonlinear Materials](#)

[8.4.2. Material Model Combination Examples](#)

### 8.4.1. Nonlinear Materials

If a material displays nonlinear stress-strain behavior, use the **TB** family of commands to define the nonlinear material property relationships in terms of a data table: **TB**, **TBTEMP**, **TBDATA**, **TBPT**, **TBCOPY**, **TBLIST**, **TBPLLOT**, **TBDELE** (**Main Menu**> **Preprocessor**> **Material Props**> **Material Models**> **Structural**> **Nonlinear**). The precise form of these commands varies depending on the type of nonlinear material behavior being defined.

The various material behavior options are described briefly below. See [Material Models](#) in the [Material Reference](#) for details about each material behavior type.

Topics covering the following general categories of nonlinear material models are available:

- 8.4.1.1. Plasticity
- 8.4.1.2. Hyperelasticity Material Model
- 8.4.1.3. Bergstrom-Boyce Hyperviscoelastic Material Model
- 8.4.1.4. Mullins Effect Material Model
- 8.4.1.5. Anisotropic Hyperelasticity Material Model
- 8.4.1.6. Creep Material Model
- 8.4.1.7. Shape Memory Alloy Material Model
- 8.4.1.8. Viscoplasticity
- 8.4.1.9. Viscoelasticity
- 8.4.1.10. Swelling
- 8.4.1.11. User-Defined Material Model

### 8.4.1.1. Plasticity

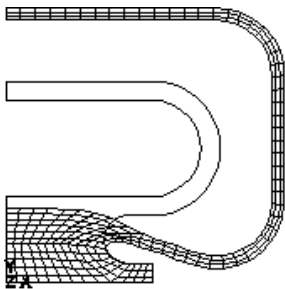
To learn about the material model options for describing plasticity behavior, see [Rate-Independent Plasticity](#) in the *Material Reference*.

### 8.4.1.2. Hyperelasticity Material Model

A material is said to be hyperelastic (**TB,HYPER**) if there exists an elastic potential function (or strain energy density function), which is a scalar function of one of the strain or deformation tensors, whose derivative with respect to a strain component determines the corresponding stress component.

Hyperelasticity can be used to analyze rubber-like materials (elastomers) that undergo large strains and displacements with small volume changes (nearly incompressible materials). Large strain theory is required (**NLGEOM,ON**). A representative hyperelastic structure (a balloon seal) is shown in [Figure 8.9: Hyperelastic Structure](#) (p. 202).

**Figure 8.9: Hyperelastic Structure**



All [current-technology elements](#) except for link and beam elements are suitable for simulating hyperelastic materials.

The material response in hyperelastic models can be either isotropic or anisotropic, and it is assumed to be isothermal. Because of this assumption, the strain energy potentials are expressed in terms of strain invariants. Unless indicated otherwise, the hyperelastic materials are also assumed to be nearly or purely incompressible. Material thermal expansion is also assumed to be isotropic.

Support is available for several options of strain energy potentials for simulating of incompressible or nearly incompressible hyperelastic materials. All options apply to the elements listed in [Material Model Element Support](#) for hyperelasticity.

The [Mooney-Rivlin hyperelasticity option](#) (**TB,MOONEY**) also applies to [explicit dynamic elements](#).

Tools are available to help you determine the coefficients for all of the hyperelastic options defined by **TB,HYPER**. The **TBFT** command allows you to compare your experimental data with existing material data curves and visually fit your curve for use in the **TB** command. All of the **TBFT** command capability (except for plotting) is available via batch and interactive (GUI) mode. See [Material Curve Fitting](#) for more information.

The following topics describing each of the hyperelastic options (**TB,HYPER,,,,TBOPT**) are available:

- 8.4.1.2.1. Mooney-Rivlin Hyperelastic Option (**TB,HYPER,,,,MOONEY**)
- 8.4.1.2.2. Ogden Hyperelastic Option (**TB,HYPER,,,,OGDEN**)
- 8.4.1.2.3. Neo-Hookean Hyperelastic Option (**TB,HYPER,,,,NEO**)
- 8.4.1.2.4. Polynomial Form Hyperelastic Option (**TB,HYPER,,,,POLY**)
- 8.4.1.2.5. Arruda-Boyce Hyperelastic Option (**TB,HYPER,,,,BOYCE**)
- 8.4.1.2.6. Gent Hyperelastic Option (**TB,HYPER,,,,GENT**)
- 8.4.1.2.7. Yeoh Hyperelastic Option (**TB,HYPER,,,,YEOH**)
- 8.4.1.2.8. Blatz-Ko Foam Hyperelastic Option (**TB,HYPER,,,,BLATZ**)
- 8.4.1.2.9. Ogden Compressible Foam Hyperelastic Option (**TB,HYPER,,,,FOAM**)
- 8.4.1.2.10. Response Function Hyperelastic Option (**TB,HYPER,,,,RESPONSE**)
- 8.4.1.2.11. User-Defined Hyperelastic Option (**TB,HYPER,,,,USER**)

#### 8.4.1.2.1. Mooney-Rivlin Hyperelastic Option (**TB,HYPER,,,,MOONEY**)

The Mooney-Rivlin option (**TB,HYPER,,,,MOONEY**), which is the default, allows you to define 2, 3, 5, or 9 parameters through the *NPTS* argument of the **TB** command. For example, to define a 5 parameter model you would issue **TB,HYPER,1,,5,MOONEY**.

The 2 parameter Mooney-Rivlin option has an applicable strain of about 100% in tension and 30% in compression. Compared to the other options, higher orders of the Mooney-Rivlin option may provide better approximation to a solution at higher strain.

The following example input listing shows a typical use of the Mooney-Rivlin option with 3 parameters:

```
TB,HYPER,1,,3,MOONEY      !Activate 3 parameter Mooney-Rivlin data table
TBDATA,1,0.163498        !Define c10
TBDATA,2,0.125076        !Define c01
TBDATA,3,0.014719        !Define c11
TBDATA,4,6.93063E-5      !Define incompressibility parameter
                          !(as 2/K, K is the bulk modulus)
```

Refer to [Mooney-Rivlin Hyperelastic Material \(\*\*TB,HYPER\*\*\)](#) for a description of the material constants required for this option.

#### 8.4.1.2.2. Ogden Hyperelastic Option (**TB,HYPER,,,,OGDEN**)

The Ogden option (**TB,HYPER,,,,OGDEN**) allows you to define an unlimited number of parameters via the *NPTS* argument of the **TB** command. For example, to define a three-parameter model, use **TB,HYPER,1,,3,OGDEN**.

Compared to the other options, the Ogden option usually provides the best approximation to a solution at larger strain levels. The applicable strain level can be up to 700 percent. A higher parameter value can provide a better fit to the exact solution. It may however cause numerical difficulties in fitting the material constants, and it requires enough data to cover the whole range of deformation for which you may be interested. For these reasons, a high parameter value is not recommended.

The following example input listing shows a typical use of the Ogden option with 2 parameters:

```
TB,HYPER,1,,2,OGDEN      !Activate 2 parameter Ogden data table
TBDATA,1,0.326996        !Define μ1
```

```
TB,DATA,2,2           !Define  $\alpha_1$ 
TB,DATA,3,-0.250152  !Define  $\mu_2$ 
TB,DATA,4,-2         !Define  $\alpha_2$ 
TB,DATA,5,6.93063E-5 !Define incompressibility parameter
                    !(as  $2/K$ ,  $K$  is the bulk modulus)
                    !(Second incompressibility parameter  $d_2$  is zero)
```

Refer to [Ogden Hyperelastic Material Constants](#) for a description of the material constants required for this option.

#### 8.4.1.2.3. Neo-Hookean Hyperelastic Option (TB,HYPER,,,,NEO)

The Neo-Hookean option (TB,HYPER,,,,NEO) represents the simplest form of strain energy potential, and has an applicable strain range of 20-30%.

An example input listing showing a typical use of the Neo-Hookean option is presented below.

```
TB,HYPER,1,,,NEO      !Activate Neo-Hookean data table
TB,DATA,1,0.577148    !Define mu shear modulus
TB,DATA,2,7.0e-5      !Define incompressibility parameter
                    !(as  $2/K$ ,  $K$  is the bulk modulus)
```

Refer to [Neo-Hookean Hyperelastic Material](#) for a description of the material constants required for this option.

#### 8.4.1.2.4. Polynomial Form Hyperelastic Option (TB,HYPER,,,,POLY)

The polynomial form option (TB,HYPER,,,,POLY) allows you to define an unlimited number of parameters through the *NPTS* argument of the TB command. For example, to define a 3 parameter model you would issue TB,HYPER,1,,3,POLY.

Similar to the higher order Mooney-Rivlin options, the polynomial form option may provide a better approximation to a solution at higher strain.

For *NPTS* = 1 and constant  $c_{01} = 0$ , the polynomial form option is equivalent to the Neo-Hookean option (see [Neo-Hookean Hyperelastic Option \(TB,HYPER,,,,NEO\)](#) (p. 204) for an example input listing). Also, for *NPTS* = 1, it is equivalent to the 2 parameter Mooney-Rivlin option. For *NPTS* = 2, it is equivalent to the 5 parameter Mooney-Rivlin option, and for *NPTS* = 3, it is equivalent to the 9 parameter Mooney-Rivlin option (see [Mooney-Rivlin Hyperelastic Option \(TB,HYPER,,,,MOONEY\)](#) (p. 203) for an example input listing).

Refer to [Polynomial Form Hyperelastic Material Constants](#) for a description of the material constants required for this option.

#### 8.4.1.2.5. Arruda-Boyce Hyperelastic Option (TB,HYPER,,,,BOYCE)

The Arruda-Boyce option (TB,HYPER,,,,BOYCE) has an applicable strain level of up to 300%.

An example input listing showing a typical use of the Arruda-Boyce option is presented below.

```
TB,HYPER,1,,,BOYCE    !Activate Arruda-Boyce data table
TB,DATA,1,200.0       !Define initial shear modulus
TB,DATA,2,5.0         !Define limiting network stretch
TB,DATA,3,0.001       !Define incompressibility parameter
                    !(as  $2/K$ ,  $K$  is the bulk modulus)
```

Refer to [Arruda-Boyce Hyperelastic Material Constants](#) for a description of the material constants required for this option.

#### 8.4.1.2.6. Gent Hyperelastic Option (TB,HYPER,,,,GENT)

The Gent option (TB,HYPER,,,,GENT) has an applicable strain level of up to 300%.

An example input listing showing a typical use of the Gent option is presented below.

```
TB,HYPER,1,,,GENT      !Activate Gent data table
TBDATA,1,3.0           !Define initial shear modulus
TBDATA,2,42.0          !Define limiting I1 - 3
TBDATA,3,0.001         !Define incompressibility parameter
                       !(as 2/K, K is the bulk modulus)
```

Refer to [Gent Hyperelastic Material Constants](#) for a description of the material constants required for this option.

#### 8.4.1.2.7. Yeoh Hyperelastic Option (TB,HYPER,,,,YEOH)

The Yeoh option (TB,HYPER,,,,YEOH) is a reduced polynomial form of the hyperelasticity option TB,HYPER,,,,POLY. An example of a 2 term Yeoh model is TB,HYPER,1,,2,YEOH.

Similar to the polynomial form option, the higher order terms may provide a better approximation to a solution at higher strain.

For  $NPTS = 1$ , the Yeoh form option is equivalent to the Neo-Hookean option (see [Neo-Hookean Hyperelastic Option \(TB,HYPER,,,,NEO\)](#) (p. 204) for an example input listing).

The following example input listing shows a typical use of the Yeoh option with 2 terms and 1 incompressibility term:

```
TB,HYPER,1,,2,YEOH     !Activate 2 term Yeoh data table
TBDATA,1,0.163498      !Define C1
TBDATA,2,0.125076      !Define C2
TBDATA,3,6.93063E-5    !Define first incompressibility parameter
```

Refer to [Yeoh Hyperelastic Material Constants](#) for a description of the material constants required for this option.

#### 8.4.1.2.8. Blatz-Ko Foam Hyperelastic Option (TB,HYPER,,,,BLATZ)

The Blatz-Ko option (TB,HYPER,,,,BLATZ) is the simplest option for simulating the compressible foam type of elastomer. This option is analogous to the Neo-Hookean option of incompressible hyperelastic materials.

An example input listing showing a typical use of the Blatz-Ko option is presented below.

```
TB,HYPER,1,,,BLATZ     !Activate Blatz-Ko data table
TBDATA,1,5.0           !Define initial shear modulus
```

Refer to [Blatz-Ko Foam Hyperelastic Material Constants](#) for a description of the material constants required for this option.

#### 8.4.1.2.9. Ogden Compressible Foam Hyperelastic Option (TB,HYPER,,,,FOAM)

The Ogden compressible foam option (TB,HYPER,,,,FOAM) simulates highly compressible foam material. An example of a 3 parameter model is TB,HYPER,1,,3,FOAM. Compared to the Blatz-Ko option, the Ogden foam option usually provides the best approximation to a solution at larger strain levels. The higher the number of parameters, the better the fit to the experimental data. It may however cause numerical difficulties in fitting the material constants, and it requires sufficient data to cover the whole range of

deformation for which you may be interested. For these reasons, a high parameter value is not recommended.

The following example input listing shows a typical use of the Ogden foam option with two parameters:

```

TB,HYPER,1,,2,FOAM      !Activate 2 parameter Ogden foam data table
TBDATA,1,1.85          !Define  $\mu_1$ 
TBDATA,2,4.5           !Define  $\alpha_1$ 
TBDATA,3,-9.20        !Define  $\mu_2$ 
TBDATA,4,-4.5         !Define  $\alpha_2$ 
TBDATA,5,0.92         !Define first compressibility parameter
TBDATA,6,0.92         !Define second compressibility parameter

```

Refer to [Ogden Compressible Foam Hyperelastic Material Constants](#) for a description of the material constants required for this option.

#### 8.4.1.2.10. Response Function Hyperelastic Option (TB,HYPER,,,,RESPONSE)

The response function hyperelastic option (TB,HYPER,,,,RESPONSE) works with experimental data (TB,EXPE).

The TB,HYPER command's *NPTS* argument defines the number of terms in the volumetric potential function. The data table includes entries for the deformation limit cutoff for the stiffness matrix as the first entry, and the volumetric potential function incompressibility parameters starting in the third position. (The second position in the data table is unused.)

The following example input shows the use of the response function option with two terms in the volumetric potential function:

```

TB,HYPER,1,,2,RESPONSE ! Activate Response Function data table
TBDATA,1,1E-4          ! Define deformation limit cutoff
TBDATA,3,0.002         ! Define first incompressibility parameter
TBDATA,4,0.00001      ! Define second incompressibility parameter

```

For a description of the material constants required for this option, see [Response Function Hyperelasticity](#) in the *Material Reference*. For detailed information about response functions determined via experimental data, see [Experimental Response Functions](#) in the *Mechanical APDL Theory Reference*.

Experimental data for the model is entered via the TB,EXPE command, where the *TBOPT* argument specifies the type of data to be input:

- Uniaxial tension is input via *TBOPT* = UNITENSION.
- Uniaxial compression is input via *TBOPT* = UNICOMPRESSION.
- Equibiaxial tension is input via *TBOPT* = BIAXIAL.
- Planar shear is input via *TBOPT* = SHEAR
- Combined uniaxial tension and compression is input via *TBOPT* = UNIAXIAL.

The response function hyperelastic model must include experimental data for at least one of the listed deformations.

Any combination of uniaxial tension, equibiaxial tension, or planar shear is also valid. For incompressible and nearly incompressible materials, uniaxial compression can be used in place of equibiaxial tension.

Combined uniaxial tension and compression data can be used to model material behavior that is different in tension than in compression, but such data can only be used in combination with pressure-volume experimental data.

Volumetric behavior is specified with either experimental data or a polynomial volumetric potential function. Incompressible behavior results if no volumetric model or data is given.

Volumetric experimental data is input as two values per data point with volume ratio as the independent variable and pressure as the dependent variable. For uniaxial tension, uniaxial compression, equibiaxial tension, planar shear, and combined uniaxial tension and compression deformations, the experimental data is entered in either of these formats:

- Two values per data point: engineering strain as the independent variable and engineering stress as the dependent variable
- Three values per data point: engineering strain in the loading direction as the independent variable, engineering strain in the lateral direction as the first dependent variable, and engineering stress as the second dependent variable. For uniaxial compression data, the lateral strain is ignored and incompressibility is assumed for the experimental data.

The input format must be consistent within the table for an individual experimental deformation, but can change between tables for different experimental deformations.

For example, incompressible uniaxial tension and planar shear data are used as input to the response function hyperelastic material defined above. Three experimental data points for incompressible uniaxial deformation are input with the following commands:

```

TB,EXPERIMENTAL,1,,UNITENSION ! Activate uniaxial data table
TBFIELD,TEMP,21 ! Temperature for following data
TBPT,, 0.0, 0.0 !
TBPT,, 0.2, 1.83 ! 1st data point
TBPT,, 1.0, 5.56 ! 2nd data point
TBPT,, 4.0, 17.6 ! 3rd data point

```

Four experimental data points for incompressible planar shear deformation are input with the following commands:

```

TB,EXPERIMENTAL,1,,SHEAR ! Activate planar shear data table
TBFIELD,TEMP,21 ! Temperature for following data
TBPT,, 0.0, 0.0 !
TBPT,, 0.24, 2.69 ! 1st data point
TBPT,, 0.96, 6.32 ! 2nd data point
TBPT,, 4.2, 19.7 ! 3rd data point
TBPT,, 5.1, 27.4 ! 4th data point

```

Combined uniaxial tension and compression are input via `TBOPT = UNIAXIAL` on the `TB,EXPE` command, as shown in this example:

```

TB,EXPE,1,,UNIAXIAL ! Activate uniaxial data table
TBFIELD,TEMP,21 ! Temperature for following data
TBPT,, 0.0, 0.0 !
TBPT,, 0.01, 0.0915 ! 1st tension data point
TBPT,, 0.2, 1.83 ! 2nd tension data point
TBPT,, 1.0, 5.56 ! 3rd tension data point
TBPT,, -0.01, -0.0915 ! 1st compression data point
TBPT,, -0.2, -3.66 ! 2nd compression data point
TBPT,, -1.0, -22.24 ! 3rd compression data point

```

For all input data, the zero stress-strain point should be entered as a data point; otherwise, interpolation or extrapolation of the data to zero strain should yield a value of zero stress. For combined tension and compression data, the initial slope of the data should be the same in tension and compression.



Data outside the experimental strain values are assumed to be constant; therefore, all experimental data should cover the simulated deformation range as measured by the first deformation invariant (expressed by Equation 4.196 in the *Mechanical APDL Theory Reference*).

The following table shows various  $I_1$  values and the corresponding engineering strains in each experimental deformation for an incompressible material:

Example $I_1$ Values and Corresponding Experimental Strains				
$I_1$	3.01	3.1	4.0	10.0
<b>Uniaxial tension</b>	0.059	0.193	0.675	2.057
<b>Biaxial tension</b>	0.030	0.098	0.362	1.232
<b>Uniaxial compression</b>	-0.057	-0.171	-0.461	-0.799
<b>Planar shear</b>	0.051	0.171	0.618	1.981

For example, a simulation that includes deformation up to  $I_1 = 10.0$  requires experimental data in uniaxial tension up to about 206 percent engineering strain, biaxial tension to 123 percent, uniaxial compression to -80 percent, and planar shear to 198 percent. The values in the table were obtained by solving Equation 4.252 for uniaxial tension, Equation 4.261 for biaxial tension, Equation 4.268 for planar shear (all described in the *Mechanical APDL Theory Reference*), and converting the biaxial tension strain to equivalent uniaxial compression strain.

Experimental data that does not include the lateral strain are assumed to be for incompressible material behavior; however, this data can be combined with a volumetric potential function to simulate the behavior of nearly incompressible materials. Combining incompressible experimental data with a volumetric model that includes significant compressibility is not restricted, but should be considered carefully before use in a simulation.

#### 8.4.1.2.11. User-Defined Hyperelastic Option (TB,HYPER,,,,USER)

The User option (TB,HYPER,,,,USER) allows you to use the subroutine USERHYPER to define the derivatives of the strain energy potential with respect to the strain invariants. Refer to the *Guide to User-Programmable Features* for a detailed description on writing a user hyperelasticity subroutine.

#### 8.4.1.3. Bergstrom-Boyce Hyperviscoelastic Material Model

Use the Bergstrom-Boyce material model (TB,BB) for modeling the strain-rate-dependent, hysteretic behavior of materials that undergo substantial elastic and inelastic strains. Examples of such materials include elastomers and biological materials. The model assumes an inelastic response only for shear distortional behavior; the response to volumetric deformations is still purely elastic.

The following example input listing shows a typical use of the Bergstrom-Boyce option:

```
TB, BB, 1, , , ISO           !Activate Bergstrom-Boyce ISO data table
TBDATA, 1, 1.31             !Define material constant  $\mu_A$  ,
TBDATA, 2, 9.0              !Define  $N0=(\lambda_A^{lock})^2$ 
TBDATA, 3, 4.45             !Define material constant  $\mu_B$ 
TBDATA, 4, 9.0              !Define  $N1=(\lambda_B^{lock})^2$ 
TBDATA, 5, 0.33             !Define material constant
TBDATA, 6, -1               !Define material constant  $c$ 
TBDATA, 7, 5.21            !Define material constant  $m$ 
!
TB, BB, 1, , , PVOL         !Activate Bergstrom-Boyce PVOL data table
TBDATA, 1, 0.001           ! as  $1/K$ ,  $K$  is the bulk modulus
```



## Additional Information

For a description of the material constants required for this option, see [Bergstrom-Boyce Material](#) in the [Material Reference](#). For more detailed information about this material model, see the documentation for the **TB,BB** command, and [Bergstrom-Boyce](#) in the [Mechanical APDL Theory Reference](#).

### 8.4.1.4. Mullins Effect Material Model

Use the Mullins effect option (**TB,CDM**) for modeling load-induced changes to constitutive response exhibited by some hyperelastic materials. Typical of filled polymers, the effect is most evident during cyclic loading where the unloading response is more compliant than the loading behavior. The condition causes a hysteresis in the stress-strain response and is a result of irreversible changes in the material.

The Mullins effect option is used with any of the nearly- and fully-incompressible isotropic [hyperelastic](#) constitutive models (all **TB,HYPER** options with the exception of  $TBOPT = BLATZ$  or  $TBOPT = FOAM$ ) and modifies the behavior of those models. The Mullins effect model is based on maximum previous load, where the load is the strain energy of the virgin hyperelastic material. As the maximum previous load increases, changes to the virgin hyperelastic constitutive model due to the Mullins effect also increase. Below the maximum previous load, the Mullins effect changes are not evolving; however, the Mullins effect still modifies the hyperelastic constitutive response based on the maximum previous load.

To select the modified Ogden-Roxburgh pseudo-elastic Mullins effect model, use the **TB** command to set  $TBOPT = PSE2$ . The pseudo-elastic model results in a scaled stress given by

$$S_{ij} = \eta S_{ij}^0$$

where  $\eta$  is a damage variable.

The functional form of the modified Ogden-Roxburgh damage variable is

$$\eta = 1 - \frac{1}{r} \operatorname{erf} \left[ \frac{W_m - W_0}{m + \beta W_m} \right], \quad (TBOPT = PSE2)$$

where  $W_m$  is the maximum previous strain energy and  $W_0$  is the strain energy for the virgin hyperelastic material. The modified Ogden-Roxburgh damage function requires and enforces  $NPTS = 3$  with the three material constants  $r$ ,  $m$ , and  $\beta$ .

Select the material constants to ensure  $\eta \in (0, 1]$  over the range of application. This condition is guaranteed for  $r > 0$ ,  $m > 0$ , and  $\beta \geq 0$ ; however, it is also guaranteed by the less stringent bounds  $r > 0$ ,  $m > 0$ , and  $(m + \beta W_m) > 0$ . The latter bounds are solution-dependent, so you must ensure that the limits for  $\eta$  are not violated if  $\beta < 0$ .

Following is an example input fragment for the modified Ogden-Roxburgh pseudo-elastic Mullins effect model:

```
TB,CDM,1,,3,PSE2           !Modified Ogden Roxburgh pseudo-elastic
TBDATA,1,1.5,1.0E6,0.2     !Define r, m, and beta
```

## Additional Information

For a description of the material constants required for this option, see [Mullins Effect](#) in the [Material Reference](#). For more detailed information about this material model, see the documentation for the **TB,CDM** command, and [Mullins Effect](#) in the [Mechanical APDL Theory Reference](#).

### 8.4.1.5. Anisotropic Hyperelasticity Material Model

You can use anisotropic hyperelasticity to model the directional differences in material behavior. This is especially useful when modeling elastomers with reinforcements, or for biomedical materials such as muscles or arteries. You use the format **TB,AHYPER,,,TBOPT** to define the material behavior.

The TBOPT field allows you to specify the isochoric part, the material directions and the volumetric part for the material simulation. You must define one single TB table for each option.

You can enter temperature dependent data for anisotropic hyperelastic material with the **TBTEMP** command. For the first temperature curve, you issue **TB,AHYPER,,,TBOPT**, then input the first temperature using the **TBTEMP** command. The subsequent **TBDATA** command inputs the data.

See the **TB** command, and [Anisotropic Hyperelasticity](#) in the *Mechanical APDL Theory Reference* for more information.

The following example shows the definition of material constants for an anisotropic hyperelastic material option:

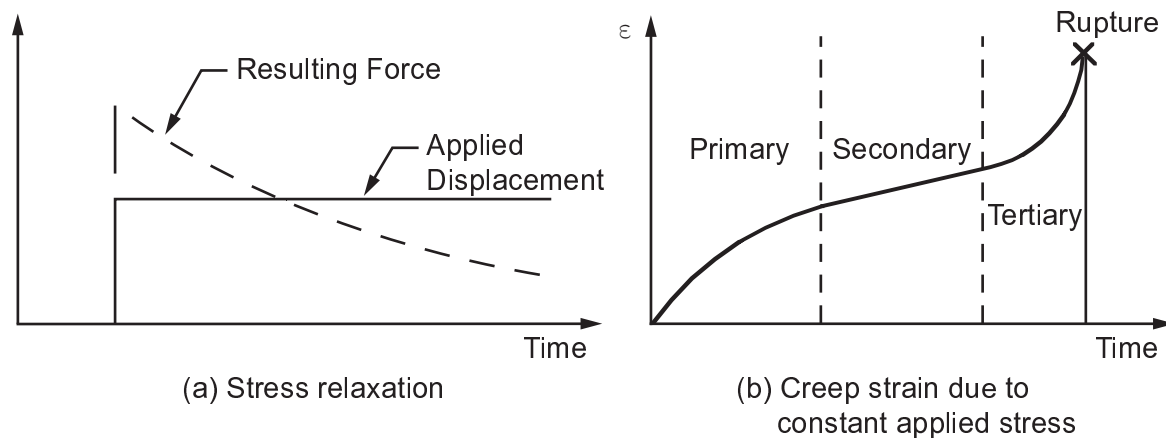
```
! defining material constants for anistoropic hyperelastic option
tb,ahyper,1,1,31,poly
! a1,a2,a3
tbdata,1,10,2,0.1
! b1,b2,b3
tbdata,4,5,1,0.1
! c2,c3,c4,c5,c6
tbdata,7,1,0.02,0.002,0.001,0.0005
! d2,d3,d4,d5,d6
tbdata,12,1,0.02,0.002,0.001,0.0005
! e2,e3,e4,e5,e6
tbdata,17,1,0.02,0.002,0.001,0.0005
! f2,f3,f4,f5,f6
tbdata,22,1,0.02,0.002,0.001,0.0005
! g2,g3,g4,g5,g6
tbdata,27,1,0.02,0.002,0.001,0.0005

!compressibility parameter d
tb,ahyper,1,1,1,pvol
tbdata,1,1e-3

!orientation vector A=A(x,y,z)
tb,ahyper,1,1,3,avec
tbdata,1,1,0,0
!orientation vector B=B(x,y,z)
tb,ahyper,1,1,3,bvec
tbdata,1,1/sqrt(2),1/sqrt(2),0
```

### 8.4.1.6. Creep Material Model

*Creep* is a rate-dependent material nonlinearity in which the material continues to deform under a constant load. Conversely, if a displacement is imposed, the reaction force (and stresses) diminish over time (stress relaxation; see [Figure 8.10: Stress Relaxation and Creep \(p. 211\)\(a\)](#)). The three stages of creep are shown in [Figure 8.10: Stress Relaxation and Creep \(p. 211\)\(b\)](#). The program has the capability of modeling the first two stages (primary and secondary). The tertiary stage is usually not analyzed since it implies impending failure.

**Figure 8.10: Stress Relaxation and Creep**

Creep is important in high temperature stress analyses, such as for nuclear reactors. For example, suppose you apply a preload to some part in a nuclear reactor to keep adjacent parts from moving. Over a period of time at high temperature, the preload would decrease (stress relaxation) and potentially let the adjacent parts move. Creep can also be significant for some materials such as prestressed concrete. Typically, the creep deformation is permanent.

The program analyzes creep using two time-integration methods. Both are applicable to static or transient analyses. The *implicit* creep method is robust, fast, accurate, and recommended for general use. It can handle temperature dependent creep constants, as well as simultaneous coupling with isotropic hardening plasticity models. The *explicit* creep method is useful for cases where very small time steps are required. Creep constants cannot be dependent on temperature. Coupling with other plastic models is available by superposition only.

The terms *implicit* and *explicit* as applied to creep have no relationship to [explicit dynamic analysis](#) or to any elements referred to as “explicit dynamic elements.”

The creep strain rate may be a function of stress, strain, temperature, and neutron flux level. Built-in libraries of creep strain rate equations are used for primary, secondary, and irradiation induced creep. (See [Creep Equations](#) for discussions of, and input procedures for, these various creep equations.) Some equations require specific units. For the explicit creep option in particular, temperatures used in the creep equations should be based on an absolute scale.

The following topics related to creep are available:

[8.4.1.6.1. Implicit Creep Procedure](#)

[8.4.1.6.2. Explicit Creep Procedure](#)

### 8.4.1.6.1. Implicit Creep Procedure

The basic procedure for using the implicit creep method involves issuing the **TB** command with *Lab* = CREEP, and choosing a creep equation by specifying a value for *TBOPT*. The following example input shows the use of the implicit creep method. *TBOPT* = 2 specifies that the primary creep equation for model 2 is used. Temperature dependency is specified using the **TBTEMP** command, and the four constants associated with this equation are specified as arguments with the **TBDATA** command.

```
TB, CREEP, 1, 1, 4, 2
TBTEMP, 100
TBDATA, 1, C1, C2, C3, C4
```

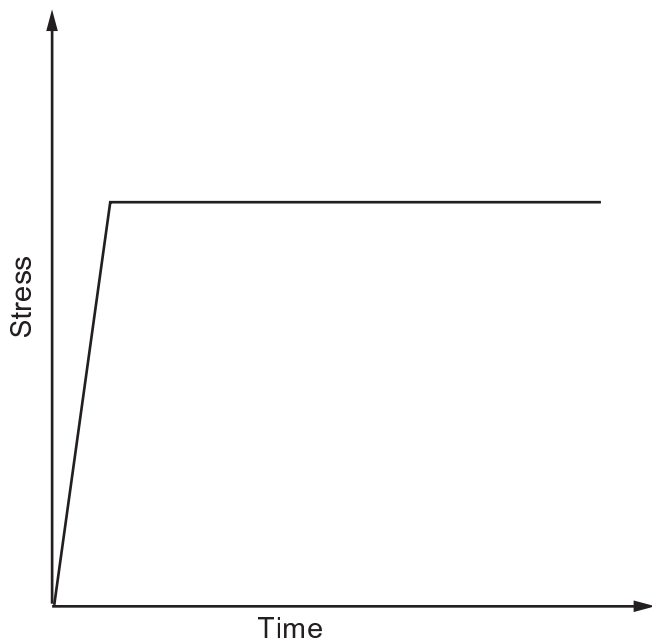
You can input other creep expressions using the user programmable feature and setting  $TBOPT = 100$ . You can define the number of state variables using the **TB** command with  $Lab = STATE$ . The following example shows how five state variables are defined.

```
TB,STATE,1,,5
```

You can simultaneously model creep (**TB,CREEP**) and isotropic, bilinear kinematic, and Hill anisotropy options to simulate more complex material behaviors. See [Material Model Combinations](#) for the combination possibilities. Also, see [Material Model Combination Examples](#) (p. 216).

To perform an implicit creep analysis, you must also issue the solution **RATE** command, with  $Option = ON$  (or 1). The following example shows a procedure for a time hardening creep analysis. (See [Figure 8.11: Time Hardening Creep Analysis](#) (p. 212).)

**Figure 8.11: Time Hardening Creep Analysis**



The user applied mechanical loading in the first load step, and turned the **RATE** command OFF to bypass the creep strain effect. Since the time period in this load step affects the total time thereafter, the time period for this load step should be small. For this example, the user specified a value of 1.0E-8 seconds. The second load step is a creep analysis. The **RATE** command must be turned ON. Here the mechanical loading was kept constant, and the material creeps as time increases.

```
/SOLU                !First load step, apply mechanical loading
RATE,OFF             !Creep analysis turned off
TIME,1.0E-8         !Time period set to a very small value
...
SOLV                 !Solve this load step
                    !Second load step, no further mechanical load
RATE,ON              !Creep analysis turned on
TIME,100             !Time period set to desired value
...
SOLV                 !Solve this load step
```

The **RATE** command works only when modeling *implicit* creep with either von Mises or Hill potentials.

For most materials, the creep strain rate changes significantly at an early stage. Because of this, a general recommendation is to use a small initial incremental time step, then specify a large maximum incremental time step by using solution command **DELTIM** or **NSUBST**. For implicit creep, you may need

to examine the effect of the time increment on the results carefully because the program does not enforce any creep ratio control by default. You can always enforce a creep limit ratio using the creep ratio control option in commands **CRPLIM** or **CUTCONTROL,CRPLIMIT**. A recommended value for a creep limit ratio ranges from 1 to 10. The ratio may vary with materials so your decision on the best value to use should be based on your own experimentation to gain the required performance and accuracy. For larger analyses, a suggestion is to first perform a time increment convergence analysis on a simple small size test.

Tools are available to help you determine the coefficients for all of the implicit creep options defined in **TB,CREEP**. The **TBFT** command allows you to compare your experimental data with existing material data curves and visually fit your curve for use in the **TB** command. All of the **TBFT** command capability (except for plotting) is available via batch and interactive (GUI) mode. See [Material Curve Fitting](#) for more information.

#### 8.4.1.6.2. Explicit Creep Procedure

The basic procedure for using the explicit creep method involves issuing the **TB** command with *Lab* = CREEP and choosing a creep equation by adding the appropriate constant as an argument with the **TBDATA** command. *TBOPT* is either left blank or = 0. The following example input uses the explicit creep method. Note that all constants are included as arguments with the **TBDATA** command, and that there is no temperature dependency.

```
TB,CREEP,1
TBDATA,1,C1,C2,C3,C4, ,C6
```

For the explicit creep method, you can incorporate other creep expressions into the program by using User Programmable Features (see the [Guide to User-Programmable Features](#)).

For highly nonlinear creep strain vs. time curves, a small time step must be used with the explicit creep method. Creep strains are not computed if the time step is less than 1.0e-6. A creep time step optimization procedure is available (**AUTOTS** and **CRPLIM**) for automatically adjusting the time step as appropriate.

#### 8.4.1.7. Shape Memory Alloy Material Model

The **Shape Memory Alloy (TB,SMA)** material behavior option describes the superelastic behavior of nitinol alloy. Nitinol is a flexible metal alloy that can undergo very large deformations in loading-unloading cycles without permanent deformation. The material behavior has three distinct phases: an austenite phase (linear elastic), a martensite phase (also linear elastic), and the transition phase between these two.

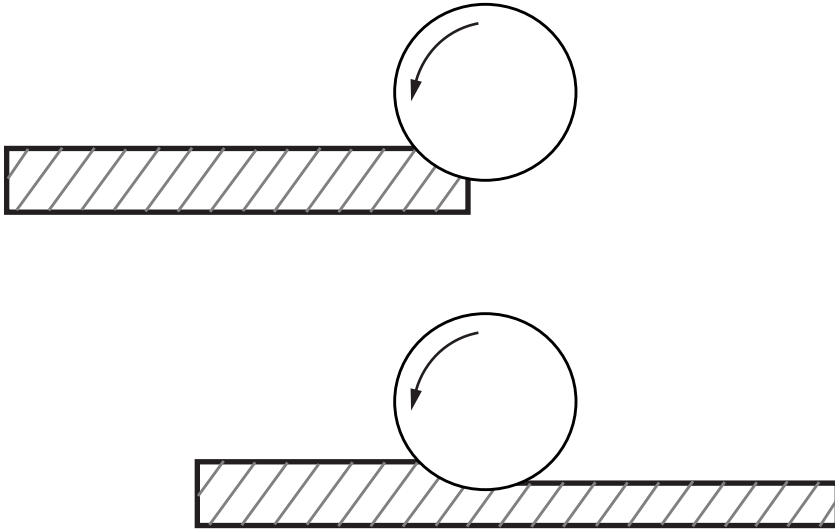
For more information, see [Shape Memory Alloy \(SMA\)](#) in the [Material Reference](#).

#### 8.4.1.8. Viscoplasticity

*Viscoplasticity* is a time-dependent plasticity phenomenon, where the development of the plastic strain is dependent on the rate of loading. The primary application is high-temperature metal-forming (such as rolling and deep drawing) which involves large plastic strains and displacements with small elastic strains. (See [Figure 8.12: Viscoplastic Behavior in a Rolling Operation \(p. 214\)](#).)

Viscoplasticity is defined by unifying [plasticity](#) and [creep](#) via a set of flow and evolutionary equations. A constraint equation preserves volume in the plastic region.

For more information about modeling viscoplasticity, see [Nonlinear Stress-Strain Materials](#).

**Figure 8.12: Viscoplastic Behavior in a Rolling Operation**

### Rate-Dependent Plasticity (Viscoplasticity)

The **TB,RATE** command option allows you to introduce the strain rate effect in material models to simulate the time-dependent response of materials. Typical applications include metal forming and microelectromechanical systems (MEMS).

The Perzyna, Peirce, Anand and Chaboche material options (described in [Rate-Dependent Plasticity](#) in the *Mechanical APDL Theory Reference*) are available, as follows:

- **Perzyna and Peirce options**

Unlike other rate-dependent material options (such as [creep](#) or the [Anand option](#)), the Perzyna and Peirce options include a yield surface. The plasticity, and thus the strain rate hardening effect, is active only after plastic yielding. For more information, see [Perzyna and Peirce Options](#) in the *Material Reference*.

To simulate viscoplasticity, use the Perzyna and Peirce models in combination with the **TB** command's BISO, MISO, or NLISO material options. Further, you can simulate anisotropic viscoplasticity by combining the HILL option. (See [Material Model Combinations](#) for combination possibilities. For example input listings of material combinations, see [Material Model Combination Examples \(p. 216\)](#) in this guide.)

For isotropic hardening, the intent is to simulate the strain rate hardening of materials rather than softening. Large-strain analysis is supported.

- **Exponential Visco-Hardening (EVH) option**

The exponential visco-hardening (EVH) rate-dependent material option uses explicit functions to define the static yield stresses of materials and therefore does not need to combine with other plastic options (such as BIO, MISO, NLISO, and PLASTIC) to define it. For more information, see [Exponential Visco-Hardening \(EVH\) Option](#) in the *Material Reference*.

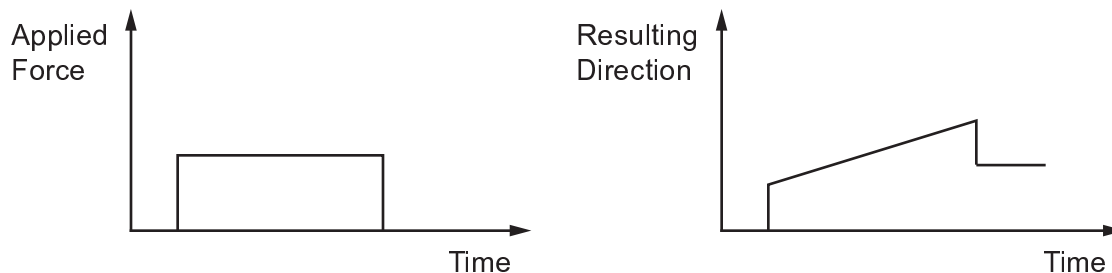
- **Anand option**

The Anand rate-dependent material option offers a unified plasticity model requiring no combination with other material models. For more information, see [Anand Option](#) in the *Material Reference*.

### 8.4.1.9. Viscoelasticity

Viscoelasticity is similar to creep, but part of the deformation is removed when the loading is taken off. A common viscoelastic material is glass. Some plastics are also considered to be viscoelastic. One type of viscoelastic response is illustrated in [Figure 8.13: Viscoelastic Behavior \(Maxwell Model\)](#) (p. 215).

**Figure 8.13: Viscoelastic Behavior (Maxwell Model)**



Small- and large-deformation models are available to model viscoelasticity in the time domain. The small-deformation model can be used in the harmonic domain to model the steady-state response of viscoelastic materials.

The elasticity constants correspond to the material behavior at the fast load limit. The elastic constants are specified as follows (assuming the use of current-technology elements):

- For small-deformation hypoelasticity, issue either the **MP** or **TB,ELASTIC** command.
- For large-deformation hyperelasticity, issue the **TB,HYPER** command.

Specify the bulk and shear relaxation properties via the **TB,PRONY** command. Specify the time-temperature superposition properties via the **TB,SHIFT** command.

```
!Small Strain Viscoelasticity
mp,ex,1,20.0E5 !elastic properties
mp,nuxy,1,0.3

tb,prony,1,,2,shear !define viscosity parameters (shear)
tbdata,1,0.5,2.0,0.25,4.0
tb,prony,1,,2,bulk !define viscosity parameters (bulk)
tbdata,1,0.5,2.0,0.25,4.0

!Large Strain Viscoelasticity
tb,hyper,1,,moon !elastic properties
tbdata,1,38.462E4,,1.2E-6

tb,prony,1,,1,shear !define viscosity parameters
tbdata,1,0.5,2.0
tb,prony,1,,1,bulk !define viscosity parameters
tbdata,1,0.5,2.0
```

For use in the harmonic domain, an alternative method for specifying the viscoelastic properties is available. You can input the properties using experimental values of the complex moduli (**TB,EXPE**).

See [Viscoelastic Material Constants](#) and the [Mechanical APDL Theory Reference](#) for details about how to input viscoelastic material properties using the **TB** family of commands.

Tools are available to help you determine the relaxation properties (defined via **TB,PRONY**). The **TBFT** command allows you to compare your experimental data with existing material data curves and visually fit your curve for use in the **TB** command. All of the **TBFT** command capability (except for plotting) is available via batch and interactive (GUI) mode. See [Material Curve Fitting](#) for more information.



### 8.4.1.10. Swelling

Certain materials respond to neutron flux by enlarging volumetrically, or *swelling*. To include swelling effects, initialize a material data table for swelling (**TB**,**SWELL**) and define the swelling constants (**TB****DATA**).

Several swelling model options (**TB**,**SWELL**,,,,**TBOPT**) are available, including a user-defined option (via subroutine `userswstrain`, described in *Guide to User-Programmable Features*).

#### Example 8.1: Defining Linear Swelling with Two Temperatures

```
!define elastic properties
mp,ex,1,20.0E5 !elastic properties
mp,nuxy,1,0.3

tb,swell,1,2,1,linear !define linear swelling option with two temperatures
tbtemp,25
tbdata,1,0.1e-5
tbtemp,100
tbdata,1,0.2e-5
```

For more information about using the **TB**,**SWELL** and the **TB** family of commands to input constants for the swelling equations, see *Swelling* in the *Material Reference*.

Swelling can also be related to other phenomena, such as moisture content. The commands for defining nuclear swelling can be used analogously to define swelling due to other causes.

### 8.4.1.11. User-Defined Material Model

The **User-Defined** material model (**TB**,**USER**) describes input parameters for defining your own material model via the `UserMat` subroutine.

For more information about user-defined materials, see *User-Defined Materials*, and *Subroutine UserMat (Creating Your Own Material Model)* in the *Guide to User-Programmable Features*.

## 8.4.2. Material Model Combination Examples

You can combine several material model options to simulate complex material behaviors. *Material Model Combinations* presents the model options you can combine along with the associated **TB** command labels and links to example input listings.

The following example input listings are presented in sections identified by the **TB** command labels.

- 8.4.2.1. RATE and CHAB and BISO Example
- 8.4.2.2. RATE and CHAB and MISO Example
- 8.4.2.3. RATE and CHAB and PLAS (Multilinear Isotropic Hardening) Example
- 8.4.2.4. RATE and CHAB and NLISO Example
- 8.4.2.5. BISO and CHAB Example
- 8.4.2.6. MISO and CHAB Example
- 8.4.2.7. PLAS (Multilinear Isotropic Hardening) and CHAB Example
- 8.4.2.8. NLISO and CHAB Example
- 8.4.2.9. PLAS (Multilinear Isotropic Hardening) and EDP Example
- 8.4.2.10. MISO and EDP Example
- 8.4.2.11. GURSON and BISO Example
- 8.4.2.12. GURSON and MISO Example
- 8.4.2.13. GURSON and PLAS (MISO) Example
- 8.4.2.14. NLISO and GURSON Example
- 8.4.2.15. RATE and BISO Example



- 8.4.2.16. MISO and RATE Example
- 8.4.2.17. RATE and PLAS (Multilinear Isotropic Hardening) Example
- 8.4.2.18. RATE and NLISO Example
- 8.4.2.19. BISO and CREEP Example
- 8.4.2.20. MISO and CREEP Example
- 8.4.2.21. PLAS (Multilinear Isotropic Hardening) and CREEP Example
- 8.4.2.22. NLISO and CREEP Example
- 8.4.2.23. BKIN and CREEP Example
- 8.4.2.24. HILL and BISO Example
- 8.4.2.25. HILL and MISO Example
- 8.4.2.26. HILL and PLAS (Multilinear Isotropic Hardening) Example
- 8.4.2.27. HILL and NLISO Example
- 8.4.2.28. HILL and BKIN Example
- 8.4.2.29. HILL and MKIN Example
- 8.4.2.30. HILL and KINH Example
- 8.4.2.31. HILL, and PLAS (Kinematic Hardening) Example
- 8.4.2.32. HILL and CHAB Example
- 8.4.2.33. HILL and BISO and CHAB Example
- 8.4.2.34. HILL and MISO and CHAB Example
- 8.4.2.35. HILL and PLAS (Multilinear Isotropic Hardening) and CHAB Example
- 8.4.2.36. HILL and NLISO and CHAB Example
- 8.4.2.37. HILL and RATE and BISO Example
- 8.4.2.38. HILL and RATE and MISO Example
- 8.4.2.39. HILL and RATE and NLISO Example
- 8.4.2.40. HILL and CREEP Example
- 8.4.2.41. HILL, CREEP and BISO Example
- 8.4.2.42. HILL and CREEP and MISO Example
- 8.4.2.43. HILL, CREEP and PLAS (Multilinear Isotropic Hardening) Example
- 8.4.2.44. HILL and CREEP and NLISO Example
- 8.4.2.45. HILL and CREEP and BKIN Example
- 8.4.2.46. HYPER and VISCO (Hyperelasticity and Viscoelasticity (Implicit)) Example
- 8.4.2.47. AHYPER and PRONY (Anisotropic Hyperelasticity and Viscoelasticity (Implicit)) Example
- 8.4.2.48. EDP and CREEP and PLAS (MISO) Example
- 8.4.2.49. CAP and CREEP and PLAS (MISO) Example
- 8.4.2.50. CHAB and CREEP Example
- 8.4.2.51. CHAB and CREEP and NLISO Example
- 8.4.2.52. CHAB and CREEP and HILL Example
- 8.4.2.53. CHAB and CREEP and HILL and MISO Example

### **8.4.2.1. RATE and CHAB and BISO Example**

This input listing illustrates an example of combining viscoplasticity and Chaboche nonlinear kinematic hardening plasticity and bilinear isotropic hardening plasticity.

```

MP,EX,1,185.0E3                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,RATE,1,,PERZYNA            ! RATE TABLE
TBDATA,1,0.5,1

TB,CHAB,1                      ! CHABOCHE TABLE
TBDATA,1,180,100,3

TB,BISO,1                      ! BISO TABLE
TBDATA,1,180,200

```

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#), and [Viscoplasticity \(p. 213\)](#) in this document.

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### **8.4.2.2. RATE and CHAB and MISO Example**

This input listing illustrates an example of combining viscoplasticity and Chaboche nonlinear kinematic hardening plasticity and multilinear isotropic hardening plasticity.

```
MP,EX,1,185E3                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,RATE,1,, ,PERZYNA        ! RATE TABLE
TBDATA,1,0.5,1

TB,CHAB,1                    ! CHABOCHE TABLE
TBDATA,1,180,100,3        ! THIS EXAMPLE ISOTHERMAL

TB,MISO,1                    ! MISO TABLE
TBPT,,9.7E-4,180
TBPT,,1.0,380
```

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#), and the RATE option, see [Viscoplasticity \(p. 213\)](#), and in this document.

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### **8.4.2.3. RATE and CHAB and PLAS (Multilinear Isotropic Hardening) Example**

In addition to the **TB,MISO** example (above), you can also use material plasticity, the multilinear isotropic hardening option - **TB,PLAS,,,MISO** to combine viscoplasticity and Chaboche nonlinear kinematic hardening plasticity. An example of the combination is as follows:

```
MP,EX,1,185E3                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,RATE,1,, ,PERZYNA        ! RATE TABLE
TBDATA,1,0.5,1

TB,CHAB,1                    ! CHABOCHE TABLE
TBDATA,1,180,100,3        ! THIS EXAMPLE ISOTHERMAL

TB,PLAS,,,MISO              ! MISO TABLE
TBPT,,0.0,180
TBPT,,0.99795,380
```

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#), and the RATE option, see [Viscoplasticity \(p. 213\)](#), and in this document.

For information about the PLAS option, see [Multilinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.4. RATE and CHAB and NLISO Example

This input listing illustrates an example of combining viscoplasticity and Chaboche nonlinear kinematic hardening plasticity and nonlinear isotropic hardening plasticity.

```

MP,EX,1,20.0E5                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,RATE,1,,PERZYNA           ! RATE TABLE
TBDATA,1,0.5,1

TB,CHAB,1,3,5                ! CHABOCHE TABLE
TBTEMP,20,1                  ! THIS EXAMPLE TEMPERATURE DEPENDENT
TBDATA,1,500,20000,100,40000,200,10000
TBDATA,7,1000,200,100,100,0
TBTEMP,40,2
TBDATA,1,880,204000,200,43800,500,10200
TBDATA,7,1000,2600,2000,500,0
TBTEMP,60,3
TBDATA,1,1080,244000,400,45800,700,12200
TBDATA,7,1400,3000,2800,900,0

TB,NLISO,1,2                 ! NLISO TABLE
TBTEMP,40,1
TBDATA,1,880,0.0,80.0,3
TBTEMP,60,2
TBDATA,1,1080,0.0,120.0,7

```

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#), and [Viscoplasticity \(p. 213\)](#) in this document.

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.5. BISO and CHAB Example

This input listing illustrates an example of combining bilinear isotropic hardening plasticity with Chaboche nonlinear kinematic hardening plasticity.

```

MP,EX,1,185.0E3              ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,CHAB,1                    ! CHABOCHE TABLE
TBDATA,1,180,100,3

TB,BISO,1                     ! BISO TABLE
TBDATA,1,180,200

```

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.6. MISO and CHAB Example

This input listing illustrates an example of combining multilinear isotropic hardening plasticity with Chaboche nonlinear kinematic hardening plasticity.

```

MP,EX,1,185E3                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,CHAB,1                    ! CHABOCHE TABLE
TBDATA,1,180,100,3          ! THIS EXAMPLE ISOTHERMAL

```

```

TB,MISO,1                                ! MISO TABLE
TBPT,,9.7E-4,180
TBPT,,1.0,380
    
```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.7. PLAS (Multilinear Isotropic Hardening) and CHAB Example

In addition to the **TB,MISO** example (above), you can also use material plasticity. The multilinear isotropic hardening option - **TB,PLAS,,,MISO** is combined with Chaboche nonlinear kinematic hardening in the following example:

```

MP,EX,1,185E3                            ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,CHAB,1                                 ! CHABOCHE TABLE
TBDATA,1,180,100,3                       ! THIS EXAMPLE ISOTHERMAL

TB,PLAS,,,MISO                           ! MISO TABLE
TBPT,,0.0,180
TBPT,,0.99795,380
    
```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.8. NLISO and CHAB Example

This input listing illustrates an example of combining nonlinear isotropic hardening plasticity with Chaboche nonlinear kinematic hardening plasticity.

```

MP,EX,1,20.0E5                            ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,CHAB,1,3,5                             ! CHABOCHE TABLE
TBTEMP,20,1                               ! THIS EXAMPLE TEMPERATURE DEPENDENT
TBDATA,1,500,20000,100,40000,200,10000
TBDATA,7,1000,200,100,100,0
TBTEMP,40,2
TBDATA,1,880,204000,200,43800,500,10200
TBDATA,7,1000,2600,2000,500,0
TBTEMP,60,3
TBDATA,1,1080,244000,400,45800,700,12200
TBDATA,7,1400,3000,2800,900,0

TB,NLISO,1,2                              ! NLISO TABLE
TBTEMP,40,1
TBDATA,1,880,0.0,80.0,3
TBTEMP,60,2
TBDATA,1,1080,0.0,120.0,7
    
```

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.9. PLAS (Multilinear Isotropic Hardening) and EDP Example

You can use the **TB,PLAS** capability in conjunction with Extended Drucker-Prager plasticity. The multilinear isotropic hardening option - **TB,PLAS,,,MISO** is combined with Extended Drucker-Prager plasticity in the following example:

```

/prep7
mp,ex,1,2.1e4    ! Elastic Properties
mp,nuxy,1,0.1

ys=7.894657
s1=1000.0

tb,edp,1,, ,LYFUN
tbdata,1,2.2526,ys

tb,edp,1,, ,LFPOT
tbdata,1,0.566206

tb,plas,1,1,2,miso
tbpt,defi,0.0,7.894
tbpt,defi,1,1007.894

```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the EDP option, see [Extended Drucker-Prager](#).

### 8.4.2.10. MISO and EDP Example

The **TB**,MISO option can also be used to combine multilinear isotropic hardening with Extended Drucker-Prager plasticity, as shown in the following example:

```

/prep7
mp,ex,1,2.1e4    ! Elastic Properties
mp,nuxy,1,0.1

ys=7.894657
s1=1000.0

tb,edp,1,, ,LYFUN
tbdata,1,2.2526,ys

tb,edp,1,, ,LFPOT
tbdata,1,0.566206

tb,miso,1,1,2
tbpt,defi,0.000375905,7.894
tbpt,defi,1,1.047994952,1007.894

```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the EDP option, see [Extended Drucker-Prager](#).

### 8.4.2.11. GURSON and BISO Example

The **TB**,BISO option can also be used to combine bilinear isotropic hardening with Gurson plasticity, as shown in the following example:

```

q1=1.5
q2=1
q3=q1*q1
sigma_Y=E/300.0
Yield=1.0/sigma_Y
rone=1.0
rthree=3.0
f_0= 0.04
f_N= 0.04
S_N=0.1
strain_N=0.3
Power_N=0.1

tb,GURS,1,, ,5,BASE          ! Gurson's BASE model
tbdata,1,sigma_Y,f_0,q1,q2,q3

```

```
tb,GURS,1,,3,SNNU          ! Gurson's SNNU model
tbdata,1,f_N,strain_N,S_N

TB,BISO,1                  ! BISO TABLE
TBDATA,1,Yield, Power_N
```

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

For information about the GURSON option, see [Gurson's Model](#).

### 8.4.2.12. GURSON and MISO Example

The **TB,MISO** option can also be used to combine multilinear isotropic hardening with Gurson plasticity, as shown in the following example:

```
Young=1000000
sigma_Y=Young/300.0
yield=1.0d0/sigma_Y/3.1415926
! define elastic Properties
mp,ex,1,Young
mp,nuxy,1,0.3
! Define Gurson's coefficients
q1=1.5
q2=1
q3=q1*q1
f_0= 0.000000
f_N= 0.04
S_N=0.1
strain_N=0.3
Power_N=0.1
f_c=0.15
f_F=0.25
! Gurson Model
tb,gurs,1,,5,BASE ! BASE DEFINED
tbdata,1,sigma_Y,f_0,q1,q2,q3

tb,gurs,1,,3,SNNU ! SNNU DEFINED
tbdata,1,f_N,strain_N,S_N

tb,gurs,1,,2,COAL ! COAL DEFINED
tbdata,1,f_c,f_F

tb,miso,,6
tbpt,,0.003333333, 3333.333333
tbpt,,0.018982279, 3966.666667
tbpt,,0.103530872, 4700
tbpt,,0.562397597, 5566.666667
tbpt,,1.006031106, 5900
tbpt,,2.934546576, 6566.666667
```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the GURSON option, see [Gurson's Model](#).

### 8.4.2.13. GURSON and PLAS (MISO) Example

The **TB,PLAS,,,MISO** option can also be used to combine multilinear isotropic hardening with Gurson plasticity, as shown in the following example:

```
q1=1.5
q2=1
q3=q1*q1
sigma_Y=E/300.0
Yield=1.0/sigma_Y
rone=1.0
```

```

rthree=3.0
f_0= 0.04
f_N= 0.04
S_N=0.1
strain_N=0.3
Power_N=0.1

tb,GURS,1,,5,BASE           ! Gurson's BASE model
tbdata,1,sigma_Y,f_0,q1,q2,q3

tb,GURS,1,,3,SNNU          ! Gurson's SNNU model
tbdata,1,f_N,strain_N,S_N

tb,plas,1,,4,miso
tbpt, defl, 0.0, Yield
tbpt, defl, 1, 10.0*Yield

```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the GURSON option, see [Gurson's Model](#).

#### 8.4.2.14. NLISO and GURSON Example

The **TB,NLISO** option can also be used to combine nonlinear isotropic hardening with Gurson plasticity, as shown in the following example:

```

q1=1.5
q2=1
q3=q1*q1
sigma_Y=E/300.0
Yield=1.0/sigma_Y
rone=1.0
rthree=3.0
f_0= 0.04
f_N= 0.04
S_N=0.1
strain_N=0.3
Power_N=0.1

tb,GURS,1,,5,BASE           ! Gurson's BASE model
tbdata,1,sigma_Y,f_0,q1,q2,q3

tb,GURS,1,,3,SNNU          ! Gurson's SNNU model
tbdata,1,f_N,strain_N,S_N

tb,nliso,1,1,2,POWER
tbdata,1,sigma_Y,power_N

```

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

For information about the GURSON option, see [Gurson's Model](#).

#### 8.4.2.15. RATE and BISO Example

This input listing illustrates an example of combining bilinear isotropic hardening plasticity with the **TB,RATE** command to model viscoplasticity.

```

MP,EX,1,20.0E5             ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,BISO,1                  ! BISO TABLE
TBDATA,1,9000,10000

TB,RATE,1,, ,PERZYNA      ! RATE TABLE
TBDATA,1,0.5,1

```

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#), and [Viscoplasticity](#) (p. 213).

#### 8.4.2.16. MISO and RATE Example

This input listing illustrates an example of combining multilinear isotropic hardening plasticity with the **TB,RATE** command to model viscoplasticity.

```
MP,EX,1,20.0E5           ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,MISO,1                ! MISO TABLE
TBPT,,0.015,30000
TBPT,,0.020,32000
TBPT,,0.025,33800
TBPT,,0.030,35000
TBPT,,0.040,36500
TBPT,,0.050,38000
TBPT,,0.060,39000

TB,RATE,1,, ,PERZYNA    ! RATE TABLE
TBDATA,1,0.5,1
```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#), and [Viscoplasticity](#) (p. 213).

#### 8.4.2.17. RATE and PLAS (Multilinear Isotropic Hardening) Example

In addition to the **TB,MISO** example (above), you can also use material plasticity. The multilinear isotropic hardening option - **TB,PLAS,,,MISO** is combined with RATE-dependent viscoplasticity in the following example:

```
MP,EX,1,20.0E5           ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,PLAS,,,MISO          ! MISO TABLE
TBPT,,0.00000,30000
TBPT,,4.00E-3,32000
TBPT,,8.10E-3,33800
TBPT,,1.25E-2,35000
TBPT,,2.18E-2,36500
TBPT,,3.10E-2,38000
TBPT,,4.05E-2,39000

TB,RATE,1,, ,PERZYNA    ! RATE TABLE
TBDATA,1,0.5,1
```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#), and [Viscoplasticity](#) (p. 213).

#### 8.4.2.18. RATE and NLISO Example

This input listing illustrates an example of combining nonlinear isotropic hardening plasticity with the **TB,RATE** command to model viscoplasticity.

```
MP,EX,1,20.0E5           ! ELASTIC CONSTANTS
MP,NUXY,1,0.3
```



```

TB,NLISO,1                                ! NLISO TABLE
TBDATA,1,30000,100000,5200,172

TB,RATE,1,,PERZYNA                        ! RATE TABLE
TBDATA,1,0.5,1

```

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials and Viscoplasticity \(p. 213\)](#).

#### 8.4.2.19. BISO and CREEP Example

This input listing illustrates an example of combining bilinear isotropic hardening plasticity with implicit creep.

```

MP,EX,1,20.0E5                            ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,BISO,1                                  ! BISO TABLE
TBDATA,1,9000,10000

TB,CREEP,1,,2                              ! CREEP TABLE
TBDATA,1,1.5625E-14,5.0,-0.5,0.0

```

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

#### 8.4.2.20. MISO and CREEP Example

This input listing illustrates an example of combining multilinear isotropic hardening plasticity with implicit creep.

```

MP,EX,1,20.0E5                            ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,MISO,1                                  ! MISO TABLE
TBPT,,0.015,30000
TBPT,,0.020,32000
TBPT,,0.025,33800
TBPT,,0.030,35000
TBPT,,0.040,36500
TBPT,,0.050,38000
TBPT,,0.060,39000

TB,CREEP,1,,2                              ! CREEP TABLE
TBDATA,1,1.5625E-14,5.0,-0.5,0.0

```

For information about the MISO option, see [Multilinear Isotropic Hardening..](#)

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

#### 8.4.2.21. PLAS (Multilinear Isotropic Hardening) and CREEP Example

In addition to the **TB,MISO** example (above), you can also use material plasticity. The multilinear isotropic hardening option - **TB,PLAS,,,MISO** is combined with implicit CREEP in the following example:

```

MP,EX,1,20.0E5                            ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

```

```

TB,PLAS,,,,MISO                ! MISO TABLE
TBPT,,0.00000,30000
TBPT,,4.00E-3,32000
TBPT,,8.10E-3,33800
TBPT,,1.25E-2,35000
TBPT,,2.18E-2,36500
TBPT,,3.10E-2,38000
TBPT,,4.05E-2,39000

TB,CREEP,1,,,2                ! CREEP TABLE
TBDATA,1,1.5625E-14,5.0,-0.5,0.0
    
```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

### 8.4.2.22. NLISO and CREEP Example

This input listing illustrates an example of combining nonlinear isotropic hardening plasticity with implicit creep.

```

MP,EX,1,20.0E5                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,NLISO,1                    ! NLISO TABLE
TBDATA,1,30000,100000,5200,172

TB,CREEP,1,,,2                ! CREEP TABLE
TBDATA,1,1.5625E-14,5.0,-0.5,0.0
    
```

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

### 8.4.2.23. BKIN and CREEP Example

This input listing illustrates an example of combining bilinear kinematic hardening plasticity with implicit creep.

```

MP,EX,1,1e7                   ! ELASTIC CONSTANTS
MP,NUXY,1,0.32

TB,BKIN,1                     ! BKIN TABLE
TBDATA,1,42000,1000

TB,CREEP,1,,,6                ! CREEP TABLE
TBDATA,1,7.4e-21,3.5,0,0,0,0
    
```

For information about the BKIN option, see [Bilinear Kinematic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

### 8.4.2.24. HILL and BISO Example

This input listing illustrates an example of modeling anisotropic plasticity with bilinear isotropic hardening.

```

MP,EX,1,20.0E5                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3
    
```

```

TB,HILL,1,2                                ! HILL TABLE
TBTEMP,100
TBDATA,1,1,1.0402,1.24897,1.07895,1,1
TBTEMP,200
TBDATA,1,0.9,0.94,1.124,0.97,0.9,0.9

TB,BISO,1,2                                ! BISO TABLE
TBTEMP,100
TBDATA,1,461.0,374.586
TBTEMP,200
TBDATA,1,400.0,325.0

```

For information about the HILL option, see [Generalized Hill](#).

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

### 8.4.2.25. HILL and MISO Example

This input listing illustrates an example of modeling anisotropic plasticity with multilinear isotropic hardening.

```

MP,EX,1,20.0E5                             ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,MISO,1                                   ! MISO TABLE
TBPT,,0.015,30000
TBPT,,0.020,32000
TBPT,,0.025,33800
TBPT,,0.030,35000
TBPT,,0.040,36500
TBPT,,0.050,38000
TBPT,,0.060,39000

TB,HILL,1                                   ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80

```

For information about the HILL option, see [Generalized Hill](#).

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

### 8.4.2.26. HILL and PLAS (Multilinear Isotropic Hardening) Example

In addition to the **TB,MISO** example (above), you can also use material plasticity. The multilinear isotropic hardening option - **TB,PLAS,,,MISO** is combined with HILL anisotropic plasticity in the following example:

```

MP,EX,1,20.0E5                             ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,PLAS,,,MISO                             ! MISO TABLE
TBPT,,0.00000,30000
TBPT,,4.00E-3,32000
TBPT,,8.10E-3,33800
TBPT,,1.25E-2,35000
TBPT,,2.18E-2,36500
TBPT,,3.10E-2,38000
TBPT,,4.05E-2,39000

TB,HILL,1                                   ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80

```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the HILL option, see [Generalized Hill](#).

### 8.4.2.27. HILL and NLISO Example

This input listing illustrates an example of modeling anisotropic plasticity with nonlinear isotropic hardening.

```
MP,EX,1,20.0E5          ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,NLISO,1              ! NLISO TABLE
TBDATA,1,30000,100000,5200,172

TB,HILL,1               ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80
```

For information about the HILL option, see [Generalized Hill](#).

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

### 8.4.2.28. HILL and BKIN Example

This input listing illustrates an example of modeling anisotropic plasticity with bilinear kinematic hardening.

```
MP,EX,1,20.0E5          ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,BKIN,1               ! BKIN TABLE
TBDATA,1,9000,10000

TB,HILL,1               ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80
```

For information about the HILL option, see [Generalized Hill](#).

For information about the BKIN option, see [Bilinear Kinematic Hardening](#).

### 8.4.2.29. HILL and MKIN Example

This input listing illustrates an example of modeling anisotropic plasticity with multilinear kinematic hardening.

```
MPTEMP,1,20,400,650,800,950      ! ELASTIC CONSTANTS

MPDATA,EX,1,1,30.00E6,27.36E6,25.20E6,23.11E6,20.76E6
MPDATA,EY,1,1,30.00E6,27.36E6,25.20E6,23.11E6,20.76E6
MPDATA,EZ,1,1,30.00E6,27.36E6,25.20E6,23.11E6,20.76E6

MPDATA,PRXY,1,1,0.351,0.359,0.368,0.375,0.377
MPDATA,PRYZ,1,1,0.351,0.359,0.368,0.375,0.377
MPDATA,PRXZ,1,1,0.351,0.359,0.368,0.375,0.377

MPDATA,GXY,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4
MPDATA,GYZ,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4
MPDATA,GXZ,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4

TB,MKIN,1,5,5              ! MKIN TABLE
TBTEMP,,strain
TBDATA,1,0.0015,0.006,0.04,0.08,0.1
TBTEMP,20
TBDATA,1,45000,60000,90000,115000,120000
TBTEMP,400
TBDATA,1,41040,54720,82080,104880,109440
TBTEMP,650
TBDATA,1,37800,50400,75600,96600,100800
```

```

TBTEMP,800
TBDATA,1,34665,46220,69330,88588,92440
TBTEMP,950
TBDATA,1,31140,41520,62280,79580,83040

```

```

TB,HILL,1,5                ! HILL TABLE
TBTEMP,20.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,400.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,650.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,800.0
TBDATA,1,1.0,1.0,1.0,1.00,1.00,1.00
TBTEMP,950.0
TBDATA,1,1.0,1.0,1.0,1.00,1.00,1.00

```

For information about the HILL option, see [Generalized Hill](#).

For information about the MKIN option, see [Multilinear Kinematic Hardening](#).

### 8.4.2.30. HILL and KINH Example

This input listing illustrates an example of modeling anisotropic plasticity with multilinear kinematic hardening.

```

MP,EX,1,20E6                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,KINH,1,,3                ! KINH TABLE
TBPT,,5E-5,1E3
TBPT,,0.01,2E3
TBPT,,0.60,6E4

TB,HILL,1                    ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.90,0.95

```

For information about the HILL option, see [Generalized Hill](#).

For information about the KINH option, see [Multilinear Kinematic Hardening](#).

### 8.4.2.31. HILL, and PLAS (Kinematic Hardening) Example

In addition to the **TB,KINH** example (above), you can also use material plasticity. The kinematic hardening option - **TB,PLAS,,,KINH** is combined with HILL anisotropic plasticity in the following example:

```

MP,EX,1,20E6                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,PLAS,,,KINH             ! KINH TABLE
TBPT,,0.00000,1E3
TBPT,,9.90E-3,2E3
TBPT,,5.97E-1,6E4

TB,HILL,1

```

For information about the HILL option, see [Generalized Hill](#).

For information about the KINH option, see [Multilinear Kinematic Hardening](#).

### 8.4.2.32. HILL and CHAB Example

This input listing illustrates an example of modeling anisotropic plasticity with Chaboche nonlinear kinematic hardening.

```
MP,EX,1,185E3           ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,CHAB,1              ! CHABOCHE TABLE
TBDATA,1,180,400,3,0

TB,HILL,1              ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80
```

For information about the HILL option, see [Generalized Hill](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.33. HILL and BISO and CHAB Example

This input listing illustrates an example of modeling anisotropic plasticity with bilinear isotropic hardening and Chaboche nonlinear kinematic hardening.

```
MP,EX,1,185E3           ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,CHAB,1              ! CHABOCHE TABLE
TBDATA,1,180,100,3

TB,BISO,1              ! BISO TABLE
TBDATA,1,180,200

TB,HILL,1              ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80
```

For information about the HILL option, see [Generalized Hill](#).

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.34. HILL and MISO and CHAB Example

This input listing illustrates an example of modeling anisotropic plasticity with multilinear isotropic hardening and Chaboche nonlinear kinematic hardening.

```
MP,EX,1,185E3           ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,CHAB,1              ! CHABOCHE TABLE
TBDATA,1,185,100,3

TB,MISO,1              ! MISO TABLE
TBPT,,0.001,185
TBPT,,1.0,380

TB,HILL,1              ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80
```

For information about the HILL option, see [Generalized Hill](#).

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.35. HILL and PLAS (Multilinear Isotropic Hardening) and CHAB Example

In addition to the **TB,MISO** example (above), you can also use material plasticity. The multilinear isotropic hardening option - **TB,PLAS,,,MISO** is combined with HILL anisotropic plasticity and Chaboche nonlinear kinematic hardening in the following example:

```
MP,EX,1,185E3                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,CHAB,1                    ! CHABOCHE TABLE
TBDATA,1,185,100,3

TB,PLAS,,,,MISO              ! MISO TABLE
TBPT,,0.001,185
TBPT,,0.998,380

TB,HILL,1                    ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80
```

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

For information about the HILL option, see [Generalized Hill](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### 8.4.2.36. HILL and NLISO and CHAB Example

This input listing illustrates an example of combining anisotropic plasticity with nonlinear isotropic hardening and Chaboche nonlinear kinematic hardening.

```
MPTEMP,1,20,200,400,550,600,650          ! ELASTIC CONSTANTS
MPTEMP,,700,750,800,850,900,950
!
MPDATA,EX,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EX,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4
!
MPDATA,EY,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EY,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4
!
MPDATA,EZ,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EZ,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4
!
MPDATA,PRXY,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRXY,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,PRYZ,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRYZ,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,PRXZ,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRXZ,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,GXY,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GXY,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4
!
MPDATA,GYZ,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GYZ,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4
!
MPDATA,GXZ,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GXZ,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4

TB,NLISO,1                      ! NLISO TABLE
TBDATA,1,180,0.0,100.0,5
```

```

!
TB, CHAB, 1                                ! CHABOCHE TABLE
TBDATA, 1, 180, 100, 3

TB, HILL, 1, 5                             ! HILL TABLE
TBTEMP, 750.0
TBDATA, 1, 1.0, 1.0, 1.0, 0.93, 0.93, 0.93
TBTEMP, 800.0
TBDATA, 1, 1.0, 1.0, 1.0, 0.93, 0.93, 0.93
TBTEMP, 850.0
TBDATA, 1, 1.0, 1.0, 1.0, 0.93, 0.93, 0.93
TBTEMP, 900.0
TBDATA, 1, 1.0, 1.0, 1.0, 1.00, 1.00, 1.00
TBTEMP, 950.0
TBDATA, 1, 1.0, 1.0, 1.0, 1.00, 1.00, 1.00
    
```

For information about the HILL option, see [Generalized Hill](#).

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

### **8.4.2.37. HILL and RATE and BISO Example**

This input listing illustrates an example of modeling anisotropic viscoplasticity with bilinear isotropic hardening plasticity.

```

MPTEMP, 1, 20, 400, 650, 800, 950        ! ELASTIC CONSTANTS
!
MPDATA, EX, 1, 1, 30.00E6, 27.36E6, 25.20E6, 23.11E6, 20.76E6
!
MPDATA, EY, 1, 1, 30.00E6, 27.36E6, 25.20E6, 23.11E6, 20.76E6
!
MPDATA, EZ, 1, 1, 30.00E6, 27.36E6, 25.20E6, 23.11E6, 20.76E6
!
MPDATA, PRXY, 1, 1, 0.351, 0.359, 0.368, 0.375, 0.377
!
MPDATA, PRYZ, 1, 1, 0.351, 0.359, 0.368, 0.375, 0.377
!
MPDATA, PRXZ, 1, 1, 0.351, 0.359, 0.368, 0.375, 0.377
!
MPDATA, GXY, 1, 1, 1.190E4, 1.160E4, 1.110E4, 1.080E4, 1.060E4
!
MPDATA, GYZ, 1, 1, 1.190E4, 1.160E4, 1.110E4, 1.080E4, 1.060E4
!
MPDATA, GXZ, 1, 1, 1.190E4, 1.160E4, 1.110E4, 1.080E4, 1.060E4

TB, BISO, 1,                                ! BISO TABLE
TBDATA, 1, 45000, 760000

TB, RATE, 1, 2, , PERZYNA                 ! RATE TABLE
TBTEMP, 20
TBDATA, 1, 0.1, 0.3
TBTEMP, 950
TBDATA, 1, 0.3, 0.5

TB, HILL, 1, 5                             ! HILL TABLE
TBTEMP, 750.0
TBDATA, 1, 1.0, 1.0, 1.0, 0.93, 0.93, 0.93
TBTEMP, 800.0
TBDATA, 1, 1.0, 1.0, 1.0, 0.93, 0.93, 0.93
TBTEMP, 850.0
    
```



```
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,900.0
TBDATA,1,1.0,1.0,1.0,1.00,1.00,1.00
TBTEMP,950.0
TBDATA,1,1.0,1.0,1.0,1.00,1.00,1.00
```

For information about the HILL option, see [Generalized Hill](#).

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#), and [Viscoplasticity](#) (p. 213).

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

### 8.4.2.38. HILL and RATE and MISO Example

This input listing illustrates an example of modeling anisotropic viscoplasticity with multilinear isotropic hardening plasticity.

```
MP,EX,1,20.0E5                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,MISO,1                    ! MISO TABLE
TBPT,,0.015,30000
TBPT,,0.020,32000
TBPT,,0.025,33800
TBPT,,0.030,35000
TBPT,,0.040,36500
TBPT,,0.050,38000
TBPT,,0.060,39000

TB,HILL,1                    ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80

TB,RATE,1,,PERZYNA          ! RATE TABLE
TBDATA,1,0.5,1
```

For information about the HILL option, see [Generalized Hill](#).

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#) and [Viscoplasticity](#) (p. 213).

For information about the MISO option, see [Bilinear Isotropic Hardening](#).

### 8.4.2.39. HILL and RATE and NLISO Example

This input listing illustrates an example of modeling anisotropic viscoplasticity with nonlinear isotropic hardening plasticity.

```
MP,EX,1,20.0E5                ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,NLISO,1                   ! NLISO TABLE
TBDATA,1,30000,100000,5200,172

TB,HILL,1                    ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80

TB,RATE,1,,PERZYNA          ! RATE TABLE
TBDATA,1,0.5,1
```

For information about the HILL option, see [Generalized Hill](#).

For information about the RATE option, see [Rate-Dependent Viscoplastic Materials](#) and [Viscoplasticity](#) (p. 213).

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

### 8.4.2.40. HILL and CREEP Example

This input listing illustrates an example of modeling anisotropic implicit creep.

```

MPTEMP,1,20,200,400,550,600,650           ! ELASTIC CONSTANTS
MPTEMP,,700,750,800,850,900,950
!
MPDATA,EX,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EX,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4
!
MPDATA,EY,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EY,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4
!
MPDATA,EZ,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EZ,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4
!
MPDATA,PRXY,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRXY,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,PRYZ,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRYZ,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,PRXZ,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRXZ,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,GXY,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GXY,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4
!
MPDATA,GYZ,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GYZ,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4
!
MPDATA,GXZ,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GXZ,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4

TB,CREEP,1,,2                               ! CREEP TABLE
TBDATA,1,5.911E-34,6.25,-0.25

TB,HILL,1,5                                 ! HILL TABLE
TBTEMP,750.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,800.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,850.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,900.0
TBDATA,1,1.0,1.0,1.0,1.00,1.00,1.00
TBTEMP,950.0
TBDATA,1,1.0,1.0,1.0,1.00,1.00,1.00

```

For information about the HILL option, see [Generalized Hill](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

### 8.4.2.41. HILL, CREEP and BISO Example

This input listing illustrates an example of modeling anisotropic implicit creep with bilinear isotropic hardening plasticity.

```

MPTEMP,1,20,200,400,550,600,650           ! ELASTIC CONSTANTS
MPTEMP,,700,750,800,850,900,950
!
MPDATA,EX,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EX,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4

```

```

!
MPDATA,EY,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EY,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4
!
MPDATA,EZ,1,1,1.250E4,1.210E4,1.140E4,1.090E4,1.070E4,1.050E4
MPDATA,EZ,1,,1.020E4,0.995E4,0.963E4,0.932E4,0.890E4,0.865E4
!
MPDATA,PRXY,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRXY,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,PRYZ,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRYZ,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,PRXZ,1,1,0.351,0.359,0.368,0.375,0.377,0.380
MPDATA,PRXZ,1,,0.382,0.384,0.386,0.389,0.391,0.393
!
MPDATA,GXY,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GXY,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4
!
MPDATA,GYZ,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GYZ,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4
!
MPDATA,GXZ,1,1,1.190E4,1.160E4,1.110E4,1.080E4,1.060E4,1.040E4
MPDATA,GXZ,1,,1.020E4,1.000E4,0.973E4,0.946E4,0.908E4,0.887E4

```

```

TB,BISO,1                                ! BISO TABLE
TBDATA,1,180,200

```

```

TB,CREEP,1,,2                            ! CREEP TABLE
TBDATA,1,5.911E-34,6.25,-0.25

```

```

TB,HILL,1,5                              ! HILL TABLE
TBTEMP,750.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,800.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,850.0
TBDATA,1,1.0,1.0,1.0,0.93,0.93,0.93
TBTEMP,900.0
TBDATA,1,1.0,1.0,1.0,1.00,1.00,1.00
TBTEMP,950.0
TBDATA,1,1.0,1.0,1.0,1.00,1.00,1.00

```

For information about the HILL option, see [Generalized Hill](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

For information about the BISO option, see [Bilinear Isotropic Hardening](#).

### 8.4.2.42. HILL and CREEP and MISO Example

This input listing illustrates an example of modeling anisotropic implicit creep with multilinear isotropic hardening plasticity.

```

MP,EX,1,20.0E5                            ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,MISO,1                                  ! MISO TABLE
TBPT,,0.015,30000
TBPT,,0.020,32000
TBPT,,0.025,33800
TBPT,,0.030,35000
TBPT,,0.040,36500
TBPT,,0.050,38000
TBPT,,0.060,39000

```

```

TB,HILL,1                                ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80

TB,CREEP,1,,2                            ! CREEP TABLE
TBDATA,1,1.5625E-14,5.0,-0.5,0.0
    
```

For information about the HILL option, see [Generalized Hill](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

### 8.4.2.43. HILL, CREEP and PLAS (Multilinear Isotropic Hardening) Example

In addition to the **TB,MISO** example (above), you can also use material plasticity. The multilinear isotropic hardening option - **TB,PLAS,,,MISO** is combined with HILL anisotropic plasticity and implicit CREEP in the following example:

```

MP,EX,1,20.0E5                          ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,PLAS,1,,7,MISO                        ! MISO TABLE
TBPT,,0.00000,30000
TBPT,,4.00E-3,32000
TBPT,,8.10E-3,33800
TBPT,,1.25E-2,35000
TBPT,,2.18E-2,36500
TBPT,,3.10E-2,38000
TBPT,,4.05E-2,39000

TB,HILL,1                                ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80

TB,CREEP,1,,2                            ! CREEP TABLE
TBDATA,1,1.5625E-14,5.0,-0.5,0.0
    
```

For information about the HILL option, see [Generalized Hill](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

### 8.4.2.44. HILL and CREEP and NLISO Example

This input listing illustrates an example of modeling anisotropic implicit creep with nonlinear isotropic hardening plasticity.

```

MP,EX,1,20.0E5                          ! ELASTIC CONSTANTS
MP,NUXY,1,0.3

TB,NLISO,1                               ! NLISO TABLE
TBDATA,1,30000,100000,5200,172

TB,HILL,1                                ! HILL TABLE
TBDATA,1,1.0,1.1,0.9,0.85,0.9,0.80

TB,CREEP,1,,2                            ! CREEP TABLE
TBDATA,1,1.5625E-14,5.0,-0.5,0.0
    
```

For information about the HILL option, see [Generalized Hill](#).

For information about the CREEP option, see [Implicit Creep Equations](#), and [Implicit Creep Procedure](#) (p. 211).

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

#### **8.4.2.45. HILL and CREEP and BKIN Example**

This input listing illustrates an example of modeling anisotropic implicit creep with bilinear kinematic hardening plasticity.

```
MP,EX,1,1e7                ! ELASTIC CONSTANTS
MP,NUXY,1,0.32

TB,BKIN,1                  ! BKIN TABLE
TBDATA,1,42000,1000

TB,CREEP,1,,6             ! CREEP TABLES
TBDATA,1,7.4e-21,3.5,0,0,0,0

TB,HILL,1                  ! HILL TABLE
TBDATA,1,1.15,1.05,1.0,1.0,1.0,1.0
```

For information about the HILL option, see [Generalized Hill](#).

For information about the CREEP option, see [Implicit Creep Equations](#), and [Implicit Creep Procedure](#) (p. 211).

For information about the BKIN option, see [Bilinear Kinematic Hardening](#).

#### **8.4.2.46. HYPER and VISCO (Hyperelasticity and Viscoelasticity (Implicit)) Example**

This input listing illustrates the combination of implicit hyperelasticity and viscoelasticity.

```
c10=293
c01=177
TB,HYPER,1,,MOON          !!!! type 1 is Mooney-Rivlin
TBDATA,1,c10,c01
a1=0.1
a2=0.2
a3=0.3
t1=10
t2=100
t3=1000
tb,prony,1,,3,shear      ! define Prony constants
tbdata,1,a1,t1,a2,t2,a3,t3
```

For information about hyperelasticity, see [Hyperelastic Material Constants](#), and [Hyperelasticity Material Model](#) (p. 202).

For information about the viscoelasticity, see [Viscoelastic Material Constants](#), and [Viscoelasticity](#) (p. 215).

#### **8.4.2.47. AHYPER and PRONY (Anisotropic Hyperelasticity and Viscoelasticity (Implicit)) Example**

This input listing illustrates the combination of anisotropic hyperelasticity and viscoelasticity.

```
! defininig material constants for anistoropic hyperelastic option with TB,AHYPER command
tb,ahyper,1,1,31,poly
! a1,a2,a3
tbdata,1,10,2,0.1
! b1,b2,b3
tbdata,4,5,1,0.1
! c2,c3,c4,c5,c6
```

```

tbdata,7,1,0.02,0.002,0.001,0.0005
! d2,d3,d4,d5,d6
tbdata,12,1,0.02,0.002,0.001,0.0005
! e2,e3,e4,e5,e6
tbdata,17,1,0.02,0.002,0.001,0.0005
! f2,f3,f4,f5,f6
tbdata,22,1,0.02,0.002,0.001,0.0005
! g2,g3,g4,g5,g6
tbdata,27,1,0.02,0.002,0.001,0.0005

!compressibility parameter d
tb,ahyper,1,1,1,pvol
tbdata,1,1e-3

!orientation vector A=A(x,y,z)
tb,ahyper,1,1,3,avec
tbdata,1,1,0,0
!orientation vector B=B(x,y,z)
tb,ahyper,1,1,3,bvec
tbdata,1,1/sqrt(2),1/sqrt(2),0

! defininig material constants for Prony series with TB,PRONY command
a1=0.1
a2=0.2
a3=0.3
t1=10
t2=100
t3=1000
tb,prony,1,,3,hear          ! define Prony constants
tbdata,1,a1,t1,a2,t2,a3,t3

```

For information about anisotropic hyperelasticity, see [Anisotropic Hyperelastic Material Constants \(TB,AHYPER\)](#) in the *Material Reference*, and [Anisotropic Hyperelasticity Material Model \(p. 210\)](#) in this document.

Viscoelastic behavior is assumed to be isotropic. For information about the viscoelasticity, see [Viscoelastic Material Model](#) in the *Material Reference*, and [Viscoelasticity \(p. 215\)](#) in this document.

## 8.4.2.48. EDP and CREEP and PLAS (MISO) Example

This input listing illustrates an example of modeling Extended Drucker-Prager with implicit creep and with multilinear hardening.

```

ys=100.0
alpha=0.1
!
!define edp for material 1
!
tb,edp,1,,LYFUN
tbdata,1,alpha,ys
tb,edp,1,,LFPOT
tbdata,1,alpha
!
!define miso hardening for material 1
!
tb,plastic,1,,2,miso
tbpt,defi,0.0,ys
tbpt,defi,1,1000+ys
!
!define implicit creep for material 1
!
tb,creep,1,,4,1
tbdata,1,1.0e-2,0.5,0.5,0.0

/solu
KBC,0
nlgeom,on
cnvtol,F,1.0,1.0e-10

```

```

rate,on
outres,all,all
time,5
nsub,100,1000,10
solv

```

For information about the EDP option, see:

- The EDP argument and associated specifications in the **TB** command documentation.
- [Extended Drucker-Prager](#) in the *Material Reference*.
- [Extended Drucker-Prager Creep Model](#) in the *Mechanical APDL Theory Reference*.

For information about the MISO and other material-hardening options, see [Multilinear Isotropic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations](#) and [Implicit Creep Procedure](#) (p. 211).

### 8.4.2.49. CAP and CREEP and PLAS (MISO) Example

This input listing illustrates an example of modeling geomaterial cap with implicit creep and multilinear hardening.

```

TB,EDP,1,,11,CYFUN
tbdata, 1, 1.0
tbdata, 2, 1.0
tbdata, 3, -80
tbdata, 4, 10
tbdata, 5, 0.001
tbdata, 6, 2
tbdata, 7, 0.05
tbdata, 8, 1.0
tbdata, 9, 0.6
tbdata, 10, 3.0/1000
tbdata, 11, 0.0
tb,plastic,1,,2,miso
tbpt,defi,0.0,8.0
tbpt,defi,1.0,100.0
tb,creep,1,,4,1
tbeo,capc,shea
tbdata,1,1.0e-4,0.6,0.4,0.0
tb,creep,1,,4,1
tbeo,capc,comp
tbdata,1,2.0e-2,0.5,0.5,0.0nlgeom,on
cnvtol,F,1.0,1.0e-10
rate,on
outres,all,all
time,5
nsub,100,1000,10
solv

```

For information about the cap model, see:

- The **TB,EDP** command's cap model argument (*TBOPT*) and associated specifications.
- [EDP Cap Material Constants](#) in the *Material Reference*.
- [Cap Creep Model](#) in the *Mechanical APDL Theory Reference*.

For information about the MISO and other material-hardening options, see [Multilinear Isotropic Hardening](#) in the *Material Reference*.

For information about the CREEP option, see [Implicit Creep Equations](#) in the *Material Reference*, and [Implicit Creep Procedure](#) (p. 211).

### 8.4.2.50. CHAB and CREEP Example

This input listing illustrates an example of modeling Chaboche nonlinear kinematic hardening with implicit creep.

```

YOUNGS= 30e3
NU      =0.3
SIGMA0= 18.0

MP,EX,1,YOUNGS
MP,PRXY,1,NU

TB,CHAB,1,1,3
TBDATA,1,SIGMA0,
TBDATA,2,5174000,4607500,17155,1040,895.18,9

TB,CREEP,1,1,,1
TBDATA,1, 2.0E-10,0.01,0.1,0

```

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations](#) and [Implicit Creep Procedure](#) (p. 211).

### 8.4.2.51. CHAB and CREEP and NLISO Example

This input listing illustrates an example of modeling Chaboche nonlinear kinematic hardening with implicit creep and nonlinear power law isotropic hardening.

```

YOUNGS= 30e3
NU      =0.3
SIGMA0= 18.0

MP,EX,1,YOUNGS
MP,PRXY,1,NU

TB,CHAB,1,1,3
TBDATA,1,SIGMA0,
TBDATA,2,5174000,4607500,17155,1040,895.18,9

TB,CREEP,1,1,,1
TBDATA,1, 2.0E-10,0.01,0.1,0

TB,NLISO,1,, ,POWER
TBDATA,1,SIGMA0
TBDATA,2,0.5

```

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations](#) and [Implicit Creep Procedure](#) (p. 211).

For information about the NLISO option, see [Nonlinear Isotropic Hardening](#).

### 8.4.2.52. CHAB and CREEP and HILL Example

This input listing illustrates an example of modeling anisotropic plasticity with Chaboche nonlinear kinematic hardening and implicit creep.



```

YOUNGS= 30e3
NU      =0.3
SIGMA0= 18.0

MP,EX,1,YOUNGS
MP,PRXY,1,NU

TB,CHAB,1,1,3
TBDATA,1,SIGMA0,
TBDATA,2,5174000,4607500,17155,1040,895.18,9

TB,CREEP,1,1, ,1
TBDATA,1, 2.0E-10,0.01,0.1,0

RXX=1.1
RYY=1.1
RZZ=1.0
RXY=1.0
RYZ=1.0
RZX=1.0

TB,HILL,1,1
TBDATA,1,RXX,RYY,RZZ,RXY,RYZ,RZX

```

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

For information about the HILL option, see [Hill Yield Criterion](#).

### **8.4.2.53. CHAB and CREEP and HILL and MISO Example**

This input listing illustrates an example of modeling anisotropic plasticity with Chaboche nonlinear kinematic hardening, multilinear isotropic hardening and implicit creep.

```

YOUNGS= 30e3
NU      =0.3
SIGMA0= 18.0

MP,EX,1,YOUNGS
MP,PRXY,1,NU

TB,CHAB,1,1,3
TBDATA,1,SIGMA0,
TBDATA,2,5174000,4607500,17155,1040,895.18,9

TB,CREEP,1,1, ,1
TBDATA,1, 2.0E-10,0.01,0.1,0

RXX=1.1
RYY=1.1
RZZ=1.0
RXY=1.0
RYZ=1.0
RZX=1.0

TB,HILL,1,1
TBDATA,1,RXX,RYY,RZZ,RXY,RYZ,RZX

```

For information about the CHAB option, see [Nonlinear Kinematic Hardening](#).

For information about the CREEP option, see [Implicit Creep Equations and Implicit Creep Procedure \(p. 211\)](#).

For information about the HILL option, see [Hill Yield Criterion](#).

For information about the MISO option, see [Multilinear Isotropic Hardening](#).

## 8.5. Running a Nonlinear Analysis

The program uses an automatic solution-control method that, based on the physics of your problem, sets various nonlinear analysis controls to the appropriate values. If you are not satisfied with the results obtained with these values, you can manually override the settings. The following commands are set to optimal defaults:

<b>ARCLEN</b>	<b>EQSLV</b>	<b>NROPT</b>
<b>AUTOTS</b>	<b>KBC</b>	<b>NSUBST</b>
<b>CDWRITE</b>	<b>LNSRCH</b>	<b>OPNCONTROL</b>
<b>CNVTOL</b>	<b>LSWRITE</b>	<b>PRED</b>
<b>CUTCONTROL</b>	<b>MONITOR</b>	<b>TINTP</b>
<b>DELTIM</b>	<b>NEQIT</b>	

These commands and the settings they control are discussed in later sections. You can also refer to the individual command descriptions in the [Command Reference](#).

If you choose to override the program-specified settings, or if you wish to use an input list from a previous release, issue **SOLCONTROL,OFF** in the **/SOLU** phase. See the **SOLCONTROL** command description for more details.

Automatic solution control is active for the following analyses:

- Single-field nonlinear or transient structural and solid mechanics analysis where the solution DOFs are combinations of UX, UY, UZ, ROTX, ROTY, and ROTZ.
- Single-field nonlinear or transient thermal analysis where the solution DOF is TEMP.

The Solution Controls dialog box cannot be used to set solution controls for a thermal analysis. Instead, use the standard set of solution commands and the standard corresponding menu paths.

## 8.6. Performing a Nonlinear Static Analysis

The procedure for performing a nonlinear static analysis consists of these tasks:

- 8.6.1. Build the Model
- 8.6.2. Set Solution Controls
- 8.6.3. Set Additional Solution Options
- 8.6.4. Apply the Loads
- 8.6.5. Solve the Analysis
- 8.6.6. Review the Results
- 8.6.7. Terminating a Running Job; Restarting

### 8.6.1. Build the Model

This step is essentially the same for both linear and nonlinear analyses, although a nonlinear analysis might include special elements or nonlinear material properties. See [Using Nonlinear \(Changing-Status\) Elements \(p. 257\)](#), and [Modeling Material Nonlinearities \(p. 201\)](#), for more details. If your analysis includes large-strain effects, your stress-strain data must be expressed in terms of *true* stress and *true* (or *logarithmic*) strain. For more information about building models, see the [Modeling and Meshing Guide](#).

After you have created a model, set solution controls (analysis type, analysis options, load step options, and so on), apply loads, and solve. A nonlinear solution differs from a linear solution in that it often requires multiple load increments, and always requires equilibrium iterations. The general procedure for performing these tasks follows. See [Example Nonlinear Analysis \(GUI Method\) \(p. 273\)](#) for an example problem that walks you through a specific nonlinear analysis.

## 8.6.2. Set Solution Controls

Setting solution controls for a nonlinear analysis involves the same options and method of access (the Solution Controls dialog box) as those used for a linear structural static analysis. For a nonlinear analysis, the default settings in the Solution Controls dialog box are essentially the same settings employed by the automatic solution control method described in [Running a Nonlinear Analysis \(p. 242\)](#). See the following sections in [Structural Static Analysis \(p. 9\)](#), with exceptions noted:

- [Set Solution Controls \(p. 10\)](#)
- [Access the Solution Controls Dialog Box \(p. 11\)](#)
- [Using the Basic Tab \(p. 11\)](#)
- [The Transient Tab \(p. 12\)](#)
- [Using the Sol'n Options Tab \(p. 12\)](#)
- [Using the Nonlinear Tab \(p. 12\)](#)
- [Using the Advanced NL Tab \(p. 13\)](#)

### 8.6.2.1. Using the Basic Tab: Special Considerations

Special considerations for setting these options in a nonlinear structural static analysis include:

- When setting **ANTYPE** and **NLGEOM**, choose **Large Displacement Static** if you are performing a new analysis. (Not all nonlinear analyses produce large deformations, however. See [Using Geometric Nonlinearities \(p. 200\)](#) for a further discussion of large deformations.) Choose **Restart Current Analysis** if you want to restart a failed nonlinear analysis. You cannot change this setting after the first load step (that is, after you issue your first **SOLVE** command). Typically, you perform a new analysis, rather than a restart. (Restarts are discussed in the [Basic Analysis Guide](#).)
- When working with time settings, remember that these options can be changed at any load step. See [Loading](#) in the [Basic Analysis Guide](#) for more information about these options. Advanced time/frequency options, in addition to those available on the Solution Controls dialog box, are discussed in [Advanced Load Step Options You Can Set on the Solution Controls Dialog Box \(p. 244\)](#).

A nonlinear analysis requires multiple substeps (or time steps; the two terms are equivalent) within each load step so that the program can apply the specified loads gradually and obtain an accurate solution. The **NSUBST** and **DELTIM** commands both achieve the same effect (establishing a load step's starting, minimum, and maximum step size), but by reciprocal means. **NSUBST** defines the number of substeps to be taken within a load step, whereas **DELTIM** defines the time step size explicitly. If automatic time stepping is off (**AUTOTS**), then the starting substep size is used throughout the load step.

- **OUTRES** controls the data on the results file (Jobname .RST). By default, only the last substep is written to the results file in a nonlinear analysis.

### 8.6.2.2. Advanced Analysis Options You Can Set on the Solution Controls Dialog Box

The following sections provide more detail about some of the advanced analysis options that you can set on the Solution Controls dialog box.

#### 8.6.2.2.1. Equation Solver

Automatic solution control activates the sparse direct solver (**EQSLV,SPARSE**) for most cases. Other options include the PCG and ICCG solvers. For applications using solid elements, the PCG solver may be faster, especially for 3-D modeling.

If using the PCG solver, you may be able to reduce memory usage via the **MSAVE** command. The **MSAVE** command triggers an element-by-element approach for the parts of the model that use **SOLID185**, **SOLID186**, **SOLID187**, **SOLID272**, **SOLID273**, and/or **SOLID285** elements with linear material properties. (**MSAVE** does not support the *layered* option of the **SOLID185** and **SOLID186** elements.) To use **MSAVE**, you must be performing a static or a modal analysis with PCG Lanczos enabled. When using **SOLID185**, **SOLID186**, and/or **SOLID187**, only small strain (**NLGEOM,OFF**) analyses are allowed. Other parts of the model that do not meet the above criteria are solved using global assembly for the stiffness matrix. **MSAVE,ON** can result in a memory savings of up to 70 percent for the part of the model that meets the criteria, although the solution time may increase depending on the capabilities of your computer and the element options selected.

The sparse direct solver, in sharp contrast to the iterative solvers included in the program, is a robust solver. Although the PCG solver can solve indefinite matrix equations, when the PCG solver encounters an ill-conditioned matrix, the solver iterates to the specified number of iterations and stop if it fails to converge. When this happens, it triggers bisection. After completing the bisection, the solver continues the solution if the resulting matrix is well-conditioned. Eventually, the entire nonlinear load step can be solved.

Use the following guidelines for selecting either the sparse or the PCG solver for nonlinear structural analysis:

- If it is a beam/shell or beam/shell and solid structure, choose the sparse direct solver.
- If it is a 3-D solid structure and the number of DOF is relatively large (that is, 200,000 or more DOF), choose the PCG solver.
- If the problem is ill-conditioned (triggered by poor element shapes), or has a big difference in material properties in different regions of the model, or has insufficient displacement boundary constraints, choose the sparse direct solver.

### 8.6.2.3. Advanced Load Step Options You Can Set on the Solution Controls Dialog Box

The following sections provide more detail about some of the advanced load step options that you can set on the Solution Controls dialog box.

#### 8.6.2.3.1. Automatic Time Stepping

Automatic solution control turns automatic time stepping on (**AUTOTS,ON**). An internal auto-time step scheme ensures that the time step variation is neither too aggressive (resulting in many bisection/cut-backs) nor too conservative (time step size is too small). At the end of a time step, the size of the next time step is predicted based on four factors:

- Number of equilibrium iterations used in the last time step (more iterations cause the time step size to be reduced)
- Predictions for nonlinear element status change (time step sizes are decreased when a status change is imminent)
- Size of the plastic strain increment
- Size of the creep strain increment

### 8.6.2.3.2. Convergence Criteria

The program continues to perform equilibrium iterations until the *convergence criteria* (**CNVTOL**) are satisfied (or until the maximum number of equilibrium equations is reached (**NEQIT**)). You can define custom criteria if the default settings are not suitable.

Automatic solution control uses L2-norm of force (and moment) tolerance (*TOLER*) equal to 0.5%, a setting that is appropriate for most cases. In most cases, an L2-norm check on displacement with *TOLER* equal to 5% is also used in addition to the force norm check. The check that the displacements are loosely set serves as a double-check on convergence.

By default, the program checks for force (and, when rotational degrees of freedom are active, moment) convergence by comparing the square root sum of the squares (SRSS) of the force imbalances against the product of *VALUE*\**TOLER*. The default value of *VALUE* is the SRSS of the applied loads (or, for applied displacements, of the Newton-Raphson restoring forces), or *MINREF* (which defaults to 0.01), whichever is greater. The default value of *TOLER* is 0.005. If **SOLCONTROL,OFF**, *TOLER* defaults to 0.001 and *MINREF* defaults to 1.0 for force convergence.

You should almost always use force convergence checking. You can also add displacement (and, when applicable, rotation) convergence checking. For displacements, the program bases convergence checking on the change in deflections ( $\Delta u$ ) between the current (i) and the previous (i-1) iterations:

$$\Delta u = u_i - u_{i-1}$$

If you explicitly define any custom convergence criteria (**CNVTOL**), the entire default criteria is overwritten. Thus, if you define displacement convergence checking, you need to redefine force convergence checking. (Use multiple **CNVTOL** commands to define multiple convergence criteria.)

Using tighter convergence criteria improves the accuracy of your results, but at the cost of more equilibrium iterations. If you want to tighten (or loosen, which is not recommended) your criteria, you should change *TOLER* by one or two orders of magnitude. In general, you should continue to use the default value of *VALUE*; that is, change the convergence criteria by adjusting *TOLER*, not *VALUE*. You should make certain that the default value of *MINREF* = 0.001 makes sense in the context of your analysis. If your analysis uses certain sets of units or has very low load levels, you might want to specify a smaller value for *MINREF*.

Also, we do not recommend putting two or more disjointed structures into one model for a nonlinear analysis because the convergence check tries to relate these disjointed structures, often producing some unwanted residual force.

#### *Checking Convergence in a Single and Multi-DOF System*

To check convergence in a single degree of freedom (DOF) system, you compute the force (and moment) imbalance for the one DOF, and compare this value against the established convergence criteria (*VALUE*\**TOLER*). (You can also perform a similar check for displacement (and rotation) convergence

for your single DOF.) However, in a multi-DOF system, you might want to use a different method of comparison.

The program provides three different vector norms to use for convergence checking:

- The *infinite norm* repeats the single-DOF check at each DOF in your model.
- The *L1 norm* compares the convergence criterion against the sum of the absolute values of force (and moment) imbalance for all DOFs.
- The *L2 norm* performs the convergence check using the square root sum of the squares of the force (and moment) imbalances for all DOFs. (Of course, additional L1 or L2 checking can be performed for a displacement convergence check.)

#### *Example*

For the following example, the substep is considered to be converged if the out-of-balance force (checked at each DOF separately) is less than or equal to  $5000 \times 0.0005$  (that is, 2.5), *and* if the change in displacements (checked as the square root sum of the squares) is less than or equal to  $10 \times 0.001$  (that is, 0.01).

```
CNVTOL,F,5000,0.0005,0  
CNVTOL,U,10,0.001,2
```

### 8.6.2.3.3. Maximum Number of Equilibrium Iterations

Automatic solution control sets the value of **NEQIT** to between 15 and 26 iterations, depending upon the physics of the problem. The idea is to employ a small time step with fewer quadratically converging iterations.

This option limits the maximum number of equilibrium iterations to be performed at each substep (default = 25 if solution control is off). If the convergence criteria have not been satisfied within this number of equilibrium iterations, and if auto time stepping is on (**AUTOTS**), the program attempts to bisect. If bisection is not possible, then the analysis either terminates or moves on to the next load step (according to **NCNV** command settings).

### 8.6.2.3.4. Predictor-Corrector Option

Automatic solution control activates a predictor (**PRED,ON**) if no **SOLID65** elements are present. If the time step size is reduced greatly in the current substep, **PRED** is deactivated. The predictor is also deactivated for transient analyses.

You can activate a predictor on the DOF solution for the first equilibrium iteration of each substep. This feature accelerates convergence and is particularly useful if nonlinear response is relatively smooth, as in the case of ramped loads.

### 8.6.2.3.5. VT Accelerator

This option selects an advanced predictor-corrector algorithm based on Variational Technology to reduce the overall number of iterations (**STAOPT,VT** for static analyses, **TRNOPT,VT** for transient). It is applicable to analyses that include large deflection (**NLGEOM**), hyperelasticity, viscoelasticity, and creep nonlinearities. Rate-independent plasticity and nonlinear contact analyses may not show any improvement in convergence rates; however, you may choose this option with these nonlinearities if you wish to **rerun** the analysis with changes to the input parameters later.

### 8.6.2.3.6. Line Search Option

Automatic solution control toggles line search on and off as needed. For most contact problems, **LNSRCH** is toggled on. For most non-contact problems, **LNSRCH** is toggled off.

This convergence-enhancement tool multiplies the calculated displacement increment by a program-calculated scale factor (having a value between 0 and 1), whenever a stiffening response is detected. Because the line search algorithm is intended to be an *alternative* to the adaptive descent option (**NROPT**), adaptive descent is *not* automatically activated if the line search option is on. We do not recommend activating both line search and adaptive descent simultaneously.

When an imposed displacement exists, a run cannot converge until at least one of the iterations has a line search value of 1. The program scales the entire  $\Delta U$  vector, including the imposed displacement value; otherwise, a "small" displacement occurs everywhere except at the imposed degree of freedom. Until one of the iterations has a line search value of 1, the program does not impose the full value of the displacement.

### 8.6.2.3.7. Cutback Criteria

For finer control over bisections and cutback in time step size, use (**CUTCONTROL**, *Lab*, *VALUE*, *Option*). By default, for *Lab* = PLSLIMIT (maximum plastic strain increment limit), *VALUE* is set to 15%. This field is set to such a large value for avoiding unnecessary bisections caused by high plastic strain due to a local singularity which is not normally of interest to the user. For explicit creep (*Option* = 0), *Lab* = CRPLIM (creep increment limit) and *VALUE* is set to 10%. This is a reasonable limit for creep analysis. For implicit creep (*Option* = 1), there is no maximum creep criteria by default. You can however, specify any creep ratio control. The number of points per cycle for second order dynamic equations (*Lab* = NPOINT) is set to *VALUE* = 13 by default to gain efficiency at little cost to accuracy.

## 8.6.3. Set Additional Solution Options

This section discusses additional options that you can set for the solution. These options do not appear on the Solution Controls dialog box because they are used infrequently, and their default settings rarely need to be changed. Menu paths are provided in this section to help you access these options for those cases in which you choose to override the program-assigned defaults.

### 8.6.3.1. Advanced Analysis Options You Cannot Set via the Solution Controls Dialog Box

The following sections describe some advanced analysis options that you can set for your analysis. You cannot use the Solution Controls dialog box to set these options. Instead, set them using the standard set of solution commands and the standard corresponding menu paths.

#### 8.6.3.1.1. Stress Stiffness

#### 8.6.3.1.2. Newton-Raphson Option

#### 8.6.3.1.1. Stress Stiffness

To account for buckling, bifurcation behavior, the program includes stress stiffness in all geometrically nonlinear analyses (**NLGEOM,ON**).

#### 8.6.3.1.2. Newton-Raphson Option

Automatic solution control uses the FULL Newton-Raphson option with adaptive descent off if there is a nonlinearity present. However, when node-to-node, node-to-surface contact elements are used for



contact analysis with friction, then adaptive descent is automatically turned on. The underlying contact elements require adaptive descent for convergence.

**Command(s):** **NROPT**

**GUI: Main Menu > Solution > Unabridged Menu > Analysis Type > Analysis Options**

Valid only in a nonlinear analysis, this option specifies how often the tangent matrix is updated during solution. The default behavior (**NROPT,AUTO**) allows the program to decide based on the kinds of nonlinearities present in your model, and adaptive descent is activated automatically when appropriate. The following additional options are available:

- **Full** (**NROPT,FULL**):

The program uses the full Newton-Raphson procedure, in which the stiffness matrix is updated at every equilibrium iteration.

If adaptive descent is on (optional), the program uses the tangent stiffness matrix only as long as the iterations remain stable (that is, as long as the residual decreases, and no negative main diagonal pivot occurs). If divergent trends are detected on an iteration, the program discards the divergent iteration and restarts the solution, using a weighted combination of the secant and tangent stiffness matrices. When the iterations return to a convergent pattern, the program resumes using the tangent stiffness matrix. Activating adaptive descent usually enhances the program's ability to obtain converged solutions for complicated nonlinear problems but is supported only for elements indicated under "Special Features" in the Input Summary table (Table 4.n.1 for an element, where *n* is the element number) in the *Element Reference*.

- **Modified** (**NROPT,MODI**):

The program uses the modified Newton-Raphson technique, in which the tangent stiffness matrix is updated at each substep. The matrix is not changed during equilibrium iterations at a substep. This option is not applicable to large-deformation analyses. Adaptive descent is not available.

- **Initial Stiffness** (**NROPT,INIT**):

The program uses the initial stiffness matrix in every equilibrium iteration. This option can be less likely to diverge than the full option, but it often requires more iterations to achieve convergence. It is not applicable to large-deformation analyses. Adaptive descent is not available.

- **Full with unsymmetric matrix** (**NROPT,UNSYM**):

The program uses the full Newton-Raphson procedure, in which the stiffness matrix is updated at every equilibrium iteration. In addition, it generates and uses unsymmetric matrices that you can use for any of the following:

- If you are running a pressure-driven collapse analysis, an unsymmetric pressure load stiffness might be helpful in obtaining convergence. You can include pressure load stiffness using **SOLCONTROL,INCP**.
- If you are defining an unsymmetric material model using **TB,USER**, you would need **NROPT,UNSYM** to fully use the property you defined.
- If you are running a contact analysis, an unsymmetric contact stiffness matrix would fully couple the sliding and the normal stiffnesses. See [Determining Contact Stiffness and Allowable Penetration](#) in the *Contact Technology Guide* for details.



You should first try **NROPT,FULL**, then try **NROPT,UNSYM** if you experience convergence difficulties. (Using an unsymmetric solver requires more computational time to obtain a solution than if you use a symmetric solver.)

If a multistatus element is in the model, it is updated at the iteration in which it changes status, irrespective of the Newton-Raphson option.

### 8.6.3.2. Advanced Load Step Options

The following sections describe some advanced load step options that you can set for your analysis. You cannot use the Solution Controls dialog box to set the options described below. Instead, set them using the standard set of solution commands and the standard corresponding menu paths.

#### 8.6.3.2.1. Creep Criteria

##### 8.6.3.2.2. Time Step Open Control

##### 8.6.3.2.3. Solution Monitoring

##### 8.6.3.2.4. Birth and Death

##### 8.6.3.2.5. Output Control

#### 8.6.3.2.1. Creep Criteria

If your structure exhibits creep behavior, you can specify a creep criterion for automatic time step adjustment (**CRPLIM**, *CRCR*, *Option*). (If automatic time stepping (**AUTOTS**) is off, this creep criterion has no effect.) The program computes the ratio of creep strain increment ( $\Delta\varepsilon_{cr}$ , the change in creep strain in the last time step) to the elastic strain ( $\varepsilon_{el}$ ), for all elements. If the maximum ratio is greater than the criterion *CRCR*, the program then decreases the next time step size; if it is less, the program might increase the next time step size. (The program also bases automatic time stepping on the number of equilibrium iterations, impending element status change, and plastic strain increment. The time step size is adjusted to the minimum size calculated for any of these items.) For explicit creep (*Option* = 0), if the ratio  $\Delta\varepsilon_{cr} / \varepsilon_{el}$  is above the stability limit of 0.25, and if the time increment cannot be decreased, a divergent solution is possible and the analysis terminates with an error message. This problem is avoidable by making the minimum time step size sufficiently small (**DELTIM** and **NSUBST**). For implicit creep (*Option* = 1), there is no maximum creep limit by default. You can however, specify any creep ratio control.

**Command(s):** **CRPLIM**

**GUI:** **Main Menu > Solution > Unabridged Menu > Load Step Opts > Nonlinear > Creep Criterion**

If you do not want to include the effects of creep in your analysis, use the **RATE** command with *Option* = OFF, or set the time steps to be longer than the previous time step, but not more than 1.0e-6 longer.

#### 8.6.3.2.2. Time Step Open Control

This option is available for thermal analysis. (Remember that you cannot perform a thermal analysis using the Solution Controls dialog box; you must use the standard set of solution commands or the standard corresponding menu paths instead.) This option's primary use is in unsteady state thermal analysis where the final temperature stage reaches a steady state. In such cases, the time step can be opened quickly. The default is that if the TEMP increment is smaller than 0.1 in three (NUMSTEP = 3) contiguous substeps, the time step size can be "opened-up" (value = 0.1 by default). The time step size can then be opened continuously for greater solution efficiency.

**Command(s):** **OPNCONTROL**

**GUI:** **Main Menu > Solution > Unabridged Menu > Load Step Opts > Nonlinear > Open Control**

### 8.6.3.2.3. Solution Monitoring

This option provides a facility to monitor a solution value at a specified node in a specified DOF. The command also provides a means to quickly review the solution convergence efficiency, rather than attempting to gather this information from a lengthy output file. For instance, if an excessive number of attempts were made for a substep, the information contained in the file provides hints to either reduce the initial time step size or increase the minimum number of substeps allowed through the **NSUBST** command to avoid an excessive number of bisections.

**Command(s): MONITOR**

**GUI: Main Menu> Solution> Unabridged Menu> Load Step Opts> Nonlinear> Monitor**

Additionally, the **NLHIST** command allows you to monitor results of interest in real time during solution. Before starting the solution, you can request nodal data such as displacements or reaction forces at specific nodes. You can also request element nodal data such as stresses and strains at specific elements to be graphed. Pair-based contact data are also available. The result data are written to a file named `Jobname.nlh`.

For example, a reaction force-deflection curve could indicate when possible buckling behavior occurs. Nodal results and contact results are monitored at every converged substep while element nodal data are written as specified via the **OUTRES** setting.

You can also track results during batch runs. To execute, either access the ANSYS Launcher and select **File Tracking** from the **Tools** menu, or type `nlhist150` in the command line. Use the supplied file browser to navigate to your `Jobname.nlh` file, and select it to invoke the tracking utility. You can use this utility to read the file at any time, even after the solution is complete.

**Command(s): NLHIST**

**GUI: Main Menu> Solution> Results Tracking**

### 8.6.3.2.4. Birth and Death

Specify birth and death options as necessary. You can deactivate (**EKILL**) and reactivate (**EALIVE**) selected elements to model the removal or addition of material in your structure. As an alternative to the standard birth and death method, you can change the material properties for selected elements (**MPCHG**) between load steps.

**Command(s): EKILL, EALIVE**

**GUI: Main Menu> Solution> Load Step Opts> Other> Birth & Death> Kill Elements**

**Main Menu> Solution> Load Step Opts> Other> Birth & Death> Activate Elem**

The program "deactivates" an element by multiplying its stiffness by a very small number (which is set by the **ESTIF** command), and by removing its mass from the overall mass matrix. Element loads (pressure, heat flux, thermal strains, and so on) for inactive elements are also set to zero. You need to define all possible elements during preprocessing; you cannot create new elements in SOLUTION.

Those elements to be "born" in later stages of your analysis should be *deactivated* before the first load step, and then *reactivated* at the beginning of the appropriate load step. When elements are reactivated, they have a zero strain state, and (if **NLGEOM,ON**) their geometric configuration (length, area, and so on) is updated to match the current displaced positions of their nodes. See the [Advanced Analysis Guide](#) for more information about birth and death.

Another way to affect element behavior during solution is to change the material property reference number for selected elements:

**Command(s): MPCHG**

**GUI: Main Menu> Solution> Load Step Opts> Other> Change Mat Props> Change Mat Num**

Use **MPCHG** with caution. Changing material properties in a nonlinear analysis may produce unintended results, particularly if you change nonlinear (**TB**) material properties.

### 8.6.3.2.5. Output Control

In addition to **OUTRES**, which you can set on the Solution Controls dialog box, there are several other output control options that you can set for an analysis:

**Command(s):** **OUTPR, ERESX**

**GUI:** **Main Menu> Solution> Unabridged Menu> Load Step Opts> Output Ctrl> Solu Printout  
Main Menu> Solution> Unabridged Menu> Load Step Opts> Output Ctrl> Integration Pt**

*Printed output* (**OUTPR**) includes any results data on the output file (`Jobname. OUT`).

*Extrapolation of results* (**ERESX**) copies an element's integration point stress and elastic strain results to the nodes instead of extrapolating them, if nonlinear strains (plasticity, creep, swelling) are present in the element. The integration point nonlinear strains are always copied to the nodes.

See [Loading](#) in the *Basic Analysis Guide* for more information about these options.

### 8.6.4. Apply the Loads

Apply loads on the model. See [Structural Static Analysis \(p. 9\)](#) in this guide and [Loading](#) in the *Basic Analysis Guide* for load information. Inertia and point loads maintain constant direction, but surface loads "follow" the structure in a large-deformation analysis. You can apply complex boundary conditions by defining a one-dimensional table (TABLE type array parameter). See [Applying Loads Using TABLE Type Array Parameters \(p. 17\)](#) in this guide for more information.

### 8.6.5. Solve the Analysis

You solve a nonlinear analysis using the same commands and procedure as you do in solving a linear static analysis. See [Solve the Analysis \(p. 18\)](#) in [Structural Static Analysis \(p. 9\)](#). If you need to define multiple load steps, you must respecify time settings, load step options, and so on, and then save and solve for each of the additional load steps. Other methods for multiple load steps - the load step file method and the array parameter method - are described in the *Basic Analysis Guide*.

### 8.6.6. Review the Results

Results from a nonlinear static analysis consist mainly of displacements, stresses, strains, and reaction forces. You can review these results in POST1, the general postprocessor, or in POST26, the time-history postprocessor.

Remember that in POST1, only one substep can be read in at a time, and that the results from that substep should have been written to `Jobname. RST`. (The load step option command **OUTRES** controls which substep results are stored on `Jobname. RST`.) A typical POST1 postprocessing sequence is described below.

#### 8.6.6.1. Points to Remember

- To review results in POST1, the database must contain the same model for which the solution was calculated.
- The results file (`Jobname. RST`) must be available.

### 8.6.6.2. Reviewing Results in POST1

1. Verify from your output file (`Jobname.OUT`) whether or not the analysis converged at all load steps.
  - If not, you probably do not want to postprocess the results, other than to determine why convergence failed.
  - If your solution converged, then continue postprocessing.

2. Enter POST1. If your model is not currently in the database, issue **RESUME**.

**Command(s):** `/POST1`

**GUI:** Main Menu > General Postproc

3. Read in results for the desired load step and substep, which can be identified by load step and substep numbers or by time.

**Command(s):** `SET`

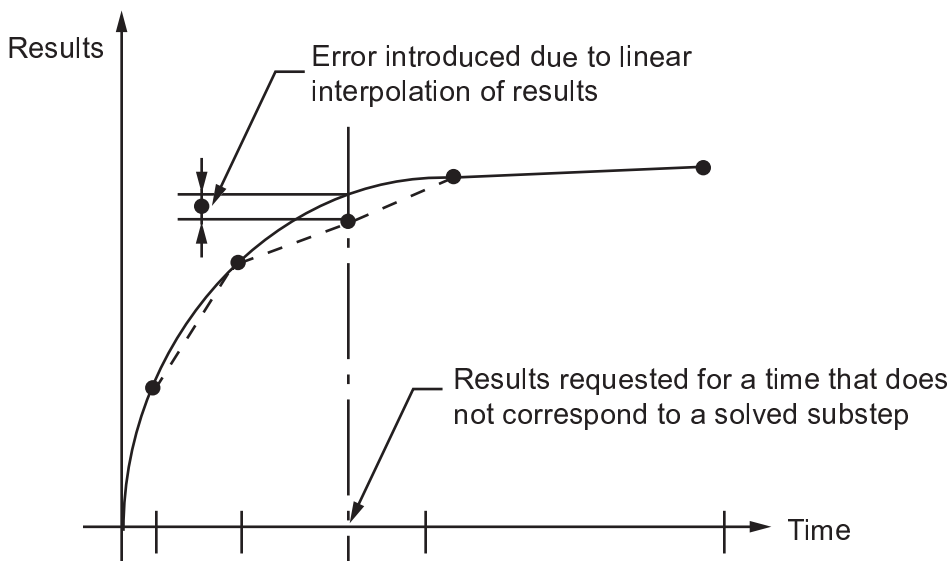
**GUI:** Main Menu > General Postproc > Read Results > *load step*

You can also use the **SUBSET** or **APPEND** commands to read in or merge results data for selected portions of the model only. The *LIST* argument on any of these commands lists the available solutions on the results file. You can also limit the amount of data written from the results file to the database through the **INRES** command. Additionally, you can use the **ETABLE** command to store result items for selected elements. See the individual command descriptions in the [Command Reference](#) for more information.

#### Caution

If you specify a TIME value for which no results are available, the program performs a linear interpolation to calculate the results at that value of TIME. Realize that this interpolation usually causes some loss of accuracy in a nonlinear analysis (see [Figure 8.14: Linear Interpolation of Nonlinear Results Can Introduce Some Error](#) (p. 252)). Therefore, for a nonlinear analysis, you should usually postprocess at a TIME that corresponds exactly to the desired substep.

**Figure 8.14: Linear Interpolation of Nonlinear Results Can Introduce Some Error**



4. Display the results using any of the following options:

**Option: Display Deformed Shape**

**Command(s):** **PLDISP**

**GUI: Main Menu> General Postproc> Plot Results> Deformed Shape**

In a large-deformation analysis, you might prefer to use a true scale display (**/DSCALE,,1**).

**Option: Contour Displays**

**Command(s):** **PLNSOL** or **PLESOL**

**GUI: Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu** or **Element Solu**

Use these options to display contours of stresses, strains, or any other applicable item. If you have adjacent elements with different material behavior (such as can occur with plastic or multilinear elastic material properties, with different material types, or with adjacent deactivated and activated elements), you should take care to avoid nodal stress averaging errors in your results. Selecting logic (described in the *Basic Analysis Guide*) provides a means of avoiding such errors.

The *KUND* field on **PLNSOL** and **PLESOL** gives you the option of overlaying the undeformed shape on the display.

You can also contour element table data and line element data:

**Command(s):** **PLETAB, PLLS**

**GUI: Main Menu> General Postproc> Element Table> Plot Element Table**

**Main Menu> General Postproc> Plot Results> Contour Plot> Line Elem Res**

Use **PLETAB** to contour element table data and **PLLS** to contour line element data.

**Option: Tabular Listings**

**Command(s):** **PRNSOL** (nodal results) **PRESOL** (element-by-element results) **PRRSOL** (reaction data) **PRETAB PRITER** (substep summary data), and so on. **NSORT ESORT**

**GUI: Main Menu> General Postproc> List Results> Nodal Solution**

**Main Menu> General Postproc> List Results> Element Solution**

**Main Menu> General Postproc> List Results> Reaction Solution**

Use the **NSORT** and **ESORT** commands to sort the data before listing them.

**Other Capabilities**

Many other postprocessing functions - mapping results onto a path, report quality listings, and so on - are available in POST1. See *The General Postprocessor (POST1)* in the *Basic Analysis Guide* for details. Load case combinations usually are not valid for nonlinear analyses.

### 8.6.6.3. Reviewing Results in POST26

You can also review the load-history response of a nonlinear structure using POST26, the time-history postprocessor. Use POST26 to compare one variable against another. For instance, you might graph the displacement at a node versus the corresponding level of applied load, or you might list the plastic strain at a node and the corresponding *TIME* value. A typical POST26 postprocessing sequence might follow these steps:

1. Verify from your output file (*Jobname.OUT*) whether or not the analysis converged at all desired load steps. You should not base design decisions on unconverged results.
2. If your solution converged, enter POST26. If your model is not currently in the database, issue **RESUME**.

**Command(s):** **/POST26**

**GUI: Main Menu> TimeHist Postpro**

3. Define the variables to be used in your postprocessing session. The **SOLU** command causes various iteration and convergence parameters to be read into the database, where you can incorporate them into your postprocessing.

**Command(s): NSOL, ESOL, RFORCE****GUI: Main Menu> TimeHist Postpro> Define Variables**

4. Graph or list the variables.

**Command(s): PLVAR** (graph variables) **PRVAR EXTREM** (list variables)**GUI: Main Menu> TimeHist Postpro> Graph Variables****Main Menu> TimeHist Postpro> List Variables****Main Menu> TimeHist Postpro> List Extremes**

## Other Capabilities

Many other postprocessing functions are available in POST26. See [The Time-History Postprocessor \(POST26\)](#) in the *Basic Analysis Guide* for details.

### 8.6.7. Terminating a Running Job; Restarting

You can stop a nonlinear analysis by creating an "abort" file (`Jobname.ABT`). See [Solution](#) in the *Basic Analysis Guide* for details. The program also stops upon successful completion of the solution, or if a convergence failure occurs.

You can often restart an analysis if it successfully completed one or more iterations before it terminated. Restart procedures are covered in [Restarting an Analysis](#) in the *Basic Analysis Guide*.

## 8.7. Performing a Nonlinear Transient Analysis

Many of the tasks that you need to perform in a nonlinear transient analysis are the same as (or similar to) those that you perform in nonlinear static analyses (described in [Performing a Nonlinear Static Analysis](#) (p. 242)) and linear full transient dynamic analyses (described in [Structural Static Analysis](#) (p. 9)). However, this section describes some additional considerations for performing a nonlinear transient analysis.

You cannot use the Solution Controls dialog box ([Performing a Nonlinear Static Analysis](#) (p. 242)) to set solution controls for a thermal analysis. Instead, use the standard set of solution commands and the standard corresponding menu paths.

### 8.7.1. Build the Model

This step is the same as for a nonlinear static analysis. If your analysis includes time-integration effects, however, include a value for mass density (**MP,DENS**). If you wish, you can also define material-dependent structural damping (**MP,BETD** or **MP,ALPD**).

### 8.7.2. Apply Loads and Obtain the Solution

1. Specify transient analysis type and define analysis options as you would for a nonlinear static analysis:
  - New Analysis or Restart (**ANTYPE**)
  - Analysis Type: Transient (**ANTYPE**)

- Large Deformation Effects (**NLGEOM**)
  - Large Displacement Transient (if using the Solution Controls dialog box to set analysis type)
2. Apply loads and specify load step options in the same manner as you would for a linear full transient dynamic analysis. A transient load history usually requires multiple load steps, with the first load step typically used to establish initial conditions (see the *Basic Analysis Guide*). The general, nonlinear, birth and death, and output control options available for a nonlinear static analysis are also available for a nonlinear transient analysis.

In a nonlinear transient analysis, time must be greater than zero. See [Transient Dynamic Analysis \(p. 107\)](#) for procedures for defining nonzero initial conditions.

For a nonlinear transient analysis, you must specify whether you want stepped or ramped loads (**KBC**). See the *Basic Analysis Guide* for further discussion about ramped vs. stepped loads.

You can also specify dynamics options: alpha and beta damping, time-integration effects, and transient integration parameters.

**Command(s):** **ALPHAD, BETAD, TIMINT, TINTP**

**GUI:** **Main Menu > Solution > Analysis Type > Sol'n Control ( : Transient Tab)**

**Main Menu > Solution > Unabridged Menu > Load Step Opts > Time/Frequenc > Damping**

**Main Menu > Solution > Unabridged Menu > Load Step Opts > Time/Frequenc > Time Integration**

An explanation of the dynamics options follows.

- **Damping**

Rayleigh damping constants are defined using the constant mass (**ALPHAD**) and stiffness (**BETAD**) matrix multipliers. In a nonlinear analysis the stiffness may change drastically - do not use **BETAD**, except with care. See [Damping \(p. 2\)](#) for details about damping.

- **Time-Integration Effects (TIMINT)**

Time-integration effects are ON by default in a transient analysis. For creep, viscoelasticity, viscoplasticity, or swelling, you should turn the time-integration effects off (that is, use a static analysis). These time-dependent effects are usually not included in dynamic analyses because the transient dynamic time step sizes are often too short for any significant amount of long-term deformation to occur.

Except in kinematic (rigid-body motion) analyses, you rarely need to adjust the transient integration parameters (**TINTP**), which provide numerical damping to the Newmark and HHT methods. (See your *Mechanical APDL Theory Reference* for more information about these parameters.)

Automatic solution control sets the defaults to a new time-integration scheme for use by first order transient equations. This is typically used for unsteady state thermal problems where  $\theta = 1.0$  (set by **SOLCONTROL**, ON); this is the backward Euler scheme. It is unconditionally stable and more robust for highly nonlinear thermal problems such as phase changes. The oscillation limit



tolerance defaults to 0.0, so that the response first order eigenvalues can be used to more precisely determine a new time step value.

---

### Note

If you are using the Solution Controls dialog box to set solution controls, you can access all of these options (**ALPHAD**, **BETAD**, **KBC**, **TIMINT**, **TINTP**, **TRNOPT**) on the **Transient** tab.

---

3. Write load data for each load step to a load step file.  
**Command(s): LSWRITE**  
**GUI: Main Menu> Solution> Load Step Opts> Write LS File**
4. Save a backup copy of the database to a named file.  
**Command(s): SAVE**  
**GUI: Utility Menu> File> Save As**
5. Start solution calculations. Other methods for multiple load steps are described in [Getting Started](#) in the *Basic Analysis Guide*.  
**Command(s): LSSOLVE**  
**GUI: Main Menu> Solution> Solve> From LS Files**
6. After you have solved all load steps, leave SOLUTION.  
**Command(s): FINISH**  
**GUI: Close the Solution menu.**

### 8.7.3. Review the Results

As in a nonlinear static analysis, you can use POST1 to postprocess results at a specific moment in time. Procedures are much the same as described previously for nonlinear static analyses. Again, you should verify that your solution has converged before you attempt to postprocess the results.

Time-history postprocessing using POST26 is essentially the same for nonlinear as for linear transient analyses. See the postprocessing procedures outlined in [Transient Dynamic Analysis](#) (p. 107).

More details of postprocessing procedures can be found in the *Basic Analysis Guide*.

## 8.8. Example Input for a Nonlinear Transient Analysis

Following is an example input listing for a nonlinear transient analysis:

```
! Build the Model:
/PREP7
---
---          ! Similar to a linear full transient model, with
---          ! these possible additions: nonlinear material
---          ! properties, nonlinear elements
---
FINISH
!
! Apply Loads and Obtain the Solution:
/SOLU
ANTYPE,TRANS
! TRNOPT,FULL by default
---
---          ! Establish initial conditions as in linear full
---          ! transient analysis
LSWRITE          ! Initial-condition load step
```



```

NLGEOM,ON          ! Nonlinear geometric effects (large deformations)
! NROPT=AUTO by default: Program chooses appropriate Newton-Raphson and
! Adaptive Descent options, depending on
! nonlinearities encountered

! Loads:
F,...
D,...
! Load Step Options:
TIME,...          ! TIME at end of load step
DELTIM,...        ! Time step controls (starting, min, max)
AUTOTS,ON         ! Automatic time stepping, including bisection
! KBC=0 by default (ramped loading)
! Dynamic Options:
ALPHAD,...        ! Mass damping
TIMINT,ON         ! TIMINT,ON by default, unless you turned it OFF for
! initial-condition load step

! Nonlinear Options:
CNVTOL,...        ! Convergence criteria
! NEQIT=25 by default
NCNV,...         ! Nonconvergence termination controls
PRED,ON          ! Predictor ON
OUTRES,ALL,ALL    ! Results for every substep written to database
LSWRITE          ! First "real" transient load step
---             ! Additional load steps, as needed
---
LSSOLVE,1,3      ! Initiate multiple l.s. solution
SAVE
FINISH
!
! Review the Results:
/POST26          ! Time-History Postprocessor
SOLU,2,CNVG      ! Check convergence
SOLU,3,FOCV
PRVAR,2,3
NSOL,...         ! Store results (displacements, stresses, etc.) as
! variables
PLVAR,...        ! Graph results vs. TIME to evaluate general quality
! of analysis, determine critical time step, etc.

FINISH
!
/POST1          ! General Postprocessor
SET,...         ! Read results from desired time step
PLDISP,...      ! Postprocess as desired
PLNSOL,...
NSORT,...
PRNSOL,...
FINISH

```

See the **ANTYPE**, **TRNOPT**, **LSWRITE**, **NLGEOM**, **NROPT**, **TIME**, **DELTIM**, **AUTOTS**, **KBC**, **ALPHAD**, **TIMINT**, **CNVTOL**, **NEQIT**, **NCNV**, **PRED**, **OUTRES**, **LSSOLVE**, and **SOLU** command descriptions for more information.

## 8.9. Restarts

Restart procedures for a transient analysis are essentially the same as for a static analysis; see [Restarting an Analysis](#) in the *Basic Analysis Guide*.

## 8.10. Using Nonlinear (Changing-Status) Elements

Nonlinear elements display an abrupt change in stiffness when they experience a change in status. For example, when a cable goes slack, its stiffness suddenly drops to zero. When two separate bodies come into contact, their overall stiffness changes drastically. These and other status-dependent stiffness changes can be modeled by using nonlinear elements (described below), by applying birth and death options to applicable elements (see the *Advanced Analysis Guide*), or by changing material properties (**MPCHG**). Some of the nonlinear element features described below are available only in the ANSYS

Multiphysics, ANSYS Mechanical, and ANSYS Structural products only. See the [Element Reference](#) for details.

### 8.10.1. Element Birth and Death

Sometimes, an element's status changes between "existent" and "nonexistent." The birth and death options (**EKILL**, **EALIVE**, **ESTIF**) (**Main Menu**> **Solution**> **Load Step Opts**> **Other**) can be used to deactivate or reactivate selected elements in such cases. The birth and death feature is discussed in detail in [Element Birth and Death](#) in the *Advanced Analysis Guide*.

### 8.11. Unstable Structures

A structure can become unstable when a load reaches its buckling value or when nonlinear material becomes unstable. It is more common in slender structures than in bulky structures. The instability could be global (such as a snap-through of a plate) or local (such as failure of a stiffener). Instability problems usually pose convergence difficulties and therefore require the application of special nonlinear techniques.

You can apply the following techniques to solve instability problems:

- **Nonlinear stabilization**

A tool for dealing with local instabilities as well as global instability. You can use it together with nearly any other nonlinear solution technique, such as line search and automatic time stepping (although not with the arc-length method).

- **Arc-length method**

This method can circumvent global instability when forces are applied. More importantly, it can simulate the negative slope portion of a load-vs.-displacement curve.

- **Running a static problem as a "slow dynamic" analysis**

This method is not strictly a different technique; rather, you use a dynamic effect to prevent divergence. This method is not especially easy to use because the analysis type changes, so you must input mass, apply a damping factor if necessary, and use proper time-integration parameters. ANSYS, Inc. therefore recommends trying nonlinear stabilization or the arc-length method first.

Alternative methods are available to help achieve convergence. For example, you could apply displacements instead of forces, if applicable. Or, you could apply artificial small stiffness to the unstable degrees of freedom. These stiffnesses could be either permanent or activated/removed at some load step by applying the stiffness using the control element [COMBIN37](#). However, such methods may be unreliable, impractical, or simply not applicable.

#### 8.11.1. Using Nonlinear Stabilization

Convergence difficulty due to an unstable problem is usually the result of a large displacement for smaller load increments. Nonlinear stabilization can be understood as adding an artificial damper or *dashpot* element at each node of an element that supports this technique.

To better conceptualize the artificial dashpot element, think of it as having two nodes: one is the node of the FE model that you create, the other is fixed on the ground. The program calculates the damping force such that it is proportional to the relative pseudo velocity of the two nodes of the artificial element, which is equal to the velocity of the node belonging to the FE model.

The pseudo velocity is calculated as a displacement increment divided by the time increment of the substep. Therefore, any DOF that tends to be unstable has a large displacement increment causing a large damping (stabilization) force; this force, in turn, reduces the displacements at the DOF so that stabilization is achieved.

For the DOFs that are stable, the dashpot elements have little effect on the results because the displacements and the stabilization forces are small relative to the physical forces.

The coefficient used to calculate the damping (stabilization) force is the damping factor. Although it has the same physical meaning and unit as physical damping, it is purely numerical in nonlinear stabilization. The program calculates a damping factor based on the energy dissipation ratio that you specify, or you can input the damping factor value directly.

### 8.11.1.1. Input for Stabilization

The only command necessary for using nonlinear stabilization is **STABILIZE**. The command activates or deactivates stabilization from one load step to another, or after a multiframe restart during a load step.

The program assumes that the first substep of a load step is stable and calculates the basic properties of the artificial dashpot elements based on this substep. Therefore, the program does not apply stabilization for the first substep unless you specify that it should do so (via the command's *SubStpOpt* option).

The following topics describe how to use the **STABILIZE** command in a nonlinear analysis:

8.11.1.1.1. Controlling the Stabilization Force

8.11.1.1.2. Applying a Constant or Reduced Stabilization Force

8.11.1.1.3. Using the Options for the First Substep

8.11.1.1.4. Setting the Limit Coefficient for Checking Stabilization Forces

#### 8.11.1.1.1. Controlling the Stabilization Force

Two methods are available for controlling the stabilization force:

- Applying an energy dissipation ratio (**STABILIZE**,,ENERGY,,)
- Applying a damping factor (**STABILIZE**,,DAMPING,,)

#### Energy Dissipation Ratio

The *energy dissipation ratio* is the ratio of work done by stabilization forces to element potential energy.

The energy dissipation ratio should be between 0 and 1. Because the value is used with predicted energies, the program allows an input value greater than 1, but use it with caution. The greater the value of the energy ratio or damping factor, the greater the stabilization force (assuming that the specified number of substeps and time remain unchanged) so that the system has a stiffer response.

The specified value should be large enough to circumvent the divergence, but small enough to avoid excessive stiffness. The ideal value is fully dependent on the specific problem, the time of the load step, and the number of substeps.

You may need a few tries to determine the best value. Generally, use a smaller value for local instability and a larger value for global instability. The smaller value should be used for solid elements and the

larger value should be used for shell, beam, and link elements. Use a smaller value if the specified time for a load step is small and a larger value if the specified time for a load step is large.

With the energy dissipation method, the program calculates the damping factor (based on the input energy dissipation ratio) during the first substep after the command executes. The program uses the calculated damping factor by predicting the element potential energy and stabilization energy at the end of the load step based on the data of the current substep, then setting the energy dissipation ratio equal to or smaller than the specified value. This prediction could be inaccurate when the problem is highly nonlinear. It is a good practice to examine the energies after the solution has completed because the energy dissipation ratio of the solution could be greater than the ratio initially specified via the **STABILIZE** command.

### Damping Factor

The numerical *damping factor* is the value that the program uses to calculate stabilization forces for all subsequent substeps. The damping factor is highly dependent on the element size, shape, material, and other factors including the size of the load step and time used in the load step. The damping factor therefore varies from element to element.

During a run using the energy dissipation method, the program calculates the damping factor and reports an element volume weighted average value in the `.out` file. The value reported provides a reference value for you to specify if you want to apply a damping factor as the stabilization control in a subsequent run. When you input a damping factor as the stabilization control, the program uses that value for *all* applicable elements; therefore, the results can differ from those of a run where you use the energy dissipation method exclusively.

The value used as a damping factor can usually have a much wider range of variance than the value used for the energy dissipation ratio (which can only change from 0 to 1 in most analyses). If it becomes apparent that your analysis is too sensitive to the energy dissipation ratio value, try using the damping factor.

#### 8.11.1.1.2. Applying a Constant or Reduced Stabilization Force

When stabilization is active, the program can apply the stabilization force in two ways: constant (**STABILIZE,CONSTANT**) or reduced (**STABILIZE,REDUCE**). The constant option keeps the *damping factor* (calculated or input) unchanged during each substep of a load step. The reduced option reduces the damping factor linearly to zero at the end of the load step.

Although the constant option works well in most cases, some stabilization forces usually remain at the end of the load step. Unless the stabilization forces are very small, convergence difficulties may occur if stabilization is deactivated in the next load step. It may be difficult to converge for the first substep of the following load step because the stabilization forces suddenly becomes zero. In such a case, use the reduced option for the previous load step.

#### Example

Convergence difficulties when using the constant option can occur in an analysis of creep phenomena, where the load is usually applied quickly in the first load step, but no new load is applied at the second load step (which usually has a very long time span). The stabilization forces could be large at the end of the first load step because the time is short and pseudo velocity is high at the first load step. In this case, if stabilization is needed for the first load step, the reduced option is best. The second load step is usually stable so that stabilization is unnecessary.

### 8.11.1.1.3. Using the Options for the First Substep

When stabilization is active, you can activate artificial dashpot elements (**STABILIZE**) for the first substep of a load step.

In most analyses, stabilization is unnecessary because the structure is initially stable, so the first substep should converge if the substep size is reasonable. When *SubStpOpt* = NO, the program calculates all necessary data for stabilization more accurately and achieves convergence more easily; therefore, ANSYS, Inc. recommends using this option whenever possible.

#### Convergence Problems at the First Substep

There are some situations where convergence is an issue at the first substep. For such cases, you can specify substep option (**STABILIZE**,,,*SubStpOpt*) MINTIME or ANYTIME.

The MINTIME option activates stabilization only when the time increment reaches the minimum time increment and the analysis still has not converged. Use this option for the *first* load step only.

The ANYTIME option activates stabilization for any time increment tried for the first substep. Use this option for any load step *other than the first load step* where constant stabilization is active (**STABILIZE**,CONSTANT).

The program uses the damping factor calculated at the previous load step to calculate the stabilization forces for the first substep. If no such value is available, the program assumes a deformation mode for the first substep and calculate a damping factor for the first substep. In either case, the program recalculates the damping factor after a successful convergence based on the solution of the first substep and uses the new value for all subsequent substeps.

Use caution with either substep option and check the final result to verify that the stabilization forces and energies are not excessive.

#### Example

Specify *SubStpOpt* = ANYTIME for the current load step after you have applied a constant stabilization force (**STABILIZE**,CONSTANT) in the previous load step and the first substep did not converge, yet the current load step also requires stabilization. This option is especially useful if you do not want to rerun the previous load step using the reduced method (**STABILIZE**,REDUCE).

### 8.11.1.1.4. Setting the Limit Coefficient for Checking Stabilization Forces

When the L2-norm of the stabilization force exceeds the product of the L2-norm of the internal force and the stabilization force coefficient, the program issues a message displaying both the stabilization force norm and the internal force norm. The message indicates that the stabilization force may be too large. In such cases, verify the results carefully, and consider adjusting the stabilization force by updating either the energy-dissipation ratio (**STABILIZE**,,ENERGY) or the damping factor (**STABILIZE**,,DAMPING).

If you want to change the stabilization force limit coefficient (by default 0.2, or 20 percent), issue a **STABILIZE**,,,*FORCELIMIT* command. (To omit a stabilization force check, specify a value of 0.)

The program checks the norms (and reports them if necessary) only *after* a substep has converged. The stabilization force check has no effect on convergence.

### 8.11.1.2. Checking Results After Applying Stabilization

Stabilization can help with convergence problems, but it can also affect accuracy if the stabilization energy or forces are too large. Although the program automatically reports the stabilization force norms and compares them to internal force norms, it is still very important to check the stabilization energy and forces to determine whether or not they are excessive.

Stabilization energy, the work done by stabilization forces, should be compared to element potential energy. The energies can be output in the .OUT file (via the **OUTPR** command). You can also access the energies as follows:

- In POST1, via **PREENERGY**, **PRESOL**, **PLESOL**, and **ETABLE** commands.
- In POST26 by **ENERSOL** and **ESOL** commands.

If the stabilization energy (which could be larger than that specified via the **STABILIZE** command) is much less than the potential energy (for example, within a 1.0 percent tolerance), the result should be acceptable and there should be no need to check the stabilization forces further.

When stabilization energy is large, check the stabilization forces at each DOF for all substeps. If the stabilization forces are much smaller than the applied loads and reaction forces (for example, within a 0.5 percent tolerance), the results are still acceptable. Such a case could occur when an elastic system is loaded first, then unloaded significantly. It is possible that the final element potential energy is small and stabilization energy is relatively large, but all stabilization forces are small. Currently, stabilization forces are accessible in the .OUT file (via **OUTPR**).

Even when both stabilization energy and forces are too large, the results could still be valid. Such a scenario is possible when a large part of an elastic structure undergoes large rigid body motion (as in a snap-through simulation). In such a case, the stabilization energy could be large as well as the stabilization force for some DOFs at some substeps, but the results could still be acceptably accurate. Nevertheless, consider the results along with other support data and use your own discretion.

### 8.11.1.3. Tips for Using Stabilization

You can use nonlinear stabilization to achieve convergence in an analysis of unstable nonlinear problems such as **post-buckling**, snap-through simulation, and analyses where material is unstable. Although you can activate nonlinear stabilization at the beginning of the solution, it is more efficient and accurate in most cases to activate stabilization in a multiframe restart.

If you wish to activate stabilization after a restart, do not restart from the last converged substep. Rather, restart from the next-to-last converged substep or at some other substep prior to the last converged substep. (The program needs one substep to prepare the data for stabilization.)

Because it is usually impossible to know when a system will become unstable during loading before an analysis starts, run the nonlinear analysis as usual while saving restart files for at least the last two converged substeps. If the analysis fails to converge because of instability, restart the analysis with stabilization activated from the next-to-last converged substep or at some other substep prior to the last converged substep. (The program needs one substep to prepare the data for stabilization.)

If the behavior of a problem is well known from a previous analysis and the structure loses stability very soon after you begin to apply loads, you can activate stabilization at the beginning of the analysis.



Be aware that when stabilization is active, the results could vary if the number of substeps changes. The behavior occurs because the pseudo velocity is different, which in turn causes different stabilization forces. The more stable the system, the less significant the difference.

If restarting from a different substep, using a damping factor (**STABILIZE,,DAMPING**) can yield more consistent results because the energy prediction may be different from substep to substep, which may necessitate quite different damping factors.

### Deactivating Stabilization

Each time that stabilization is deactivated (**STABILIZE,OFF**), the stabilization forces change suddenly, which may cause convergence problems. Before completely deactivating stabilization in such cases, use the reduced method of stabilization (**STABILIZE,REDUCE**) and specify the damping factor used for the previous load step.

#### Example

Assume that load step 1 is unstable but solvable with stabilization. Load step 2 is stable and requires no stabilization, yet does not converge if you deactivate stabilization (**STABILIZE,OFF**). In this scenario, you can add a pseudo load step (**STABILIZE,REDUCE,DAMPING,VALUE**). The damping factor should be the value from load step 1. Do not apply any new loads. This technique should help with convergence.

## 8.11.2. Using the Arc-Length Method

The arc-length method (**ARCLen** and **ARCTRM**) is another way to solve unstable problems. This method is restricted to static analyses with proportional (ramped) loads only.

When choosing the number of substeps (**NSUBST**), consider that more substeps result in a longer solution time but sometimes help the program to converge. Ideally, you want the minimum number of substeps required to produce an optimally efficient solution. You might have to make an educated guess of the desired number of substeps, and adjust and re-analyze as needed. The modification of the **MAXARC** argument of the **ARCLen** command can also help the program to converge by preventing the reference arc-length radius from increasing too rapidly (**MAXARC** = 1).

When the arc-length method is active, do not use line search (**LNSRCH**), the predictor (**PRED**), adaptive descent (**NROPT,,,ON**), automatic time stepping (**AUTOTS, DELTIM**), or time-integration effects (**TIMINT**). Likewise, do not try to base convergence on displacement (**CNVTOL,U**); instead, use the force criteria (**CNVTOL,F**).

If an arc-length solution fails to converge within the prescribed maximum number of iterations (**NEQIT**), the program automatically bisections and continues the analysis. Bisection continues until a converged solution is obtained or until the minimum arc-length radius is used. (The minimum radius is defined by **NSBSTP** (**NSUBST**) and **MINARC** (**ARCLen**)).

In general, you cannot use this method to obtain a solution at a specified load or displacement value because the value changes (along the spherical arc) as equilibrium is achieved. [Figure 8.4: Traditional](#)

[Newton-Raphson Method vs. Arc-Length Method \(p. 196\)](#) illustrates how the specified load  $F_1^a$  is used only as a starting point. The actual load at convergence is somewhat less. Similarly, it can be difficult to determine a value of limiting load or deflection within some known tolerance when using the arc-length method in a nonlinear buckling analysis. Generally, you must adjust the reference arc-length radius (**NSUBST**) by trial-and-error to obtain a solution at the limit point.

You can freely switch from the Newton-Raphson iteration method to the arc-length method at the start of any load step. However, to switch from arc-length to Newton-Raphson iterations, you must terminate the analysis and restart, deactivating the arc-length method in the first load step of the restart (**ARCLEN,OFF**).

An arc-length solution terminates under these conditions:

- When limits defined by the **ARCTRM** or **NCNV** commands are reached
- When the solution converges at the applied load
- When you use an abort file (`Jobname.ABT`)

See the *Basic Analysis Guide* for information about termination and restart procedures.

Use the load-deflection curve as a guide for evaluating and adjusting your analysis to help you achieve the desired results. It is usually good practice to graph your load-deflection curve (using **POST26 commands**) with every analysis.

The total arc-length load factor (**SOLU,,ALLF**) can be either positive or negative. Similarly, *TIME*, which in an arc-length analysis is related to the total arc-length load factor, can also be either positive or negative. Negative values of *ALLF* or *TIME* indicate that the arc-length feature is applying load in the reverse direction in order to maintain stability in the structure. Negative *ALLF* or *TIME* values are commonly seen in various snap-through analyses.

### 8.11.2.1. Checking Arc-Length Results

When reading arc-length results into the database for POST1 postprocessing (**SET**), always reference the desired results data set by its load step and substep number (*LSTEP* and *SBSTEP*) or by its data set number (*NSET*).

Do *not* reference results by a *TIME* value, because *TIME* in an arc-length analysis is not always monotonically increasing. (A single value of *TIME* might reference more than one solution.) Additionally, the program cannot correctly interpret negative *TIME* values (which might be encountered in a snap-through analysis).

If *TIME* becomes negative, define an appropriate variable range (**/XRANGE** or **/YRANGE**) before creating any POST26 graphs.

### 8.11.3. Nonlinear Stabilization vs. the Arc-Length Method

You can use nonlinear stabilization for both local and global instability with few limitations related to compatibility with other algorithms and materials. However, nonlinear stabilization cannot detect the negative-slope portion of a load-vs.-displacement curve problem with global instability (if any).

Although the results obtained before the negative slope portion of the problem are always correct, the results for the substeps *after* the negative-slope portion are also correct *if the materials are not deformation-history-dependent*. (Consider the results to be questionable if the materials *are* deformation-history-dependent.)

The arc-length method *can* detect the negative-slope portion of a load-vs.-displacement curve, but it cannot solve problems with local instability and material softening. Other limitations exist, related mostly to compatibility with certain algorithms and materials.



To help you understand when to use either method, the following table compares both methods and their applications:

<b>Nonlinear Stabilization vs. Arc-Length</b>		
<b>Analysis Problem</b>	<b>Nonlinear Stabilization</b>	<b>Arc-Length</b>
Local instability or local buckling	Yes	Yes
Global instability or global buckling	Yes	Yes
Negative slope of load-vs.-displacement curve	Cannot detect this part of the curve, but other parts can be simulated for deformation-history-independent materials, and the preceding part can be simulated for deformation- history-depend-ent materials	Yes
Rate-dependent materials and creep	Yes	Yes
Line search	Yes	No
Substep predictor ( <b>PRED,ON</b> )	Yes (1)	Different algorithm
<a href="#">Automatic time stepping</a>	Yes	Different algorithm
Displacements as load	Yes	Yes
Activate/deactivate from load step to load step, or within a load step	Yes	Limited
Linear solver use	No restrictions	Sparse solver only
Time at converged substep	Positive	Positive or negative
Nonlinear contact	Yes	No (except for bonded contact)

1. Solid elements only.

## 8.12. Guidelines for Nonlinear Analysis

This section provides information to help you perform a successful nonlinear analysis.

ANSYS, Inc. recommends using [current-technology elements](#) for nonlinear analyses.

The following topics are available:

[8.12.1. Setting Up a Nonlinear Analysis](#)

[8.12.2. Overcoming Convergence Problems](#)

### 8.12.1. Setting Up a Nonlinear Analysis

By taking your time and proceeding carefully, you can avoid many difficulties commonly associated with nonlinear analyses.

Consider these suggestions:

[8.12.1.1. Understand Your Program and Structure Behavior](#)

[8.12.1.2. Simplify Your Model](#)

[8.12.1.3. Use an Adequate Mesh Density](#)

[8.12.1.4. Apply Loading Gradually](#)

### **8.12.1.1. Understand Your Program and Structure Behavior**

If you have not used a particular nonlinear feature before, construct a very simple model (containing only a few elements), and make sure you understand how to handle this feature before you use it in a large, complicated model.

Gain preliminary insight into your structure's behavior by analyzing a preliminary simplified model first. For nonlinear static models, a preliminary *linear* static analysis can reveal which regions of your model will first experience nonlinear response, and at what load levels these nonlinearities will come into play. For nonlinear transient dynamic analyses, a preliminary model of beams, masses, and springs can provide insight into the structure's dynamics at minimal cost. Preliminary nonlinear static, linear transient dynamic, and/or modal analyses can also help you to understand various aspects of your structure's nonlinear dynamic response before you undertake the final nonlinear transient dynamic analysis.

Read and understand the program's output messages and warnings. At a minimum, before you try to postprocess your results, verify that your problem converged. For path-dependent problems, the printout's equilibrium iteration record can be especially important in helping you to determine if your results are valid or not.

### **8.12.1.2. Simplify Your Model**

Keep your final model as simple as possible. For example, if applicable to your analysis, try any or all of the following:

- Represent your 3-D structure as a 2-D plane stress, plane strain, or axisymmetric model.
- Reduce your model size by using symmetry or antisymmetry surfaces.

If your model is loaded antisymmetrically, however, you can generally *not* take advantage of antisymmetry to reduce a nonlinear model's size. Antisymmetry can also be rendered inapplicable by large deflections.

- Omit a nonlinear detail if you can do so without affecting results in critical regions of your model.
- Model transient dynamic loading in terms of static-equivalent loads.
- Consider [substructuring](#) the linear portions of your model to reduce the computational effort required for intermediate load or time increments and equilibrium iterations.

### **8.12.1.3. Use an Adequate Mesh Density**

Recognize that regions undergoing plastic deformation require a reasonable integration point density (mesh density is particularly important in plastic-hinge regions).

Provide an adequate mesh density on contact surfaces to allow contact stresses to be distributed in a smooth fashion. Likewise, provide a mesh density adequate for resolving stresses; areas where stresses or strains are of interest require a relatively fine mesh compared to that needed for displacement or nonlinearity resolution.

Use a mesh density adequate to characterize the highest mode shape of interest. The number of elements needed depends on the elements' assumed displacement shape functions, as well as on the mode shape itself. Also, use a mesh density adequate to resolve any transient dynamic wave propagation through your structure; if wave propagation is important, then provide at least 20 elements to resolve one wavelength.

### 8.12.1.4. Apply Loading Gradually

For nonconservative, path-dependent systems, you need to apply the load in small enough increments to ensure that your analysis closely follows the structure's load-response curve.

You can sometimes improve the convergence behavior of conservative systems by applying the load gradually, so as to minimize the number of Newton-Raphson equilibrium iterations required.

## 8.12.2. Overcoming Convergence Problems

This section provides information to help you fix convergence problems in a nonlinear analysis. The following topics are available:

8.12.2.1. Overview of Convergence Problems

8.12.2.2. Performing Nonlinear Diagnostics

8.12.2.3. Tracking Convergence Graphically

8.12.2.4. Automatic Time Stepping

8.12.2.5. Line Search

8.12.2.6. Nonlinear Stabilization

8.12.2.7. Arc-Length Method

8.12.2.8. Artificially Inhibit Divergence in Your Model's Response

8.12.2.9. Use the Rezoning Feature

8.12.2.10. Dispense with Extra Element Shapes

8.12.2.11. Using Element Birth and Death Wisely

8.12.2.12. Read Your Output

8.12.2.13. Graph the Load and Response History

### 8.12.2.1. Overview of Convergence Problems

When performing a nonlinear analysis you may encounter convergence difficulties due to a number of reasons. Some examples may be initially open contact surfaces causing rigid body motion, large load increments causing nonconvergence, material instabilities, or large deformations causing mesh distortion that result in element shape errors. Solution control (**SOLCONTROL**) automatically adjusts solution parameters and attempts to obtain a robust, accurate solution. In addition, several tools are available to help you identify potential problems before, during, and after a solution.

**CHECK**, **MCHECK**, and **CNCHECK** commands help you verify if there are any obvious problems with the model before you start the solution. The **CHECK** command does an overall verification of the model, including missing elastic properties, unconstrained model, and element shape checks. The **MCHECK** command can help you identify defects in the mesh such as holes or cracks, especially when the mesh is imported from a third party software. The **CNCHECK** command provides the initial contact status of contact pairs, identifying whether the contacts are initially open or closed. If, for example, a part in your model is constrained only through contact with other parts and if the contact surfaces are open, the **CNCHECK** command can help you identify this potential error condition.

When you analyze models with large deformations, some portions of the initial mesh can become highly distorted. Highly distorted elements can take on unacceptable shapes, providing inaccurate results. This can cause your nonlinear solution to stop. When this happens, use the **ESCHECK** command to perform shape checking of deformed elements in the postprocessor (based on the current set of results in database). This deformed-shape checker helps you to identify the portions of your model that require different meshing, thereby allowing them to retain acceptable shapes. Using **ESCHECK** at different time points helps you to identify the load conditions that cause mesh deterioration.

A convergence failure can also indicate a physical instability in the structure, or it can merely be the result of some numerical problem in the finite element model.

The following sections detail some of the techniques that you can use to attempt to improve the convergence performance of your analysis.

### **8.12.2.2. Performing Nonlinear Diagnostics**

The nonlinear diagnostics tool **NLDIAG** can help you find problems in your model when an analysis does not converge.

**Identify Regions of High Residual Forces** Issue the **NLDIAG,NRRE** command to write the Newton-Raphson residuals from equilibrium iterations to a file (`Jobname.nrxxx`). You can then contour plot the residual forces via the **PLNSOL,NRRE** command, which helps to identify regions of high residual forces.

Such a capability can be useful when you experience convergence difficulties in the middle of a load step, where the model has a large number of contact surfaces and other nonlinearities. You can restart the analysis and issue an **NLDIAG,NRRE** command to write out the residuals. By tracking the way the residuals change over several equilibrium iterations you can identify a portion of your model where large residuals persist. You can then focus on the nonlinearities in that area (for example, a contact pair's properties) instead of having to deal with the entire model.

**Identify Problem Elements** Typically, nonlinear analyses fail to converge for the following reasons:

- Too large a distortion
- Elements contain nodes that have near zero pivots (nonlinear analyses)
- Too large a plastic or creep strain increment
- Elements where mixed u-P constraints are not satisfied (mixed u-P option of [current-technology](#) solid elements only)

The program has default limits which, when exceeded, determine when convergence criteria have been violated. Some limits are user-controlled; for example, the **CUTCONTROL** command sets the maximum plastic/creep strain increments allowed in an iteration. Other limits are fixed.

The **NLDIAG,EFLG** command identifies elements that violate the above criteria and records them in a file (`Jobname.ndxxx`).

Convergence problems may occur when material algorithms fail (for example, local element level Newton-Raphson convergence failure, or extreme element distortion). The error message identifies the corresponding element number and/or the material ID for these cases. Be sure to read any error messages generated during solution.

**Process the Tracked Results** Issue the **NLDPOST** command to process the `.ndxxx` nonlinear diagnostics files. The command creates components of elements that violate a certain criterion for a particular equilibrium iteration (or iterations).

**Identify contact pairs causing convergence difficulties** Issue the **NLDIAG,CONT** command to write various contact information for all defined contact pairs to a single `Jobname.cnd` text file. The file is written during solution at a user-specified frequency (each iteration, substep, or load step).

Information stored in this file helps identify when and how contact occurs, determine the regions where contact is unstable, and identify the corresponding contact parameters. You can then focus on the specific settings for those particular contact pairs that need attention.

**Monitor the Diagnostics Results in Real Time** The **NLHIST** command allows you to monitor results of interest in real time during solution. Before starting the solution, you can request nodal data such as displacements or reaction forces at specific nodes. You can also request element nodal data such as stresses and strains at specific elements to be graphed. Pair-based contact data are also available. The result data are written to a file named `Jobname.nlh`.

For example, a reaction force-deflection curve could indicate when possible buckling behavior occurs. Nodal results and contact results are monitored at every converged substep while element nodal data are written as specified via the **OUTRES** setting.

You can also track results during batch runs. Either access the ANSYS Launcher and select **File Tracking** from the **Tools** menu, or type `nlhist150` at the command line. Use the supplied file browser to navigate to your `Jobname.nlh` file, and click on it to invoke the tracking utility. You can use this utility to read the file at any time, even after the solution is complete (the data in the file must be formatted correctly).

### 8.12.2.3. Tracking Convergence Graphically

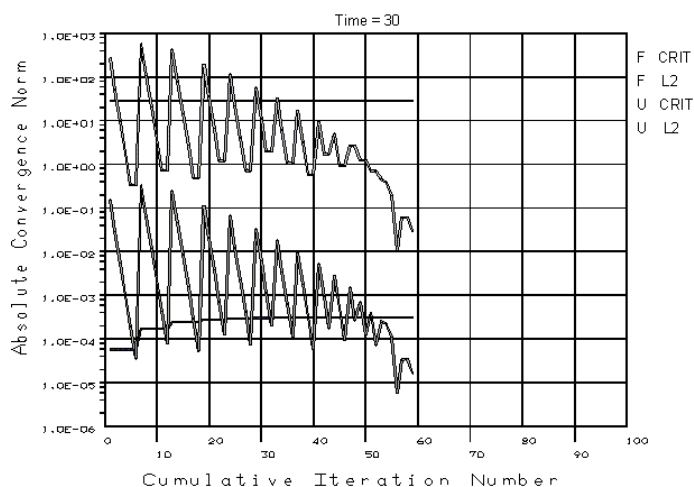
As a nonlinear structural analysis proceeds, the program calculates convergence norms with corresponding convergence criteria each equilibrium iteration. Available in both batch and interactive sessions, the Graphical Solution Tracking (GST) feature displays the computed convergence norms and criteria while the solution is in process. By default, GST is ON for interactive sessions and OFF for batch runs. To turn GST on or off, use either of the following:

**Command(s):** `/GST`

**GUI:** Main Menu > Solution > Load Step Opts > OutputCtrls > Grph Solu Track

Figure 8.15: Convergence Norms Displayed By the Graphical Solution Tracking (GST) Feature (p. 269) shows a typical GST display:

**Figure 8.15: Convergence Norms Displayed By the Graphical Solution Tracking (GST) Feature**



#### 8.12.2.4. Automatic Time Stepping

Place an upper limit on the time step size (**DELTIM** or **NSUBST**), especially for complicated models. Doing so ensures that all of the modes and behaviors of interest are accurately included. This can be important in the following situations:

- Problems that have only localized dynamic behavior (for example, turbine blade and hub assemblies) in which the low-frequency energy content of the system could dominate the high-frequency areas.
- Problems with short ramp times on some of their loads. If the time step size is allowed to become too large, ramped portions of the load history may be inaccurately characterized.
- Problems that include structures that are continuously excited over a range of frequencies (for example, seismic problems).

Exercise caution when modeling kinematic structures (systems with rigid-body motions). These following guidelines can usually help you to obtain a good solution:

- Incorporate significant numerical damping ( $0.05 < \gamma < 0.1$  on the **TINTP** command) into the solution to filter out the high frequency noise, especially if a coarse time step is used. Do not use  $\alpha$ -damping (mass matrix multiplier, **ALPHAD** command) in a dynamic kinematic analysis, as it dampens the rigid body motion (zero frequency mode) of the system.
- Avoid imposed displacement history specifications, because imposed displacement input has (theoretically) infinite jumps in acceleration, which causes stability problems for the Newmark time-integration algorithm.

#### 8.12.2.5. Line Search

Line search (**LNSRCH**) can be useful for enhancing convergence, but it can be expensive (especially with plasticity). You might consider setting line search on in the following cases:

- When your structure is force-loaded (as opposed to displacement-controlled).
- If you are analyzing a "flimsy" structure which exhibits increasing stiffness (such as a fishing pole).
- If you notice (from the program output messages) oscillatory convergence patterns.

#### 8.12.2.6. Nonlinear Stabilization

You can use the nonlinear stabilization method to solve both locally and globally unstable problems, and to overcome convergence for general problems. For more information, see [Using Nonlinear Stabilization](#) (p. 258).

#### 8.12.2.7. Arc-Length Method

You can use the arc-length method (**ARCLEN** and **ARCTRM**) to obtain numerically stable solutions for many physically unstable structures. For more information, see [Using the Arc-Length Method](#) (p. 263).

#### 8.12.2.8. Artificially Inhibit Divergence in Your Model's Response

If you do not want to use both nonlinear stabilization and the arc-length method to analyze a force-loaded structure that starts at, or passes through, a singular (zero stiffness) configuration, you can sometimes use other alternatives to artificially inhibit divergence in your model's response:

- In some cases, you can use imposed displacements instead of applied forces. This approach can be used to start a static analysis closer to the equilibrium position, or to control displacements through periods of unstable response (for example, snap-through or postbuckling).
- Another technique that can be effective in circumventing problems due to initial instability is running a static problem as a "slow dynamic" analysis (that is, using time-integration effects in an attempt to prevent the solution from diverging in any one load step).
- You can also apply temporary artificial stiffness to unstable DOFs, using control elements (such as [COMBIN37](#)), or using the birth and death option on other elements. The idea here is to artificially restrain the system during intermediate load steps in order to prevent unrealistically large displacements from being calculated. As the system displaces into a stable configuration, the artificial stiffness is removed.

### 8.12.2.9. Use the Rezoning Feature

If the solution fails to converge and the mesh is severely distorted, consider using [rezoning](#), which allows you to repair the distorted mesh and continue the simulation. For more information, see [Rezoning](#) in the *Advanced Analysis Guide*.

### 8.12.2.10. Dispense with Extra Element Shapes

The program provides "incompatible" modes" formulation (also referred to as "extra shapes") for modeling bending applications. If your problem is predominantly bulk deformation, then you may choose to turn extra shapes off to reduce CPU/storage requirements and enhance convergence. However, doing so precludes the ability to model any bending.

### 8.12.2.11. Using Element Birth and Death Wisely

Realize that any sudden change in your structure's stiffness matrix is likely to cause convergence problems. When activating or deactivating elements, try to spread the changes out over a number of substeps. (Use a small time step size if necessary to accomplish this.) Also be aware of possible singularities (such as sharp reentrant corners) that might be created as you activate or deactivate elements. Such singularities can cause convergence problems.

### 8.12.2.12. Read Your Output

The program performs a nonlinear analysis as a series of linear approximations with corrections. The program printout gives you continuous feedback on the progress of these approximations and corrections. (Printout either appears directly on your screen, is captured on `Jobname.OUT`, or is written to some other file (`/OUTPUT`.) You can examine some of this same information in `POST1`, using the [PRITER](#) command, or in `POST26`, using the [SOLU](#) and [PRVAR](#) commands. You should make sure that you understand the iteration history of your analysis before you accept the results. In particular, do not dismiss any program error or warning statements without fully understanding their meaning. A typical nonlinear output listing is shown in [Figure 8.16: Typical Nonlinear Output Listing \(p. 271\)](#).

**Figure 8.16: Typical Nonlinear Output Listing**

```
SOLVE      ***** ANSYS SOLVE      COMMAND      *****
com-
mand
echo
```



**Check-  
ing Lo-  
gic**

\*\*\* NOTE \*\*\* CP= 13.891 TIME= 11:09:22  
 Nonlinear analysis, NROPT set to 1 (full Newton-Raphson solution  
 procedure) for all DOFs.

**Load  
step  
summary  
table**

L O A D S T E P O P T I O N S

LOAD STEP NUMBER. . . . . 2  
 TIME AT END OF THE LOAD STEP. . . . . 200.00  
 AUTOMATIC TIME STEPPING . . . . . ON  
     INITIAL NUMBER OF SUBSTEPS . . . . . 100  
     MAXIMUM NUMBER OF SUBSTEPS . . . . . 10000  
     MINIMUM NUMBER OF SUBSTEPS . . . . . 10  
 MAXIMUM NUMBER OF EQUILIBRIUM ITERATIONS. . . . . 15  
 STEP CHANGE BOUNDARY CONDITIONS . . . . . NO  
 TERMINATE ANALYSIS IF NOT CONVERGED . . . . .YES (EXIT)  
 CONVERGENCE CONTROLS. . . . .USE DEFAULTS  
 PRINT OUTPUT CONTROLS . . . . .NO PRINTOUT  
 DATABASE OUTPUT CONTROLS  
     ITEM      FREQUENCY    COMPONENT  
     BASI         -10

**Load  
step 2  
substep  
1**

FORCE CONVERGENCE VALUE = 0.2006E+06 CRITERION= 1125.  
 EQUIL ITER 1 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= 0.1272E-01  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = 0.1272E-01  
 FORCE CONVERGENCE VALUE = 4267. CRITERION= 480.2  
 EQUIL ITER 2 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= -0.9019E-03  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = -0.9019E-03  
 FORCE CONVERGENCE VALUE = 1751. CRITERION= 488.2  
 EQUIL ITER 3 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= 0.1746E-03  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = 0.1746E-03  
 FORCE CONVERGENCE VALUE = 778.5 CRITERION= 497.7  
 EQUIL ITER 4 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= 0.6943E-04  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = 0.6943E-04  
 FORCE CONVERGENCE VALUE = 347.4 CRITERION= 507.7 <<< CONVERGED  
 >>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 4  
 \*\*\* LOAD STEP 2 SUBSTEP 1 COMPLETED. CUM ITER = 7  
 \*\*\* TIME = 101.000 TIME INC = 1.00000

**Load  
step 2  
substep  
2**

FORCE CONVERGENCE VALUE = 0.6674E+05 CRITERION= 594.3  
 EQUIL ITER 1 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= 0.4318E-02  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = 0.4318E-02  
 FORCE CONVERGENCE VALUE = 626.2 CRITERION= 502.9  
 EQUIL ITER 2 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= 0.8570E-04  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = 0.8570E-04  
 FORCE CONVERGENCE VALUE = 77.87 CRITERION= 512.9 <<< CONVERGED  
 >>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 2  
 \*\*\* LOAD STEP 2 SUBSTEP 2 COMPLETED. CUM ITER = 9  
 \*\*\* TIME = 102.000 TIME INC = 1.00000

**Load  
step 2  
substep  
3 Equil-  
brium it-  
eration  
summar-  
ies**

FORCE CONVERGENCE VALUE = 0.1333E+05 CRITERION= 575.4  
 EQUIL ITER 1 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= -0.5329E-02  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = -0.5329E-02  
 FORCE CONVERGENCE VALUE = 8237. CRITERION= 534.2  
 EQUIL ITER 2 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= 0.3628E-02  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = 0.3628E-02  
 FORCE CONVERGENCE VALUE = 3905. CRITERION= 532.9  
 EQUIL ITER 3 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= 0.1451E-02  
 LINE SEARCH PARAMETER = 1.000 SCALED MAX DOF INC = 0.1451E-02  
 FORCE CONVERGENCE VALUE = 1135. CRITERION= 540.3  
 EQUIL ITER 4 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= 0.1034E-03  
 LINE SEARCH PARAMETER = 0.9578 SCALED MAX DOF INC = 0.9905E-04  
 FORCE CONVERGENCE VALUE = 41.95 CRITERION= 551.4 <<< CONVERGED  
 >>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 4

**Substep  
summary**

\*\*\* LOAD STEP 2 SUBSTEP 3 COMPLETED. CUM ITER = 13  
 \*\*\* TIME = 103.500 TIME INC = 1.50000



### 8.12.2.13. Graph the Load and Response History

This verification technique may be considered to be a graphical combination of two other techniques: checking for reasonableness, and reviewing the iteration history. POST26 graphs of load and response histories should agree with your informed expectations about your structure's behavior. The results of interest (displacements, reaction forces, stresses, and so on) should show relatively smooth response histories. Any non-smoothness may indicate that too coarse of a time step was used.

## 8.13. Example Nonlinear Analysis (GUI Method)

This example runs a nonlinear analysis of an elastic-plastic circular plate under the action of a dead load and a cyclic point load. You define a kinematic hardening plasticity curve, as well as load step options, the maximum and minimum number of substeps for a load step, and the various load steps that describe externally applied loads. You also learn how to interpret the monitor file that the program writes for a nonlinear analysis.

The program uses an incremental solution procedure to obtain a solution to a nonlinear analysis. In this example, the total external load within a load step is applied in increments over a certain number of substeps. The program uses a Newton-Raphson iterative procedure to solve each substep. You must specify the number of substeps for each load step, since this number controls the size of the initial load increment applied in the first substep of the each load step. The program automatically determines the size of the load increment for each subsequent substep in a load step. You can control the size of the load increment for these subsequent substeps by specifying the maximum and minimum number of substeps. If you define the number of substeps, the maximum and minimum number of substeps all to be the same, then The program uses a constant load increment for all substeps within the load step.

### 8.13.1. Problem Description

Use an axisymmetric model for the plate, using four-node [PLANE182](#) elements with the axisymmetric option to mesh the model. Perform a geometrically nonlinear analysis. Specify the kinematic constraints as follows: The nodes located at the center of the plate are constrained to have zero radial displacement. The nodes located at the outer edge are constrained to have zero radial and axial displacement. Apply the dead load in load step 1 and the cyclic point load in six subsequent load steps. See [Problem Sketch](#) (p. 274).

Specify 10 substeps for the first load to ensure that the increment of the dead load applied over the first substep is 1/10 of the total load of  $0.125 \text{ N/m}^2$ . Also specify a maximum of 50 and a minimum of 5 substeps to ensure that if the plate exhibits a severe nonlinear behavior during the solution, then the load increment can be cut back to 1/50 the total load. If the plate exhibits mild nonlinear behavior, then the load increment can be increased up to 1/5 the size of the total load.

For the subsequent six load steps that apply the cyclic point load, 4 substeps, with a maximum of 25 and a minimum of 2 substeps.

Monitor the history over the entire solution of the vertical displacement of the node at the location where the point cyclic load is applied and the reaction force at the node located at the bottom of the clamped edge.

### 8.13.2. Problem Specifications

The circular plate has a radius of 1.0 m and a thickness of 0.1 m. The following material properties are used for this problem:

EX = 16911.23 Pa

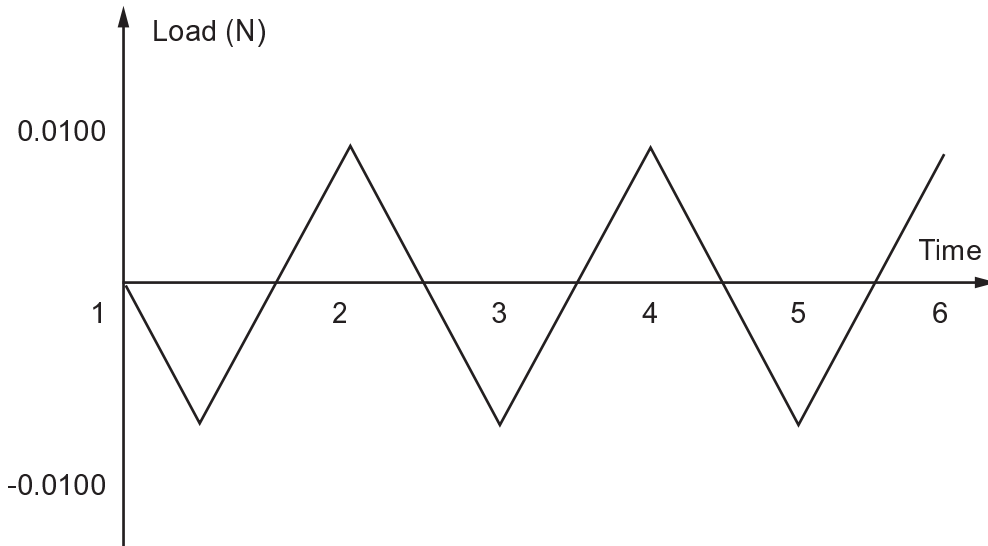
PRXY = 0.3

The kinematic hardening plasticity curve for the material is:

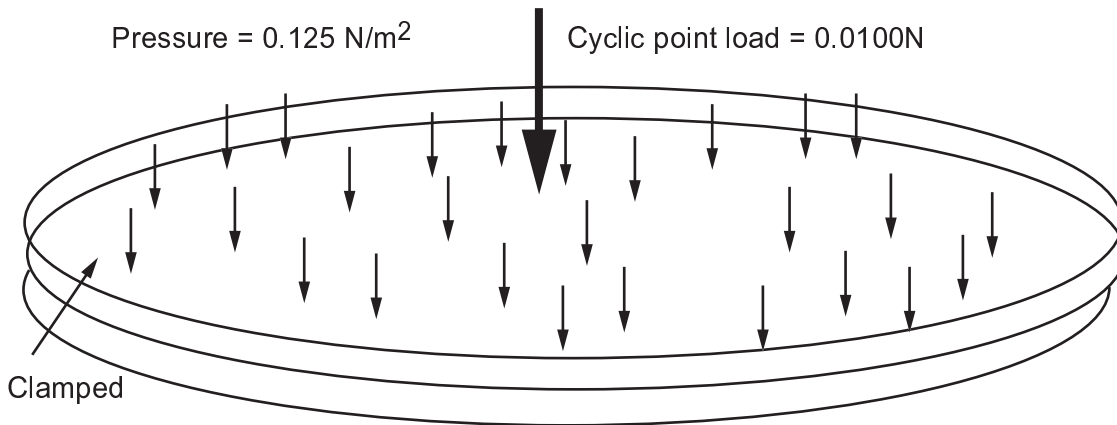
Log Strain	True Stress (Pa)
0.001123514	19.00
0.001865643	22.80
0.002562402	25.08
0.004471788	29.07
0.006422389	31.73

The plate has a dead load acting as a uniform pressure of  $0.125\text{N/m}^2$ . The history of the cyclic point load is shown here:

**Figure 8.17: Cyclic Point Load History**



### 8.13.3. Problem Sketch



### **8.13.3.1. Set the Analysis Title and Jobname**

1. Choose menu path **Utility Menu> File> Change Title**.
2. Type the text "Cyclic loading of a fixed circular plate."
3. Click on OK.
4. Choose menu path **Utility Menu> File> Change Jobname**. The Change Jobname dialog box appears.
5. Type the text "axplate" in the entry box and click OK.

### **8.13.3.2. Define the Element Types**

1. Choose menu path **Main Menu> Preprocessor> Element Type> Add/Edit/Delete**.
2. Click on Add. The Library of Element Types dialog box appears.
3. In the list on the left, click once on "Structural Solid."
4. In the list on the right, click once on "Quad 4node 182."
5. Click on OK. The Library of Element Types dialog box closes.
6. Click on Options. The **PLANE182** element type options dialog box appears.
7. In the scroll box for element behavior, scroll to "Axisymmetric" and select it.
8. Click on OK.
9. Click on Close in the Element Types dialog box.

### **8.13.3.3. Define Material Properties**

1. Choose menu path **Main Menu> Preprocessor> Material Props> Material Models**. The Define Material Model Behavior dialog box appears.
2. In the Material Models Available window, double-click on the icons next to the following options: Structural, Linear, Elastic, Isotropic. A dialog box appears.
3. Enter 16911.23 for EX (Young's modulus).
4. Enter .3 for PRXY (Poisson's ratio).
5. Click on OK. Material Model Number 1 appears in the Material Models Defined window on the left.

### **8.13.3.4. Specify the Kinematic Hardening material model (KINH)**

1. In the Material Models Available window, double-click on the following options: Nonlinear, Inelastic, Rate Independent, Kinematic Hardening Plasticity, von Mises Plasticity, Multilinear (General). A dialog box appears.
2. Enter the following Strain/Stress value pair in the table: 0.001123514, 19.00
3. Click on the **Add Point** button, and enter the next Strain/Stress value pair: 0.001865643, 22.80

4. Repeat the previous step to enter the following Strain/Stress value pairs: 0.002562402, 25.08; 0.004471788, 29.07; 0.006422389, 31.73.
5. Click on OK.
6. Choose menu path **Material> Exit** to remove the Define Material Model Behavior dialog box.

### **8.13.3.5. Label Graph Axes and Plot Data Tables**

1. Choose menu path **Utility Menu> PlotCtrls> Style> Graphs> Modify Axes**. The Axes Modifications for Graph Plots dialog box appears.
2. Enter Total Strain for the X-axis label.
3. Enter True Stress for the Y-axis label and click OK.
4. Choose menu path **Main Menu> Preprocessor> Material Props> Material Models**. The Define Material Model Behavior dialog box appears.
5. In the Material Models Defined window, double-click in Material Model Number 1, and Multilinear Kinematic (General). The dialog box appears that includes the Strain/Stress data pairs that you entered.
6. Click on the Graph button. A graph of the data table values appears in the Graphics window. If necessary, revise the stress/strain values and click on the Graph button again. Repeat revisions and graphing as needed until you are satisfied with the graphed results. Click on OK.
7. Choose menu path **Material> Exit** to remove the Define Material Model Behavior dialog box.
8. Click on SAVE\_DB on the Toolbar.

### **8.13.3.6. Create Rectangle**

1. Choose menu path **Utility Menu> Parameters> Scalar Parameters**. The Scalar Parameters dialog box appears.
2. Type "radius=1.0" in the Selection field and click Accept. This value is the radius of the plate.
3. Type "thick=0.1" in the Selection field and click Accept. This value is the thickness of the plate. Click Close.
4. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Areas> Rectangle> By Dimensions**. The Create Rectangle by Dimensions dialog box appears.
5. Enter "0, radius" for X-coordinates.
6. Enter "0, thick" for Y-coordinates and click on OK. A rectangle appears in the Graphics window.
7. Choose menu path **Utility Menu> Plot> Lines**.

### **8.13.3.7. Set Element Size**

1. Choose menu path **Main Menu> Preprocessor> Meshing> MeshTool**. The MeshTool dialog box appears.

2. Click **Size Controls> Lines> Set**. The Element Size on Picked Lines picking menu appears. Click on the two vertical lines (2 and 4). Click OK on the picking menu. The Element Sizes on Picked Lines dialog box appears.
3. Enter 8 for number of element divisions and click on OK.
4. Repeat these steps (1-3), but choose horizontal lines 1 and 3, and specify 40 element divisions.

### **8.13.3.8. Mesh the Rectangle**

1. On the MeshTool, pick Quad and Map, then click MESH. The Mesh Areas picking menu appears.
2. Click on Pick All.
3. Click on SAVE\_DB on the Toolbar.
4. Click on Close on the MeshTool.

### **8.13.3.9. Assign Analysis and Load Step Options**

1. Choose menu path **Main Menu> Solution> Unabridged Menu> Analysis Type> Analysis Options**. The Static or Steady-State Analysis dialog box appears.
2. Turn large-deformation effects ON and click OK.
3. Choose menu path **Main Menu> Solution> Load Step Opts> OutputCtrls> DB/Results File**. The Controls for Database and Results File Writing dialog box appears.
4. Verify that All items are selected, and choose Every substep for the File write frequency. Click OK.

### **8.13.3.10. Monitor the Displacement**

In this step, you monitor the displacement of the node located at the axes of symmetry, as well as the reaction force at the fixed end of the plate.

1. Choose menu path **Utility Menu> Parameters> Scalar Parameters**. The Scalar Parameters dialog box appears.
2. Type "ntop = node(0,thick,0.0)" in the Selection field and click Accept.
3. Type "nright = node(radius,0.0,0.0)" in the Selection field and click Accept, then Close.
4. Choose menu path **Main Menu> Solution> Load Step Opts> Nonlinear> Monitor**. The Monitor picking menu appears.
5. Type "ntop" in the picker and press RETURN. Click OK in the picking menu. The Monitor dialog box appears.
6. In the scroll box for Quantity to be monitored, scroll to "UY" and select it. Click OK.
7. Choose menu path **Main Menu> Solution> Load Step Opts> Nonlinear> Monitor**. The Monitor picking menu appears.
8. Type "nright" in the picker and press RETURN. Click OK in the picking menu. The Monitor dialog box appears.

9. In the scroll box for Variable to redefine, scroll to "Variable 2" and select it. In the scroll box for Quantity to be monitored, scroll to "FY" and select it. Click OK.

### **8.13.3.11. Apply Constraints**

1. Choose menu path **Utility Menu> Select> Entities**. The Select Entities dialog box appears.
2. Select Nodes and By Location in the first two selection boxes. Verify that X coordinates are selected, and enter "radius" in the Min, Max field. Click OK.
3. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On Nodes**. The Apply U,ROT on Nodes picking menu appears.
4. Click Pick All. The Apply U,ROT on Nodes dialog box appears.
5. Click on "All DOF" for DOFs to be constrained. Click OK.
6. Choose menu path **Utility Menu> Select> Entities**. The Select Entities dialog box appears. Verify that Nodes, By Location, and X coordinates are selected. Enter "0" in the Min, Max field and click OK. This action selects the nodes at the X=0 position.
7. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On Nodes**. The Apply U,ROT on Nodes picking menu appears.
8. Click Pick All. The Apply U,ROT on Nodes dialog box appears.
9. Click on "UX" for DOFs to be constrained. Click on All DOF to deselect it.
10. Enter "0.0" as the Displacement value. Click OK.
11. Choose menu path **Utility Menu> Select> Entities**. The Select Entities dialog box appears. Verify that Nodes and By Location are selected.
12. Click on Y coordinates and enter "thick" in the Min, Max field. Click OK.
13. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Pressure> On Nodes**. The Apply PRES on Nodes picking menu appears.
14. Click on Pick All. The Apply PRES on nodes dialog box appears.
15. Enter "0.125" in the Load PRES value field and click OK.
16. Choose menu path **Utility Menu> Select> Everything**.
17. Click on SAVE\_DB on the Toolbar.

### **8.13.3.12. Solve the First Load Step**

1. Choose menu path **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time and Substps**. The Time and Substep Options dialog appears.
2. Enter 10 as the number of substeps, enter 50 as the maximum number of substeps, and enter 5 as the minimum number of substeps. Click OK.
3. Choose menu path **Main Menu> Solution> Solve> Current LS**. Review the information in the /STAT window, and click on Close.

4. Click on OK on the Solve Current Load Step dialog box.
5. Click on Close on the Information dialog box when the solution is done.
6. Choose **Utility Menu> Plot> Elements**.

### **8.13.3.13. Solve the Next Six Load Steps**

1. Choose **Utility Menu> Parameters> Scalar Parameters**. The Scalar Parameters dialog box appears.
2. Enter "f = 0.010" in the Selection field and click on Accept. Click on Close.
3. Choose menu path **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time and Substps**. The Time and Substep Options dialog appears.
4. Enter "4" for the number of substeps, "25" for the maximum number of substeps, and "2" for the minimum number of substeps. Click OK.
5. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Force/Moment> On Nodes**. The Apply F/M on Nodes picking menu appears.
6. Enter "ntop" in the picker and press RETURN. Click OK in the Apply F/M on Nodes picking menu. The Apply F/M on Nodes dialog box appears.
7. Select "FY" in the Direction of force/mom selection box. Enter "-f" in the Force/moment value field. Click OK.
8. Choose menu path **Main Menu> Solution> Solve> Current LS**. Review the information in the /STAT window, and click on Close.
9. Click on OK on the Solve Current Load Step dialog box.
10. Click on Close on the Information dialog box when the solution is done.
11. Repeat Steps 5-10, entering "f" in the Force/moment value field at Step 7.
12. Repeat Steps 5-11 two more times, for a total of three cycles (six substeps).
13. Click on SAVE\_DB on the Toolbar.

### **8.13.3.14. Review the Monitor File**

1. Choose menu path **Utility Menu> List> Files> Other**. The List File dialog box appears. Select the `ax-plate.mntr` file and click on OK.
2. Review the time step size, vertical displacement, and reaction force evolution over the entire solution.
3. Click Close.

### **8.13.3.15. Use the General Postprocessor to Plot Results.**

1. Choose menu path **Main Menu> General Postproc> Read Results> Last Set**.
2. Choose menu path **Main Menu> General Postproc> Plot Results> Deformed Shape**. The Plot Deformed Shape dialog box appears.

3. Click on Def + undef edge for items to be plotted. Click OK. The deformed mesh appears in the Graphics window.
4. Choose menu path **Main Menu> General Postproc> Plot Results> Contour Plot> Element Solu**. The Contour Element Solution Data dialog box appears.
5. In the selection box on the left, choose Strain-plastic. In the selection box on the right, choose Eqv plastic EPEQ. Click OK. The contour plot appears in the Graphics window.
6. Choose **Utility Menu> Plot> Elements**.

### **8.13.3.16. Define Variables for Time-History Postprocessing**

1. Choose menu path **Utility Menu> Select> Entities**. The Select Entities dialog box appears.
2. Verify that Nodes and By Num/Pick are selected in the first two boxes. Click OK. The Select nodes picking menu appears.
3. Type "ntop" in the picker and press RETURN. Click OK.
4. Choose menu path **Utility Menu> Select> Entities**. The Select Entities dialog box appears. Choose Elements in the first drop-down selection box. Choose Attached to in the second drop-down selection box. Verify that Nodes is selected. Click OK.
5. Choose **Utility Menu> Select> Everything**.
6. Choose **Main Menu> TimeHist Postpro> Define Variables**. The Defined Time-History Variables dialog box appears. Click on Add. The Add Time-History Variable dialog box appears.
7. Click on Element results. Click OK. The Define Elemental Data picking menu appears.
8. Click on the top left element in the Graphics window. Click OK on the picking menu. The Define Nodal Data picking menu appears.
9. Click on the top left node of the top left element. Click OK on the picking menu. The Define Element Results Variable dialog box appears.
10. Verify that the reference number of the variable is 2.
11. Choose Stress in the selection list on the left. Choose Y-direction SY in the selection list on the right. Click OK. The Defined Time-History Variables dialog box reappears, with a second variable listed (ESOL). The dialog box should show element number 281, node number 50, item S, component Y, and name SY.
12. Click on Add. Repeat steps 7-10, with variable reference number 3.
13. In the Define Element Results Variable dialog box, choose Strain-elastic in the selection list on the left. Choose Y-dir'n EPEL Y in the selection list on the right. Click OK.
14. Click on Add. Repeat steps 7-10, with variable reference number 4.
15. In the Define Element Results Variable dialog box, choose Strain-plastic in the selection list on the left. Choose Y-dir'n EPPL Y in the selection list on the right. Click OK.
16. Click on Close on the Defined Time-History Variables dialog box.



17. Choose menu path **Main Menu> TimeHist Postpro> Math Operations> Add**. The Add Time-History Variables dialog box appears.
18. Enter 5 for the reference number for result, enter 3 as the 1st variable, and enter 4 as the 2nd variable. Click OK. This adds the elastic and plastic strains that you stored as variables 3 and 4. Their sum is the total strain, and it is stored as variable 5.

### 8.13.3.17. Plot Time-History Results

1. Choose menu path **Main Menu> TimeHist Postpro> Settings> Graph**. The Graph Settings dialog box appears.
2. Click on Single variable for the X-axis variable and enter 5 as the single variable number. Click OK.
3. Choose menu path **Utility Menu> PlotCtrls> Style> Graphs> Modify Axes**. The Axes Modifications for Graph Plots dialog box appears.
4. Enter Total Y-Strain as the X-axis label.
5. Enter Y-Stress as the Y-axis label. Click OK.
6. Choose menu path **Main Menu> TimeHist Postpro> Graph Variables**. The Graph Time-History Variables dialog box appears.
7. Enter 2 as the first variable to graph. Click OK.

### 8.13.3.18. Exit

1. Choose QUIT from the Toolbar.
2. Click on the save option you want, and click on OK.

## 8.14. Example Nonlinear Analysis (Command or Batch Method)

You can perform the example nonlinear static analysis of a copper cylinder impacting a rigid wall using the commands shown below instead of GUI choices. Items prefaced by an exclamation point (!) are comments.

```

/filename,axplate

/prep7
radius=1.0           ! radius of the plate (m)
thick=0.1            ! thickness of plate (m)
ym=16911.23
et,1,plane182,,,1    ! plane182 axisymmetric element
mp,ex,1,ym
mp,nuxy,1,0.3

! define a kinematic hardening plasticity curve using the kinh material model
tb,kinh,1,1,5

! define the true stress vs. total log strain curve for this material model
! using 5 points. first point defines the elastic limit

tbpt,,0.001123514,19.00
tbpt,,0.001865643,22.80
tbpt,,0.002562402,25.08
tbpt,,0.004471788,29.07

```

```

tbpt,,0.006422389,31.73

! set the axles labels for the stress-strain curve plot

/axlab,x,log strain (n/m^2)
/axlab,y,true stress (n/m^2)

tbpl,kinh,1           ! plot and verify the material stress-strain curve

! define a rectangle which is the axisymmetric cross section of the plate.
! the rectangle has a length equal to the radius of the plate and a height equal
! to the thickness of the plate

rect,,radius,,thick

! select the left and right bounding lines of the created rectangle and set
! the line division to 8 (8 elements through the thickness of the plate)

flst,5,2,4,orde,2
fitem,5,2
fitem,5,4
cm,_y,line
lsel,, , ,p51x
cm,_y1,line
cmsel,,_y
lesize,_y1, , ,8,1,
cmdel,_y
cmdel,_y1

! select the top and bottom bounding lines of the created rectangle and set
! the line division to 40 (40 elements through the radius of the plate)

flst,5,2,4,orde,2
fitem,5,1
fitem,5,3
cm,_y,line
lsel,, , ,p51x

cm,_y1,line
cmsel,,_y
lesize,_y1, , ,40,1,
cmdel,_y
cmdel,_y1

cm,_y,area
asel, , , , 1
cm,_y1,area
chkmsk,'area'
cmsel,s,_y

amesh,all
cmdel,_y
cmdel,_y1
cmdel,_y2
fini

/solve
nlgeom,on           ! turn on geometric nonlinearity

! get the node numbers for the nodes located at the top
! of the axis of symmetry and at bottom right of the model

ntop = node(0,thick,0)
nright = node(radius,0,0)

! activate the monitoring of the displacement and reaction force histories
! during the analysis. the histories are written out to the monitor file ratch.mntr

monitor,1,ntop,uy
monitor,2,nright,fy

outres,all,all           ! output all the results for all substeps to the

```

```

! results file for later postprocessing

! select the nodes located at right end and constrain their radial (x) and
! axial (y) direction displacement to be zero.

nselect,s,loc,x,radius
d,all,all

! select the nodes located at left end and constrain their radial (x) direction
! displacement to be zero.

nselect,s,loc,x,0.0
d,all,ux,0.0

! define the load for load step 1.
! select the nodes located at top surface of plate and apply a uniform pressure
! of 0.125 n/m^2 as dead load on the plate.

nselect,s,loc,y,thick
sf,all,pres,0.125

alls ! select all nodes

! define the number of substeps (10). also define maximum number of
! substeps (50), and the minimum number of substeps (5) for the automatic
! time stepping algorithm.

nsub,10,50,5
solve ! solve load step 1

f = 0.01 ! define the parameter, f, used to apply
! the cyclic point load.
! over six load steps apply a cyclic point load of magnitude f = 0.01 units
! applied at the center of the plate over three cycles

! *****
! start cycle 1
! *****

nselect,s,node,,ntop
f,all,fy,-f ! define load for load step 2
nselect,all
nsubst,4,25,2 ! set the number of substeps, max and min number
! of substeps
solve ! solve load step 2

nselect,s,node,,ntop
f,all,fy,f ! define load for load step 3
nselect,all
nsubst,4,25,2 ! set the number of substeps, max and min number
! of substeps
solve ! solve load step 3

! *****
! start cycle 2
! *****

nselect,s,node,,ntop
f,all,fy,-f ! define load for load step 4
nselect,all
nsubst,4,25,2 ! set the number of substeps, max and min number
! of substeps
solve ! solve load step 4

nselect,s,node,,ntop
f,all,fy,f ! define load for load step 5
nselect,all
nsubst,4,25,2 ! set the number of substeps, max and min number
! of substeps
solve ! solve load step 5

! *****

```

```

!          start cycle 3
! *****

nset,s,node,,ntop
f,all,fy,-f          ! define load for load step 6
nset,all
nsubst,4,25,2       ! set the number of substeps, max and min number
                    ! of substeps
solve                ! solve load step 6
nset,s,node,,ntop
f,all,fy,f          ! define load for load step 7
nset,all
nsubst,4,25,2       ! set the number of substeps, max and min number
                    ! of substeps.
solve                ! solve load step 7
save
fini

/post1
set,last            ! read in the results from the last substep of
                    ! the last step.

pldi,2             ! plot the deformed mesh with the undeformed
                    ! edge only
ples,nl,eqeq       ! plot the total accumulated equivalent
                    ! plastic strains

fini

/post26
eplo                ! plot the mesh
nset,s,node,,ntop  ! select the node where the point load is attached
esln                ! select the element attached to this node
elem=elnext(0)     ! get the number of this element
alls                ! select back everything in the model

! define variable 2 to be y component of stress at the node where the point
! load is applied

esol,2,elem,ntop,s,y,

! define variable 3 to be y component of elastic strain at the node where the
! point load is applied

esol,3,elem,ntop,epel,y,

! define variable 4 to be y component of plastic strain at the node where the
! point load is applied

esol,4,elem,ntop,eppl,y,

prvar,2,3,4
fini

```

## 8.15. Where to Find Other Examples

Several ANSYS, Inc. publications, particularly the *Mechanical APDL Verification Manual*, describe additional nonlinear analyses.

The *Mechanical APDL Verification Manual* consists of test case analyses demonstrating the analysis capabilities of the program. While these test cases demonstrate solutions to realistic analysis problems, the *Mechanical APDL Verification Manual* does not present them as step-by-step examples with lengthy data input instructions and printouts. However, most users who have at least limited finite element experience should be able to fill in the missing details by reviewing each test case's finite element model and input data with accompanying comments.

---

The *Mechanical APDL Verification Manual* includes a variety of nonlinear analysis test cases:

- VM7 - Plastic Compression of a Pipe Assembly
- VM11 - Residual Stress Problem
- VM24 - Plastic Hinge in a Rectangular Beam
- VM38 - Plastic Loading of a Thick-Walled Cylinder Under Pressure
- VM56 - Hyperelastic Thick Cylinder Under Internal Pressure
- VM78 - Transverse Shear Stresses in a Cantilever Beam
- VM80 - Plastic Response to a Suddenly Applied Constant Force
- VM104 - Liquid-Solid Phase Change
- VM124 - Discharge of Water from a Reservoir
- VM126 - Heat Transferred to a Flowing Fluid
- VM132 - Stress Relaxation of a Bolt Due to Creep
- VM133 - Motion of a Rod Due to Irradiation Induced Creep
- VM134 - Plastic Bending of a Clamped I-Beam
- VM146 - Bending of a Reinforced Concrete Beam
- VM185 - Current Carrying Ferromagnetic Conductor
- VM198 - Large Strain In-Plane Torsion Test
- VM199 - Viscoplastic Analysis of a Body Undergoing Shear Deformation
- VM200 - Viscoelastic Sandwich Seal Analysis
- VM218 - Hyperelastic Circular Plate
- VM220 - Eddy Current Loss in Thick Steel Plate



---

## Chapter 9: Linear Perturbation Analysis

---

In many engineering applications, the linear behavior of a structure based on a prior linear or nonlinear preloaded status is of interest. The linear perturbation analysis is designed to solve a linear problem from this preloaded stage. Typically, in the nonlinear analysis, the Newton-Raphson procedure is used (see [Nonlinear Structural Analysis \(p. 193\)](#)). The tangent matrix from the Newton-Raphson analysis can be used in the linear perturbation analysis in order to obtain the preloaded solution, since the linear stiffness matrix without preloading would not give an accurate solution.

Generally speaking, the linear perturbation analysis can be any analysis type. However, the program currently supports only linear perturbation static analyses, linear perturbation modal analyses, linear perturbation buckling analyses, and linear perturbation full harmonic analyses.

Most [current-technology elements](#) are supported in a linear perturbation analysis; see [Elements Under Linear Perturbation](#) in the *Element Reference*.

The following linear perturbation topics are available:

- [9.1. Understanding Linear Perturbation](#)
- [9.2. General Procedure for Linear Perturbation Analysis](#)
- [9.3. Considerations for Load Generation and Controls](#)
- [9.4. Considerations for Perturbed Stiffness Matrix Generation](#)
- [9.5. Considerations for Rotating Structures](#)
- [9.6. Example Inputs for Linear Perturbation Analysis](#)
- [9.7. Where to Find Other Examples](#)

### 9.1. Understanding Linear Perturbation

The linear perturbation analysis can be viewed as an iteration on top of a base (or prior) linear or nonlinear analysis. During the linear perturbation process, all of the linear or nonlinear effects from the base analysis are taken into account and are "frozen" so that the perturbation loads can generate structural results (such as deformation, stresses, and strains) linearly by using the "frozen" solution matrices and material properties. The linear or nonlinear effects from the base analysis are also carried over to the stress expansion pass, if applicable. However, for any downstream analysis, such as a linear dynamic analysis, only linear effects are accounted for.

If the linear or nonlinear effects from the base analysis are not of interest, there is no need to perform a linear perturbation analysis; a simple one-step linear or nonlinear analysis can serve that purpose.

Two key points are of fundamental importance for carrying out a linear perturbation analysis:

1. The total tangent stiffness matrix from the prior solution (the base analysis) must be obtained for the current linear perturbation analysis. This matrix is regenerated in the first phase of the linear perturbation procedure.
2. The total perturbation loads must be established. This load vector is calculated in the second phase of the linear perturbation procedure.

This chapter describes both points in detail. Input examples are given to aid in understanding the procedure.

## 9.2. General Procedure for Linear Perturbation Analysis

A linear perturbation analysis offers simplicity and ease of use. With the aid of a restart from the base analysis, it is easy to envision how the snapshot of the solution matrices from the base analysis is regenerated and used. All other control keys (commands) are optional.

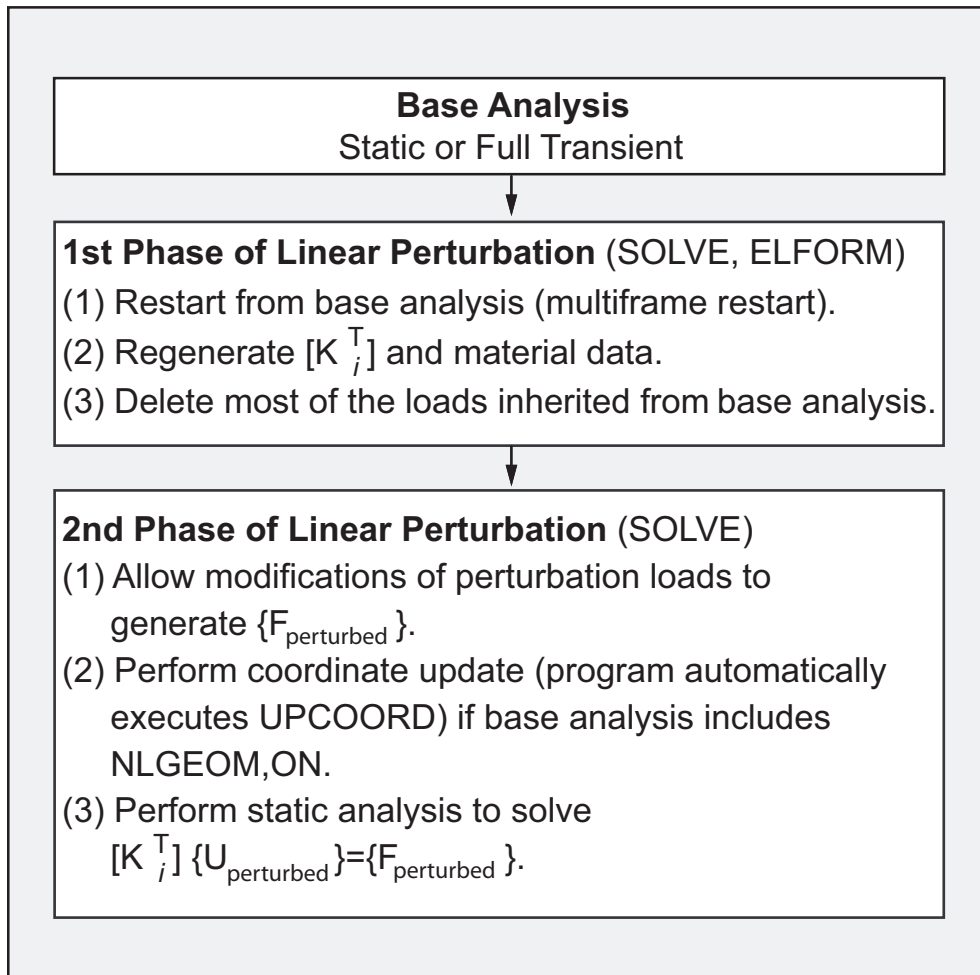
The following topics related to the linear perturbation analysis procedure are available:

- 9.2.1. Process Flow for a Linear Perturbation Analysis
- 9.2.2. The Base (Prior) Analysis
- 9.2.3. First Phase of the Linear Perturbation Analysis
- 9.2.4. Second Phase of the Linear Perturbation Analysis
- 9.2.5. Stress Calculations in a Linear Perturbation Analysis
- 9.2.6. Reviewing Results of a Linear Perturbation Analysis
- 9.2.7. Downstream Analysis Following the Linear Perturbation Analysis

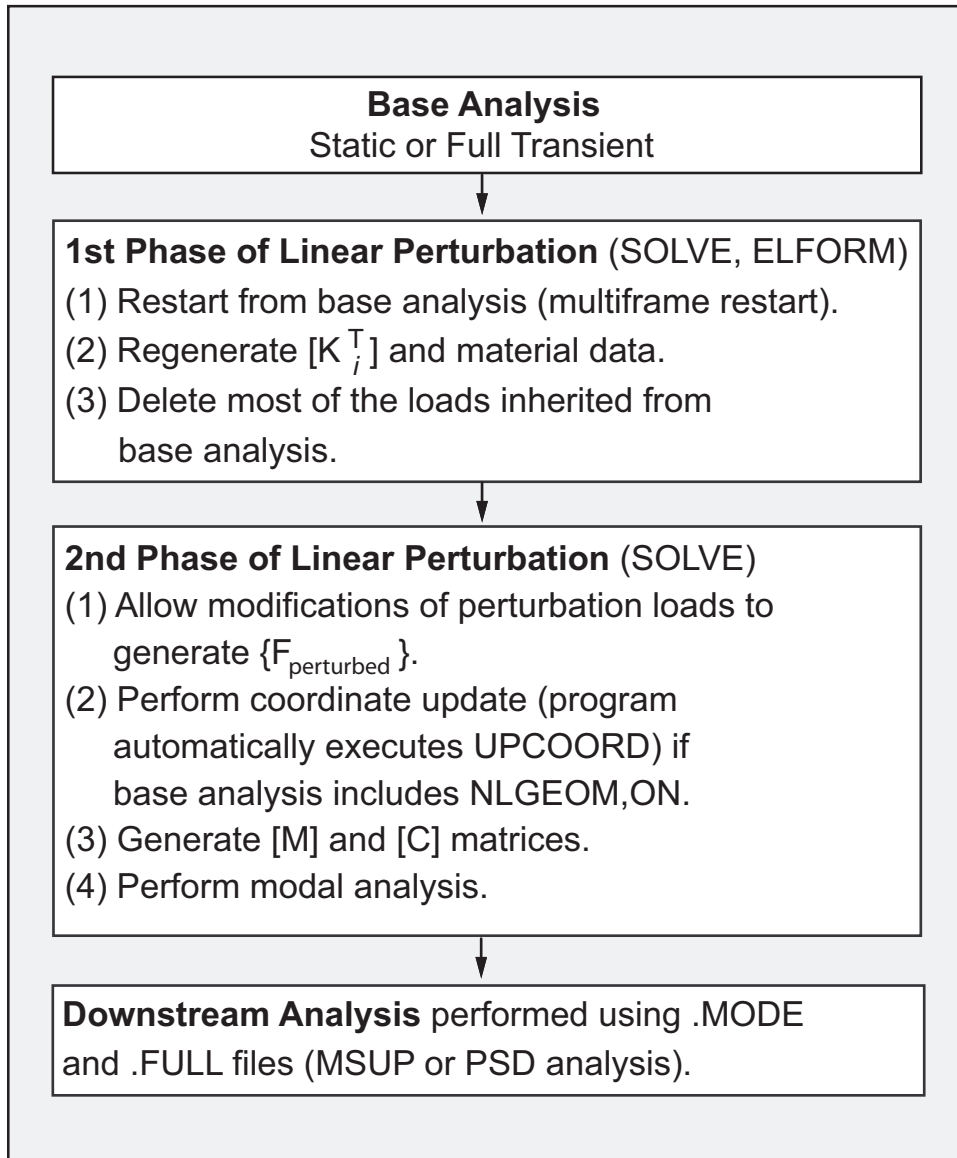
### 9.2.1. Process Flow for a Linear Perturbation Analysis

The following figures show the linear perturbation analysis flow for static, modal, eigenvalue buckling, and full harmonic analyses:

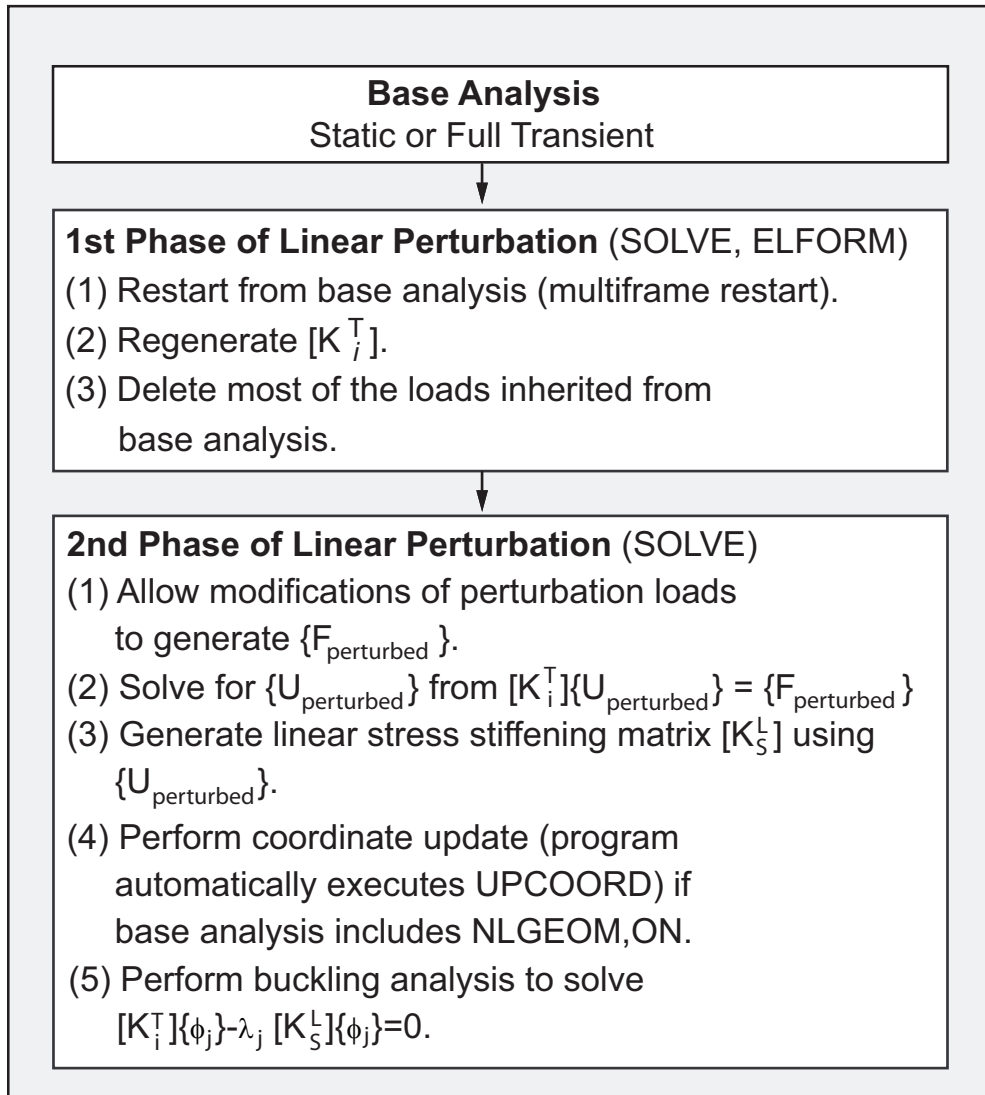
**Figure 9.1: Flowchart of Linear Perturbation Static Analysis**

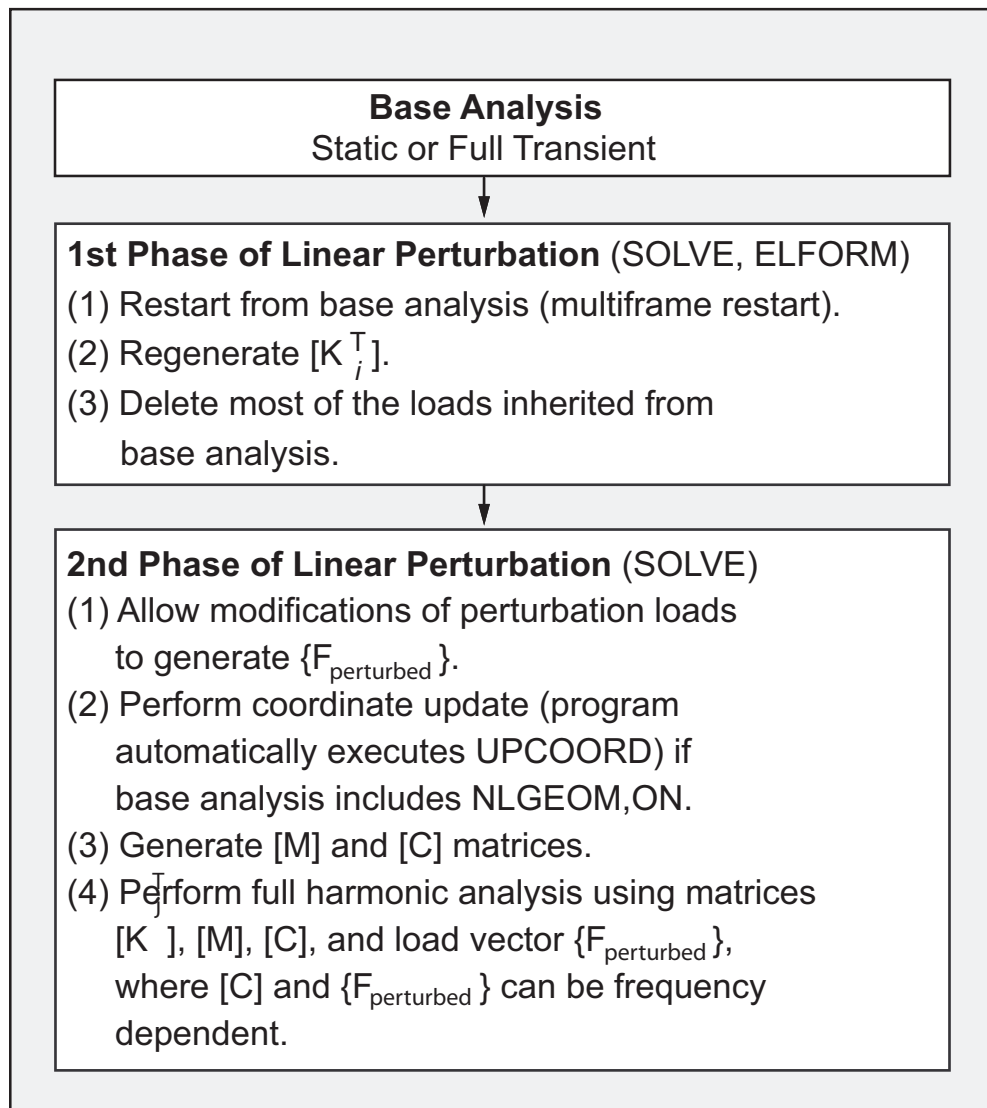




**Figure 9.2: Flowchart of Linear Perturbation Modal Analysis**

**Figure 9.3: Flowchart of Linear Perturbation Eigenvalue Buckling Analysis**



**Figure 9.4: Flowchart of Linear Perturbation Full Harmonic Analysis**

For advanced usage of the above analysis types, controls are available to affect how the snapshot matrices are modified to reflect the real-world engineering applications. These include nonlinear material controls, contact status controls, and loading controls.

### 9.2.2. The Base (Prior) Analysis

The base analysis (the analysis prior to the linear perturbation analysis) can be a linear or nonlinear, static or full (**TRNOPT**,**FULL**) transient analysis. The nonlinearity in the base analysis can be due to nonlinear materials, geometric nonlinearity, or nonlinear contact elements being used.

If the base analysis is linear static only (that is, only prestress effects are to be included in the model), the multiframe restart option must be invoked by using the **RESCONTROL**,**LINEAR** command, which is a non-default option for linear static analysis.

If the base analysis is a nonlinear static analysis or a nonlinear or linear full transient analysis, the multiframe restart is automatically invoked. However, you can use the **RESCONTROL** command to tell the

program at which time points the necessary data is to be saved for the multiframe restart. Only the converged substeps of the load step are saved for a multiframe restart, thereby automatically guaranteeing that a valid restarting point is used for the linear perturbation analysis.

The following files must be saved from the base analysis for use in the restart performed as the first phase of the linear perturbation procedure:

Jobname.RDB - ANSYS database file  
 Jobname.LDHI - Load history file  
 Jobname.Rnnn - Element saved records (restart files)

For more information on these files and the multiframe restart procedure, see [Restarting an Analysis](#) in the *Basic Analysis Guide*

### 9.2.3. First Phase of the Linear Perturbation Analysis

The purpose for the first phase in the linear perturbation analysis procedure is to regenerate the solution snapshot from the base analysis. Normally, this phase only requires the following command input:

```
/SOLU
ANTYPE,,RESTART,loadstep,substep,PERTURB
... ! (other limited commands are allowed)
PERTURB,MODAL ! can be STATIC, MODAL, BUCKLE, or HARMONIC (full harmonic)
SOLVE,ELFORM
! (do not exit solution module yet; do not issue FINISH command)
```

Upon execution of the **SOLVE,ELFORM** command, the program restarts the base analysis and regenerates

$[K_i^T]$  material data needed for the subsequent perturbation analysis and other possible solution matrices. Then, by default, the program removes all external loads inherited from the base analysis, except for displacement boundary conditions, inertia loads, and all non-mechanical loads (including thermal loads).

Since this phase is strictly used to regenerate solution matrices from the base analysis, no other actions (commands) are needed in most cases. The following items can be modified, however, so that the final solution matrices used in the linear perturbation analysis can serve various purposes for the engineering analysis:

- Change contact status via the **PERTURB** or **CNKMOD** command.
- To perform a partial nonlinear prestressed modal analysis for brake squeal simulation, issue the **CMROTATE** command.
- Modify element real constants (**RMODIF**).
- Modify material properties (**TBMODIF** or **MP** command); for example, to change the contact friction coefficient.
- Material behavior can be controlled via the **PERTURB** command. (For more information, see [Material Properties of Structural Elements in Linear Perturbation](#) in the *Element Reference*.)
- To include Coriolis effects in the modal analysis when the base analysis is static, issue the **CORIOLIS** command.

Other commands that are allowed in this phase of the linear perturbation analysis are: **MP**, **EALIVE**, **EKILL**, and **ESTIF**.

It is important to understand which matrices are used in the linear perturbation analysis:

- If the base analysis is nonlinear, the consistent tangent matrix from the prior Newton-Raphson iterations is regenerated based on the material behavior specified by the **PERTURB** command and based on the current geometry configuration if large-deflection effects are included (**NLGEOM,ON**).
- If contact elements are present in the base analysis, the stiffness matrix includes the effects of contact based on the contact status (set via **PERTURB** or **CNKMOD**).
- If the base analysis is linear, the linear stiffness matrix plus the stress stiffening matrix is used (automatically included).
- The spin-softening or gyroscopic effect is also included in this matrix regeneration phase as long as this spin-softening or gyroscopic effect is included in the base analysis.

Other commands such as **PSTRES**, **EMATWRITE**, **OMEGA**, and **CMOMEGA** are not needed in this phase, as they are accounted for automatically.

## 9.2.4. Second Phase of the Linear Perturbation Analysis

The second phase of a linear perturbation analysis is performed immediately following the **SOLVE,ELFORM** command from the first phase without exiting the solution processor (**/SOLU**); this is in order to correctly retain the snapshot of the restart status from the base analysis. Also, for the case of geometric nonlinearity, the nodal coordinates are updated automatically based on the restart point (you do not need to issue the **UPCOORD** command).

The second phase of the linear perturbation analysis varies slightly depending on whether you are performing a static, modal, eigenvalue buckling, or full harmonic analysis. Details for each type of linear perturbation are discussed in the following sections.

- 9.2.4.1. Second Phase - Static Analysis
- 9.2.4.2. Second Phase - Modal Analysis
- 9.2.4.3. Second Phase - Eigenvalue Buckling Analysis
- 9.2.4.4. Second Phase - Harmonic Analysis

### 9.2.4.1. Second Phase - Static Analysis

As described in [Figure 9.1: Flowchart of Linear Perturbation Static Analysis \(p. 288\)](#), the second phase of a linear perturbation static analysis consists of the following actions:

1. Apply linear perturbation loads to generate  $\{F_{\text{perturbed}}\}$ . (Note that thermal loads can be applied in the second phase of a linear perturbation static analysis. See [Generating and Controlling Non-mechanical Loads \(p. 299\)](#) for more information.)
2. If the base analysis included **NLGEOM,ON**, update the nodal coordinates by using the total displacement from the base analysis (similar to the **UPCOORD** command, but executed automatically and internally in this phase). From this point on, the deformed mesh is used for calculating perturbation loads and for postprocessing results from the linear perturbation analysis.

3. Perform the linear perturbation static analysis to solve 
$$\begin{bmatrix} K_i^T \end{bmatrix} \{U_{\text{Perturbed}}\} = \{F_{\text{Perturbed}}\}.$$

User action is needed only for the steps (1) and (3) shown above. The program performs step (2) automatically (see [Static Analysis Based on Linear Perturbation](#) in the *Mechanical APDL Theory Reference*).

The following example shows typical command input to accomplish these steps.

### Example Input for Linear Perturbation Static Analysis

```
F,NODE,... ! Add linear perturbation loads
...
TIME,... ! End time
SOLVE
FINISH
```

In a linear perturbation static analysis, the strain/stress calculation is done within the time substeps as it is in a standard static analysis.

Although a linear perturbation static analysis is a linear static analysis, these differences from a linear static analysis should be noted:

- The linear perturbation static analysis treats the process as a one-step analysis. It internally enforces the command **NSUBST,1,1,1** and ignores user-specified inputs entered via the **NSUBST** or **DELTIM** commands.
- If the base analysis is nonlinear and the tangent matrix becomes nearly singular (for example, reaching the nonlinear buckling point) or indefinite (for example, a plastic hinge) at the restart point, then a message concerning negative or small pivots may be printed out from the linear perturbation static analysis. This typically does not happen in a purely linear static analysis if the model is properly constrained. When this occurs in a linear perturbation analysis, it is your responsibility to verify the correctness of the solution.

### 9.2.4.2. Second Phase - Modal Analysis

As described in [Figure 9.2: Flowchart of Linear Perturbation Modal Analysis \(p. 289\)](#), the second phase of a linear perturbation modal analysis consists of the following actions:

1. Apply linear perturbation loads to generate  $\{F_{\text{perturbed}}\}$ . (Note that thermal loads *can not* be applied in the second phase of a linear perturbation modal analysis. See [Generating and Controlling Non-mechanical Loads \(p. 299\)](#) for more information.)
2. If the base analysis included **NLGEOM,ON**, update the nodal coordinates by using the total displacement from the base analysis (similar to the **UPCOORD** command, but executed automatically and internally in this phase). From this point on, the deformed mesh is used for calculating perturbation loads and for postprocessing results from the linear perturbation analysis.
3. Regenerate other needed matrices such as mass and damping matrices (**[M]** and **[C]**).
4. Perform the linear perturbation modal analysis.

User action is needed only for steps (1) and (4) shown above. The program performs steps (2) and (3) automatically (see [Modal Analysis Based on Linear Perturbation](#) in the *Mechanical APDL Theory Reference*).

The following example shows typical command input to accomplish these steps.

### Example Input for Linear Perturbation Modal Analysis

```
MODOPT,eigensolver,number_of_modes,
... (include commands to add or remove linear perturbation loads)
MXPAND,number_of_modes,
SOLVE
FINISH
```

In a linear perturbation modal analysis, after the solution phase, a stress expansion pass is typically carried out. A stress expansion must be done along with the modal analysis in order to use the appropriate material property and to obtain the total sum of elastic strain/stress due to the linear perturbation analysis and the base analysis. A separate expansion pass (**EXPASS** command) is not allowed after the linear perturbation analysis.

### 9.2.4.3. Second Phase - Eigenvalue Buckling Analysis

As described in [Figure 9.3: Flowchart of Linear Perturbation Eigenvalue Buckling Analysis \(p. 290\)](#), the second phase of a linear perturbation eigenvalue buckling analysis consists of the following actions:

1. Apply linear perturbation loads to generate  $\{F_{\text{perturbed}}\}$ . (Note that thermal loads can be applied in the second phase of a linear perturbation eigenvalue buckling analysis. See [Generating and Controlling Non-mechanical Loads \(p. 299\)](#) for more information.)

2. Solve for  $\{U_{\text{perturbed}}\}$  from  $[K_i^T]\{U_{\text{perturbed}}\} = \{F_{\text{perturbed}}\}$ . This step is executed automatically and internally by the program.

3. Generate the linear stress stiffness matrix  $[K_s^L]$  using  $\{U_{\text{perturbed}}\}$ . This step is executed automatically and internally by the program.

4. If the base analysis included **NLGEOM,ON**, update the nodal coordinates by using the total displacement from the base analysis (similar to the **UPCOORD** command, but executed automatically and internally in this phase). From this point on, the deformed mesh is used for calculating perturbation loads and for postprocessing results from the linear perturbation analysis.

5. Perform the linear perturbation eigenvalue buckling analysis to solve  $[K_i^T]\{\phi_j\} - \lambda_j [K_s^L]\{\phi_j\} = 0$ , where  $\phi_j$  are the eigenvectors and  $\lambda_j$  are the eigenvalues.

In a linear perturbation eigenvalue buckling analysis, the perturbation load  $\{F_{\text{perturbed}}\}$  does not include external loads introduced by contact gaps or penetrations, even if these loads were applied in the prior static or transient load steps.

User action is needed only for steps (1) and (5) shown above. The program performs steps (2), (3), and (4) automatically (see [Eigenvalue Buckling Analysis Based on Linear Perturbation](#) in the *Mechanical APDL Theory Reference*).

The following example shows typical command input to accomplish these steps.

#### Example Input for Linear Perturbation Buckling Analysis

```
BUCOPT,eigsolver,number_of_modes,
. . . (include commands to add or remove linear perturbation loads)
MXPAND,number_of_modes,
SOLVE
FINISH
```

Generally, the Block Lanczos eigensolver (**BUCOPT,LANB**) performs well for perturbed eigenvalue buckling analyses. However, when the tangent stiffness matrix becomes indefinite, the Block Lanczos eigensolver could fail to produce an eigensolution due to the mathematical limitation of this solver (refer to [Eigenvalue Buckling Analysis Based on Linear Perturbation](#) for more information on when this

could occur.) In this case, it is recommended that you use the subspace eigensolver (**BUCOPT**,**SUBSP**) to achieve a successful solution.

In a linear perturbation buckling analysis, after the solution phase, a stress expansion pass is typically carried out. A stress expansion must be done along with the buckling analysis in order to use the appropriate material property and to obtain the total sum of elastic strain/stress due to the linear perturbation analysis and the base analysis. A separate expansion pass (**EXPASS** command) is not allowed after the linear perturbation analysis.

#### 9.2.4.4. Second Phase - Harmonic Analysis

As described in [Figure 9.4: Flowchart of Linear Perturbation Full Harmonic Analysis \(p. 291\)](#), the second phase of a linear perturbation full harmonic analysis consists of the following actions:

1. Apply linear perturbation loads to generate  $\{F_{\text{perturbed}}\}$ . (Note that thermal loads *can not* be applied in the second phase of a linear perturbation full harmonic analysis. See [Generating and Controlling Non-mechanical Loads \(p. 299\)](#) for more information.)
2. If the base analysis included **NLGEOM,ON**, update the nodal coordinates by using the total displacement from the base analysis (similar to the **UPCOORD** command, but executed automatically and internally in this phase). From this point on, the deformed mesh is used for calculating perturbation loads and for postprocessing results from the linear perturbation analysis.
3. Regenerate the needed matrices such as mass and damping matrices ( $[M]$  and  $[C]$ ).
4. Perform the linear perturbation full harmonic analysis.

User action is needed only for steps (1) and (4) shown above. The program performs steps (2) and (3) automatically (see [Full Harmonic Analysis Based on Linear Perturbation](#) in the *Mechanical APDL Theory Reference* for more information.)

#### Example Input for Linear Perturbation Harmonic Analysis

The following example shows typical command input to accomplish these steps.

```
HROPT,FULL
. . . (include commands to add or remove lilnear perturbation loads)
HARFRQ,beginning_frequency,end_frequency
NSUB,number_of-frequency_steps
SOLVE
FINISH
```

In a linear perturbation full harmonic analysis, the strain/stress calculation is done within the frequency substeps as it is in a standard full harmonic analysis.

#### 9.2.5. Stress Calculations in a Linear Perturbation Analysis

Just as in a standard linear analysis, once the solution eigenvectors or harmonic solution are available, the stresses or strains of the structure can be calculated.

In general, two choices are available for calculating incremental (perturbation) stresses, depending on **PERTURB** command settings and base material properties. By default, the program uses the linear portion of the nonlinear material constitutive law to recover stresses for all materials, except for hyperelasticity. For hyperelasticity, the material property is based on the tangent of the hyperelastic material's constitutive law at the restart point.



Because a linear perturbation analysis can be understood as an extra iteration of the base analysis, all history-dependent results of the base analysis are inherited in the results of the linear perturbation analysis. Therefore, any plastic strains, creep strains, swelling strains, and contact results from the base analysis are available in the results data of the linear perturbation analysis.

The total strain is the sum of *all* strains (for example, **PLNSOL**,**EPTO**). The nonlinear solution quantities such as equivalent stress, stress state ratio, and plastic state variable are also available. For detailed information about element results from a linear perturbation analysis, see [Interpretation of Structural Element Results After a Linear Perturbation Analysis](#) in the *Element Reference*.

For contact elements, the contact status or contact forces are frequently of interest. The program uses the same contact status in calculations of both the stiffness matrix for the linear perturbation analysis and contact results for the expansion pass.

### 9.2.6. Reviewing Results of a Linear Perturbation Analysis

In the linear perturbation analysis, the program calculates all requested results, including stresses and strains, and stores them in the `Jobname.RSTP` file. One reason for the creation of the `Jobname.RSTP` file is that all the existing restart files from the base analysis are preserved for possible future use. The same linear perturbation or restart analysis can be performed repeatedly from the same restarting point. The `Jobname.RSTP` file only stores the results of the linear perturbation analysis. Results of the base analysis are still stored in the `Jobname.RST` file.

The stresses at the restarting point of the base static or full transient analysis should be obtained from the `Jobname.RST` file of the base analysis.

---

#### Note

When postprocessing results obtained in a separate session, you must resume the database saved **after** the linear perturbation analysis solution is finished, because the node coordinates have been updated with the base analysis displacements during the solve.

---

#### Note

When postprocessing the base analysis results, you must resume the database that was saved **before** the linear perturbation analysis.

---

### 9.2.7. Downstream Analysis Following the Linear Perturbation Analysis

Following the linear perturbation analysis, other analysis types can be performed by using the information from the linear perturbation analysis.

If the linear perturbation analysis is a modal analysis, the following analysis types are possible by using the `Jobname.MODE` file generated by the perturbation and the database of the model:

- Harmonic or transient analysis using the mode superposition method (*Method*=**MSUP** on the **HROPT** or **TRNOPT** command) . Mode superposition analysis based on QRDAMP eigensolver is not supported.
- The eigenmode reuse procedure ([Restarting a Modal Analysis \(p. 69\)](#)) is also available for linear perturbation analysis.
- Response spectrum analysis

- Random vibration analysis

The deformed mesh due to the prior static or full transient analysis is used in the linear perturbation analysis and in the downstream analysis. As such, the database used for the downstream analysis should correspond to the deformed mesh (after the linear perturbation solve).

In all analyses listed above, the first load vector used is  $\{F_{\text{perturbed}}\}$ . If more loading cases are required, a new  $\{F_{\text{perturbed}}\}$  load vector must be generated ( **MODCONT**). The program assumes purely linear analyses and uses linear material properties, even for all nonlinear materials.

### 9.3. Considerations for Load Generation and Controls

Generally, structural loads can be divided into two categories: mechanical loads and non-mechanical loads. Non-mechanical loads relevant to this procedure include thermal, swelling, initial stresses/strains, and other initial conditions. In the base analysis, a combination of mechanical and non-mechanical loads can be freely applied without any restrictions; however, for the linear perturbation analysis, only mechanical loads (and thermal loads in the case of linear perturbation static analysis and linear perturbation eigenvalue buckling analysis) are allowed.

It is good practice to delete all loads from the base analysis in the first phase of linear perturbation. By default, the program deletes all loads from the restart step, except for displacement boundary conditions, inertia loads, and non-mechanical loads (see the *LoadControl* argument on the **PERTURB** command). Non-mechanical loads (including thermal loads) must remain unmodified so that a detailed nonlinear snapshot for various solution matrices and element history variables can be regenerated in the first phase of the linear perturbation procedure.

You can apply new mechanical loads in the *second* phase of the linear perturbation process. However, new non-mechanical loads are not allowed except in a few cases. See [Generating and Controlling Non-mechanical Loads](#) (p. 299) for more information.

---

#### Note

When performing a linear perturbation full harmonic analysis, the following loads can act as harmonically varying loads:

- Nonzero displacements specified in the base analysis (**D** command).
- MPC bonded contact with initial penetration, gap, or offset. The effect of initial penetration, gap, or offset can be excluded by setting `KEYOPT(9) = 1` on the contact elements in the base analysis.
- Pretension loads (**SLOAD** command) specified in the base analysis.
- External constraint equations (**CE** command) with nonzero constant terms specified in the base analysis.

---

In a linear perturbation modal analysis, the total loads obtained from the second phase of the analysis may be used in a downstream analysis following the linear perturbation modal analysis. If a downstream analysis is not needed, then this load generation step can be ignored.

### 9.3.1. Generating and Controlling Mechanical Loads

The total perturbed loads are calculated as follows:

$$\{F_{perturbed}\} = \{F_{end}\} + \{F_{add}\}$$

where:

$\{F_{end}\}$  = total loads at the *end* of the load step of the current restart point (load applications are read from the `.LDHI` file). By default, all loads of  $\{F_{end}\}$  are deleted except for displacement boundary conditions, inertia loads, and non-mechanical loads.

$\{F_{add}\}$  = additional (new) loads prescribed by the user in the second phase of the linear perturbation analysis (after the first **SOLVE**,ELFORM command is invoked). This additional loading is optional.

In the first phase of a linear perturbation analysis, the **ANTYPE**,,RESTART command resumes the `Jobname.RDB` database and read in the `Jobname.LDHI` file to establish the  $\{F_{end}\}$  load.

New load application (adding to  $\{F_{add}\}$ ) or load removal (changing  $\{F_{end}\}$ ) can occur only in the *second* phase of the linear perturbation analysis (after the first **SOLVE**,ELFORM command), allowing flexibility in controlling the final  $\{F_{perturbed}\}$  that will be used.

$\{F_{perturbed}\}$  is used differently in each linear perturbation analysis type:

- In a linear perturbation static analysis,  $\{F_{perturbed}\}$  is used to compute the static analysis solution.
- In a linear perturbation modal analysis,  $\{F_{perturbed}\}$  is calculated and stored in the `Jobname.FULL` and `Jobname.MODE` files for a subsequent mode superposition, PSD, or other type of modal-based linear dynamic analysis.
- In a linear perturbation eigenvalue buckling analysis,  $\{F_{perturbed}\}$  is used to calculate the linearly perturbed displacements; these displacements are used for generation of the linearly perturbed stress stiffening matrix and thus the load factor for eigenvalue buckling analysis. Note that this load can be totally independent of the load used in the base analysis.
- In a linear perturbation full harmonic analysis,  $\{F_{perturbed}\}$  is used in the frequency steps for the harmonic solution.  $\{F_{perturbed}\}$  can be frequency dependent and can use complex input.

### 9.3.2. Generating and Controlling Non-mechanical Loads

Non-mechanical loads (including thermal loads) must remain unmodified in the first phase of the linear perturbation analysis so that a detailed nonlinear snapshot for various solution matrices and element history variables can be regenerated; thus, total load contributions to  $\{F_{perturbed}\}$  include non-mechanical loads in the first phase of a linear perturbation analysis.

In the second phase of a linear perturbation analysis, you cannot change, add, or remove non-mechanical loads with these exceptions: thermal loads can be defined in the second phase of a linear perturbation static analysis or a linear perturbation eigenvalue buckling analysis by specifying a new temperature. In a linear perturbation static or buckling analysis, assuming that the base analysis is nonlinear, the reference temperature is the temperature from which the linear perturbation analysis is restarted (and not the reference temperature **[TREF]** from the base analysis). If the base analysis is linear, then the

reference temperature used in the linear perturbation is the **TREF** temperature from the base analysis. See [Loads, Initial Conditions, and Other Limitations in Linear Perturbation](#) in the *Element Reference* for more information.

In the stress expansion pass of a linear perturbation modal or buckling analysis, thermal effects contribute no strain in the total strain calculation; that is, the thermal strain is zero in this expansion pass.

## 9.4. Considerations for Perturbed Stiffness Matrix Generation

The first phase of a linear perturbation analysis depends heavily on the multiframe restart procedure; it also inherits the limitation of the multiframe restart (see [Multiframe Restart Limitations](#) in the *Basic Analysis Guide*). These limitations include the following:

- The `Jobname.Rnnn` file does not save the **EKILL** and **EALIVE** commands. If **EKILL** or **EALIVE** commands are required in the first phase of the linear perturbation analysis, you must reissue these commands before the **SOLVE,ELFORM** command is executed.
- The `Jobname.RDB` file saves only the database information available at the first substep of the first load step. If you input other information after the first load step and need that information for the restart, you must input this information in the first phase of the linear perturbation analysis. This situation often occurs when APDL parameters are used. You must use **PARSAV** to save the parameters during the initial run and use **PARRES** to restore them in the restart. The situation also occurs when you want to change element real constants values; reissue the **R** command in the first phase of the linear perturbation analysis.

This limitation also applies to all other commands which modify element material properties or real constants from the base analysis in the second or later load step(s).

- All loading and boundary conditions from the base analysis are stored in the `Jobname.LDHI` file; therefore, upon the first phase of a linear perturbation analysis, changing the boundary conditions is not allowed. At the end of the first phase (after the **SOLVE,ELFORM** operation is completed), the program automatically manipulates the boundary conditions according to the specification of the `LoadControl` key on the **PERTURB** command; however, any type of load manipulation is allowed in the second phase of the linear perturbation analysis so that the desired perturbation loads can be easily obtained.

Before execution of the **SOLVE,ELFORM** command, the settings for regeneration of prior solution matrices with additional or specific requirements should be made. These specific requirements can be met by using the following commands (these are optionally needed):

**CMROTATE** -- Used to specify the rotational velocity of an element component about a user-defined rotational axis. It is needed in [brake squeal analysis](#) in conjunction with the QR damped eigensolver (**MODOPT,QRDAMP**).

**QRDOPT** -- Used by QR damped eigensolver for reuse of symmetric eigenmodes from the Block Lanczos eigensolver (LANB).

**CNKMOD** -- Used for modifying contact status from the base analysis locally for each contact pair. The `ContKey` parameter on the **PERTURB** command can modify the contact status globally.

**MP** -- Used for modifying contact frictional coefficient, MU.

**TBMODIF** -- Used for modifying frictional coefficients only (similar to **MP** command usage).

**EALIVE, EKILL, ESTIF** -- These are used for reproducing the element birth/dead status at the point of the multiframe restart. They may not be used to change element status from the restarted point.

**CORIOLIS** -- Used for including Coriolis effects in the modal analysis when the base analysis is static.

## 9.5. Considerations for Rotating Structures

When modeling rotating machinery, there are special considerations for performing a linear perturbation modal analysis that includes Coriolis effects.

If the base analysis is a full transient analysis, it is recommended that you include Coriolis effects (**CORIOLIS** command) in the full transient analysis to achieve an accurate linear perturbation analysis. These Coriolis effects are automatically carried over to the linear perturbation analysis.

If the base analysis is a static analysis, then the linear perturbation procedure depends on whether the Coriolis effects are based on a rotating reference frame or a stationary reference frame formulation. The difference comes from the fact that in a rotating reference frame, the stiffness matrix is modified by the spin-softening effect.

### Stationary Reference Frame

1. Static solve: Issue the **OMEGA/CMOMEGA** commands only if the centrifugal force is needed in the prestress solution.
2. First linear perturbation solve: Issue the **CORIOLIS** command. Use *LoadControl* = INERKEEP on the **PERTURB** command to retain **OMEGA/CMOMEGA** loads if they were defined in the base analysis.
3. Second linear perturbation solve: Issue **OMEGA/CMOMEGA** commands to specify the rotational velocity if not done in the base analysis.

Example input to perform perturbed modal solve in a stationary reference frame:

```
/solution
antype, static ! Static analysis
nlgeom, on ! Non-linear analysis
rescontrol, define, all, 1 ! Define restart files
f,... ! Define loads (for example nodal force)
solve
finish

/solution
antype, static, restart, last, last, perturb ! Do a linear perturbation analysis
perturb, modal,,, nokeep ! Perturbation modal
coriolis, on,,, on ! Coriolis in a stationary reference frame
solve, elform ! First Perturbation solve (form the stiffness matrix)

modopt, qrdamp,... ! Choose the eigensolver (for example QRDAMP)
mxpand, ... ! Expand the solutions
omega,... ! Define the rotational velocity
solve ! Second Perturbation solve (solve the eigenproblem)
finish
```

### Rotating Reference Frame

1. Static solve: Issue the **OMEGA/CMOMEGA** commands.
2. First linear perturbation solve: Issue the **CORIOLIS** command to include the Coriolis effect, and use *LoadControl* = INERKEEP on the **PERTURB** command to retain **OMEGA/CMOMEGA** loads.
3. Second linear perturbation solve: No specific commands are needed.

Example input to perform perturbed modal solve in a rotating reference frame:

```

/solution
antype, static    ! Static analysis
nlgeom, on       ! Non-linear analysis
rescontrol, define, all, 1 ! Define restart files
f,...           ! Define loads (for example nodal force)
omega,...       ! Define the rotational velocity
solve
finish

/solution
antype, static, restart, last, last, perturb ! Do a linear perturbation analysis
perturb, modal,,, inerkeep ! Perturbation modal (keep the inertial loads)
coriolis, on     ! Coriolis in a rotating reference frame
solve, elform    ! First Perturbation solve (form the stiffness matrix)

modopt, damp,... ! Choose the eigensolver (for example DAMP)
mxpand, ...     ! Expand the solutions
solve          ! Second Perturbation solve (solve the eigenproblem)
finish
    
```

## 9.6. Example Inputs for Linear Perturbation Analysis

This section contains input listings for the following linear perturbation analysis examples:

- [Linear Perturbation Static Analysis](#)
- [Simple Example of a Linear Perturbation Modal Analysis](#)
- [Linear Perturbation Modal Analysis with Large Rotation Effects](#)
- [Contact Status Control in a Linear Perturbation Modal Analysis](#)
- [Prestressed QRDAMP Linear Perturbation Modal Analysis](#)
- [Using Linear Perturbation to Predict a Buckling Load](#)
- [Linear Perturbation \(Prestressed\) Harmonic Analysis](#)

### Example 9.1: Linear Perturbation Static Analysis

```

/prep7
et,1,solid185

mp,ex,1,11.5e8
mp,nuxy,1,.3
mp,dens,1,1000

! ***** Creating geometry *****
block,0,1,0,-0.1,-1,1
esize,0.2
type,1
vmesh,1

block,1,2,0,-0.1,-1,1
esize,0.2
type,1
vmesh,2
allsel,all

! ***** Creating Bonded Contact Pair *****
*set,tid,4
*set,cid,3
et,cid,contal75
et,tid,targe170

keyo,cid,12,5                ! Bonded Contact

vsel,s,volu,,1
allsel,below,volu
type,3
    
```

```

real,3
nset,r,loc,x,1
esurf
allsel

vsel,s,volu,,2
allsel,below,volu
type,4
real,3
nset,r,loc,x,1
esurf
allsel

! ***** Displacement Load in Base Static Solve *****
*dim,disp_x,table,2,1,1
disp_x(1,0)=1
disp_x(1,1)=-1e-03
disp_x(2,0)=2
disp_x(2,1)=-1.5e-03

! ***** Force Load in Linear Perturbation Static Solve *****
*dim,force_y,table,1,1,1
force_y(1,0)=3
force_y(1,1)=1e+02

nset,s,loc,x,0
d,all,all
allsel,all

nset,s,loc,x,2
nset,r,loc,y,0
cm,compl,node
nset,all
finish

/solution
antype,static
nlgeom,on ! Turn on non-linear effects
nsubst,10,100,10
autots,on
outres,all,all
rescontrol,define,all,all, ! write restart files
nset,s,loc,x,2
d,all,ux,%disp_x%
allsel,all,all
time,2
solve ! 2nd load step
save
fini

/solution
antype,static,restart,, ,perturb
perturb,static,,current,nokeep ! Do not retain any BC from base static solve
solve,elform

nset,s,loc,x,0
d,all,all,0 ! Redefine BC
nset,all

f,compl,fy,%force_y%
time,3
allsel,all,all
solve
fini

/post1
file,,rstp
set,last
nset,s,,200,264,2
prnsol,u,sum
prnsol,s

```

```
nset,all
prrfor
finish
```

## Example 9.2: Simple Example of a Linear Perturbation Modal Analysis

```
/prep7
mp,ex,1,10      ! Simple material
mp,nuxy,1,0.3
mp,dense,1,0.5
n,1,
n,2,1
n,3,1,1
n,4,0,1
n,5,2,0
n,6,2,1
n,7,3,0
n,8,3,1

et,1,182,,,1
e,1,2,3,4
e,2,5,6,3,
e,7,8,6,5

allsel,all

d,1,all
d,4,ux
f,8,fy,-2.0     ! Apply a lateral point load at node 8
dlist
finish

/solution
antype,static   ! Static analysis
nlgeom,on       ! Large deformation and large strain
rescontrol,,all,1 ! Get restart files for all substeps
nsub,5,5,5      ! Use fixed time steps
solve
finish

!! Begin modal linear perturbation analysis

/solu
/com  FIRST PHASE OF LINEAR PERTURBATION
antype,,restart,1,3,perturb ! Restart at 1st loadstep, 3rd substep,
                                ! from the base nonlinear static analysis
perturb,modal
solve,elform     ! Execute 1st phase of linear perturbation, recovering Kt of NLGEOM,on
!!
/com  SECOND PHASE OF LINEAR PERTURBATION
dlist,all       ! Displacement boundary conditions are not deleted
flist,all       ! Lateral point load at node 8 has been deleted internally at SOLVE,elform
f,8,fy,-4.0     ! Apply new load for downstream analysis,
f,8,fx,-2.0     ! (downstream analysis is not shown in this simple example)
                                ! This newly applied point load does not affect modal solution

flist
modopt,lanb,3   ! Solve for lowest 3 modes by using block Lanczos
mexpand,3,,,yes ! Expand mode the same time
outres,esol,all
solve           ! Execute 2nd phase of linear perturbation: modal analysis
dlist          ! Verify perturbation loads again
flist
fini

/post1
file,,rstp     ! Use *.rstp file to review results from linear perturbation
set,list       ! It should list 3 eigen-modes
set,last       ! List stresses of the 3rd mode
esel,s,elem,,1
etable,seqv,s,eqv
```



```

allsel,all
etable,seqv,s,eqv
pretab,seqv
finish

```

### Example 9.3: Linear Perturbation Modal Analysis with Large Rotation Effects

```

*afun,deg

/prep7
/triad,lbot
/view,1,1,0,0
/angl,1,-90
/graph,full
rotxy=0
rotyz=0
rotzx=-90

et,1,185

n,1,0,0,0          ! Define nodes
n,2,.5,0,0
n,3,.5,1,0
n,4,0,1,0
ngen,11,10,1,4,1,0,0,1
en,1,1,2,3,4,11,12,13,14      ! Define elements
egen,10,10,1
mp,ex,1,1.0e6          ! define material
mp,ey,1,1.0e7
mp,ez,1,1.0e8
mp,nuxy,1,1.0e-8
mp,nuxz,1,1.0e-8
mp,nuyz,1,1.0e-8
mp,gxy,1,1e5
mp,gxz,1,1e5
mp,gyz,1,1e5
mp,dens,1,0.3/386.4
local,11,0,0,0,0,rotxy,rotyz,rotzx
csys,0
/psymb,cs,1
emodif,all,esys,11

tref,0
tunif,0
nsel,s,node,,1,4
d,all,ux,0
d,all,uy,0
d,all,uz,0
allsel,all
finish

!! Rigid rotation of 90 degrees

/solu
antype,static
nlgeom,on          ! large deflection
nsubst,1,1,1
save,casel,db,,model
nstep=9
dstep=90/nstep
*do,istep,1,nstep
  theta=dstep*istep
  nsel,s,node,,3,4
  d,all,uz,-sin(theta)
  d,all,uy,cos(theta)-1
  allsel
  solve
*enddo
finish

!! Linear perturbed modal analysis

```

```
/solu

/com  First phase of linear perturbation

antype,static,restart,,,perturb      ! Restart from the last substep of the rigid rotation

perturb,modal                          ! Set linear perturbed modal analysis
solve,elform                            ! Execute 1st phase of linear perturbation, recovering Kt of NLGEOM,on

/com  Second phase of linear perturbation

outres,all,all
modopt,lanb,2                          ! Solve for first 2 modes by using block lanczos
mxpand,2,,,yes                          ! Expand mode at the same time
solve                                  ! Execute 2nd phase of linear perturbation: modal analysis
finish
/post1
file,,rstp                             ! Use *.rstp file to review results from linear perturbation

set,list                                ! List 2 eigenmodes

set,1,1                                ! List strains and stresses of element 1 in the first mode
esel,s,elem,,1

rsys,0                                  ! List results in global coordinates
presol,s
presol,epto

rsys,solu                               ! List results in element coordinates
presol,s
presol,epto

finish
```

## Example 9.4: Contact Status Control in a Linear Perturbation Modal Analysis

```
/filename,test1

!***** CASE 1 *****
/prep7
et,1,182
r,1,0.1

mp,ex,1,30e6
mp,dens,1,7.324
mp,prxy,,0.3

csys,1
pcirc,10,5,0,90
pcirc,10,15,40,50
allsel,all
esize,0.7

cyclic                                ! A cyclic symmetric static problem
mshkey,1
amesh,all
allsel,all

!***** Original contact definition *****
et,2,172
mat,2
mp,mu,2,0.0001
keyopt,2,4,2
keyopt,2,2,0                          ! Penalty function
keyopt,2,12,2                          ! No-Separation contact (sliding allowed)
keyopt,2,5,1

type,2
real,2
lsel,s,line,,7
```

```

lmesh,all
allsel,all

et,3,169
lsel,s,line,,1
type,3
real,2
esurf,,reverse
lmesh,all
allsel,all

!***** Boundary Condition *****
nset,s,loc,x,5
d,all,all,0
nset,all
allsel,all

nset,s,loc,x,15
esln,s
cm,load,elem
allsel,all

nset,s,loc,x,13,14
nset,r,loc,y,40
cm,compl,node
nset,all
nrotate,all
finish

/SOLU                ! Enter the base nonlinear static analysis
antype,static
rescontrol,define,all,1
outres,all,all
nlgeom,on
nsubs,20,20,20
kbc,0
nset,s,loc,x,15
f,all,fx,-3e4
allsel,all
solve                ! 1st loadstep

time,2.0
nsubs,20,20,20
kbc,0
f,compl,fy,10
allsel,all
solve                ! 2nd loadstep
finish

/post1
set,last
/cycexpand,,on
esel,s,type,,2

!***** Contact status after Base static analysis: sliding *****
presol,cont,stat
esel,all
allsel,all
/cycexpand,,off
finish

/solu
antype,static,restart,,perturb    ! Enter linear perturbed modal analysis
                                     ! by using last loadstep and substep.
perturb,modal,,current,allkeep    ! Keep all the loads as in Base static run.
solve,elform                       ! First phase of linear perturbation

modopt,lanb,6
mxpand,6,,yes
solve                               ! Second phase of linear perturbation
finish                               ! Completion of linear perturbation

```

```

/post1
file,,rstp
set,list
/cycexpand,,on
set,1,5                ! Use the 5th mode, Harmonic Index = 0 to see solution
esel,s,type,,2

/com, *****
/com, *                CASE 1                *
/com, * Using CURRENT as contact status key on PERTURB command *
/com, *****
presol,cont,stat      ! Expected contact status: sliding (same as Base static analysis)

esel,all
/cycexpand,,off
/delete,test1,rstp
finish
/clear,nostart
/filename,test1

!***** CASE 2 *****
/solu
antype,static,restart,1,10,perturb ! Restart from loadstep 1, substep 10 (uy=2)
perturb,modal,,sticking,allkeep    ! Modify the contact from sliding to sticking
solve,elform

modopt,lanb,6
mxpand,6,,yes
solve
finish

/post1
file,,rstp
set,list
/cycexpand,,on
set,1,5
esel,s,type,,2

/com, *****
/com, *                CASE 2                *
/com, * Use sticking as contact status key on PERTURB command *
/com, *****
presol,cont,stat      ! Expected contact status: sticking

esel,all
/cycexpand,,off
/delete,test1,rstp
finish
/clear,nostart
/filename,test1

!***** CASE 3 *****
/solu
antype,static,restart,1,10,perturb ! Restart from load step 1, substep 10 (uy=2)
cnkmod,2,12,3                ! Change from Keyopt(12)=2 to Keyopt(12)=3 (bonded)
perturb,modal,,current,allkeep    ! Retain the contact status same as in Base static analysis
solve,elform
modopt,lanb,6
mxpand,6,,yes
solve
finish

/post1
file,,rstp
set,list
/cycexpand,,on
set,1,5
esel,s,type,,2

/com, *****
/com, *                CASE 3                *
/com, * Use CNKMOD command to modify individual contact pairs *

```

```

/com, *****
presol,cont,stat          ! Expected contact status: Sticking

esel,all
/cycexpand,,off
finish

```

### Example 9.5: Prestressed QRDAMP Linear Perturbation MODAL Analysis

```

/filename,modal

pres1 = 500
k1 = 1.e+4
k2 = 1.e+2

/prep7
et,1,185,,2
mp,ex,1,3.0e5
mp,nuxy,1,.3
mp,dens,1,.05
mp,mu,1,.3          ! Make friction large

lp = 0.5
hp = 0.5

block,-lp,lp,-lp,lp, ,hp
block,-lp,lp,-lp,lp,hp,2*hp

esize,1.0
vmesh,1,2

!***** Allow sliding contact between blocks *****
et,2,174
et,3,170
keyopt,2,9,1
keyopt,2,10,1
r,3

type,2
real,3
vsel,s,,,1,,1
nsel,r,loc,z,hp
esurf

type,3
vsel,s,,,2,,1
nsel,r,loc,z,hp
esurf
alls

!***** Put springs on UX *****
et,4,14
keyopt,4,2,1
r,4,k1
et,5,14
keyopt,5,2,1
r,5,k2

lsp = -hp/2
n,1000,-lp, lp,lsp
n,1001,-lp,-lp,lsp
n,1002, lp, lp,lsp
n,1003, lp,-lp,lsp

type,4
real,4
e,1000,node(-lp,lp,0)
e,1001,node(-lp,-lp,0)
e,1002,node(lp,lp,0)

type,5

```

```

real,5
e,1003,node(lp,-lp,0)

!***** Put springs on UY *****
et,6,14
keyopt,6,2,2
r,6,k1
et,7,14
keyopt,7,2,2
r,7,k2

type,6
real,6
e,1000,node(-lp,lp,0)
e,1001,node(-lp,-lp,0)
e,1002,node(lp,lp,0)

type,7
real,7
e,1003,node(lp,-lp,0)

!***** Put springs on UZ *****
et,8,14
keyopt,8,2,3
r,8,k1
et,9,14
keyopt,9,2,3
r,9,k2

type,8
real,8
e,1000,node(-lp,lp,0)
e,1001,node(-lp,-lp,0)
e,1002,node(lp,lp,0)

type,9
real,9
e,1003,node(lp,-lp,0)
finish

/solution
antype,static
nset,,loc,z,lsp
d,all,all,0           ! Apply zero displacement on second node of springs
allsel
vsel,s,,,1,,1
nset,r,loc,x,-lp
d,all,ux,0           ! Apply zero UX displacement on constrained plate
vsel,s,,,2,,1
nset,r,loc,x,-lp
d,all,ux,.01         ! Force UX displacement on free plate
vsel,s,,,2,,1
nset,r,loc,y,-lp
d,all,uy,0           ! Apply zero UY displacement on free plate
allsel
nset,s,loc,z,2*hp
sf,all,pres,pres1   ! Apply pressure on free plate
solc,,nopl
nset,all
esel,,type,,2
cm,contact2,elem    ! Make a contact component
allsel
outres,all,last
nlgeom,on
nropt,unsym         ! Use unsymmetric option for frictional contact
                        ! so that QRDAMP or UNSYM eigensolver can be used

solve
finish

/solution
antype,,restart,,,perturbation
perturb,modal

```

```

grdopt,yes                ! Allow reuse of QRDAMP Lanczos modes later
cmrot,contact2,0.0,0.0,1.9 ! Apply rotational velocity along rotZ
solve,elform              ! 1st phase of perturbation

! Free blocks to see rigid body modes, if any.
! This is an example of modifying loads or B.C. in the 2nd
! phase of linear perturbation.
vsel,s,,1,2,,1
ddelete,all,all
allsel
modopt,qr damp,20,50,,on  ! Try QRDAMP eigensolver
!modopt,unsym,20,50      ! Alternatively use UNSYM eigensolver  j
      15-0.31718E-02  1183.7  j      1
solve                    ! 2nd phase of perturbation
finish

/post1
file,modal,rstp
*get,FREQ_IMAG1,mode,1,freq,imag
*get,FREQ_IMAG2,mode,5,freq,imag
*get,FREQ_IMAG3,mode,10,freq,imag
*get,FREQ_IMAG4,mode,15,freq,imag
*get,FREQ_IMAG5,mode,20,freq,imag
*stat,FREQ_IMAG1
*stat,FREQ_IMAG2
*stat,FREQ_IMAG3
*stat,FREQ_IMAG4
*stat,FREQ_IMAG5
finish

```

### Example 9.6: Using Linear Perturbation to Predict a Buckling Load

```

!***** CASE 1 *****
!***** Base analysis *****
/prep7                ! Build the model
et,1,188

sectype,1,beam,rect  ! Define beam section
secdata,0.2,0.4
mp,ex,1,1.0e8        ! Define material
mp,dens,1,1000
mp,nuxy,1,0.3

local,11,0,0,0,0,45,0,0 ! Create model and mesh it
csys,11
k,1,0,0,0
k,2,10,0,0
l,1,2
lesize,1,,10
lmesh,1
allsel,all

d,1,all,0            ! Fix end of beam
allsel,all
nrotate,all
finish

save,model1.db      ! Save model as model1.db for use in cases 2 and 3

/solu
antype,static
outres,all,all

f,2,fx,-1000.0      ! Buckling load prediction is independent of
                    ! this load level because base analysis is linear

rescontrol,linear,all,1 ! Needed to generate restart file
solve
finish

!***** Linear perturbation buckling analysis - first phase *****

```

```

/solu
antype,static,restart,,,perturb      ! Restart the linear base analysis
perturb,buckle,,,allkeep             ! This is a LP eigen-buckling analysis
solve,elform

!***** Linear perturbation buckling analysis - second phase *****
f,2,fx,-10.0
bucopt,lanb,2,,,range
mexpand,2,,,yes
solve
finish

/post1
file,,rstp
set,1,1
*get,loadfactor,active,0,set,freq

/com,*****
/com, Case 1: Linear base analysis, Eigenvalue buckling analysis
/com, Buckling load is: Fbuckling = 0 + Lambda *(Fperturbed)
/com,
/com, Expected load factor = 66.0258
/com, predicted Fbuckling = 0-66.0258*10 = -660.258
/com,
/com,*****
*stat,loadfactor

finish
/delete,,rstp
/clear,nostart

!***** CASE 2 *****
resume,model1,db

!***** Base analysis *****
/solu
antype,static
outres,all,all
nlgeom,on                          ! Base analysis is nonlinear
rescontrol,define,all,1
nsubs,2,10,2
time,1
f,2,fx,-100.0                       ! Buckling load prediction is dependent on
                                       ! this load level because base analysis is nonlinear

solve
finish

!***** Linear perturbation buckling analysis - first phase *****
/solu
antype,static,restart,,,perturb      ! Restart at the end of loadstep
perturb,buckle,,,allkeep
solve,elform

!***** Linear perturbation buckling analysis - second phase *****
outres,all,all
f,2,fx,300
bucopt,lanb,2,,,center              ! Use CENTER option (default), we don't know if load factor
                                       ! is negative or positive

mexpand,2,,,yes
solve
finish

/post1
file,,rstp
set,1,2
*get,loadfactor2,active,0,set,freq

/com,*****
/com, Case 2: Nonlinear base analysis, Eigenvalue buckling analysis
/com, Buckling load is: Fbuckling = Frestart + Lambda *(Fperturbed),

```



```

/com, The CENTER option is used
/com,
/com, Expected load factor = -1.868
/com, predicted Fbuckling = -100-1.868*300 = -660.4
/com,
/com, *****
*stat,loadfactor2
finish
/delete,,rstp
/clear,nostart

!***** CASE 3 *****
resume,model1,db

!***** Base analysis *****
/solu
antype,static
outres,all,all
nlgeom,on ! Base analysis is nonlinear
rescontrol,define,all,1
nsubs,10,10,10
time,1
f,2,fx,-400.0 ! Buckling load prediction is dependent on
! this load level because base analysis is nonlinear

solve
finish

!***** Linear perturbation buckling analysis - first phase *****
/solu
antype,static,restart,1,5,perturb ! Restart in the middle of loadstep
perturb,buckle,,allkeep
solve,elform

!***** Linear perturbation buckling analysis - second phase *****
outres,all,all
f,2,fx,-500
bucopt,lanb,2,,range ! Expected load factor is greater than zero; no need to use CENTER
mxpand,2,,yes
solve
finish

/post1
file,,rstp
set,1,1
*get,loadfactor3,active,0,set,freq

/com, *****
/com, Case 3: Nonlinear base analysis, Eigenvalue buckling analysis
/com, Buckling load is: Fbuckling = Frestart + Lamda *(Fperturbed)
/com, No CENTER Option is used and restarting from middle.
/com,
/com, Expected load factor = 0.9205
/com, predicted Fbuckling=-400*0.5-0.9205*500=-660.25
/com,
/com, *****
*stat,loadfactor3

finish

```

### Example 9.7: Linear Perturbation (Prestressed) Harmonic Analysis

```

/prep7 ! Build the model
et,1,188

sectype,1,beam,rect ! Define beam section
secdata,0.2,0.4
mp,ex,1,2.0e11 ! Define material
mp,dens,1,7800
mp,nuxy,1,0.3

```

```

k,1,0,0,0          ! Create model and mesh it
k,2,4,0,0
l,1,2
lesize,1,,4
lmesh,all
allsel,all

nset,s,loc,x,0
d,all,all,0        ! Fix end of beam
nset,all
finish

/solu
antype,static
nsub,10,10,10
time,1.0
nlgeom,on
nset,s,loc,x,4
f,all,fx,1e5
f,all,fy,-1e6
nset,all
rescontrol,define,all,1 ! Needed to generate restart file
cnvtol,f,,1e-3         ! Tighter tolerance than default for better accuracy
cnvtol,m,,1e-4
solve
finish              ! Completion of nonlinear base analysis

!! CASE 1

!! Linear perturbation modal analysis starts here

/solu
antype,static,restart,1,5,perturb ! First phase of linear perturbation
! Restart in the middle
perturb,modal,,,allkeep
solve,elform

nset,s,loc,x,4      ! Second phase of linear perturbation
ddelete,all,ux
nset,all
modopt,lanb,24      ! Compute all the modes possible
mexpand,24,,,yes
solve
finish

!! Linear perturbation ends here

!! MSUP harmonic analysis starts here, uses modes from linear perturbation
!! MSUP is a downstream analysis

/solu
antype,harmonic
hropt,msup,24
outres,all,all
harfrq,23.5,24.5
nsub,10,10,10
kbc,1
nset,s,loc,x,4
f,all,fy,-1e7,2000
nset,all
solve
finish

!! Expansion pass

/solu
expass,on
outres,all,all
numexp,all,,,yes
solve
finish

```

```

!! End of linear perturbation based MSUP harmonic

/post26
nl=node(4,0,0)
nsol,2,nl,u,y
rforce,3,1,f,y,fyl
prcplx,1

/com,*****
/com, Case 1 : Perturbed MSUP harmonic analysis (extracting all the
/com,          modes for best accuracy)
/com,*****
prvar,2,3
finish
/clear,nostart

!! CASE 2

!! Linear perturbation full harmonic analysis starts here

/solu                                ! First phase of linear perturbation
antype,static,restart,1,5,perturb    ! Restart in the middle
perturb,harmonic,,,allkeep
solve,elform

nselect,s,loc,x,4                    ! Second phase of linear perturbation
ddelete,all,ux
nselect,all
hropt,full
harfrq,23.5,24.5
nsubst,10,10,10
kbc,1
nselect,s,loc,x,4
f,all,fy,-1e7,2000
nselect,all
solve
finish

/post26
file,,rstp
nl=node(4,0,0)
nsol,2,nl,u,y
rforce,3,1,f,y,fyl
prcplx,1

/com,*****
/com, Case 2: Perturbed FULL harmonic analysis, repeating
/com,          the same restart point from base analysis
/com,*****
prvar,2,3
finish

```

## 9.7. Where to Find Other Examples

Several ANSYS, Inc. publications, particularly the *Mechanical APDL Verification Manual* and the *Technology Demonstration Guide*, describe additional linear perturbation analysis examples.

The *Mechanical APDL Verification Manual* consists of test case analyses demonstrating the analysis capabilities of the ANSYS family of products. While these test cases demonstrate solutions to realistic analysis problems, the *Mechanical APDL Verification Manual* does not present them as step-by-step examples with lengthy data input instructions and printouts. However, most users who have at least some finite element experience can infer the necessary details by reviewing each test case's finite element model and input data with accompanying comments.

The *Mechanical APDL Verification Manual* includes the following linear perturbation analysis test cases:

### **Linear Perturbation Modal Analysis**

- [VM53](#) - Vibration of a String Under Tension
- [VM54](#) - Vibration of a Rotating Cantilever Blade
- [VM55](#) - Vibration of a Stretched Circular Membrane
- [VM59](#) - Lateral Vibration of an Axially-loaded Bar

### **Linear Perturbation Eigenvalue Buckling Analysis**

- [VM127](#) - Buckling of a Bar with Hinged Ends (Line Elements)
- [VM128](#) - Buckling of a Bar with Hinged Ends (Area Elements)
- [VMR029-T4](#) - Lateral Torsional Buckling of an Elastic Cantilever Subjected to Transverse End Load

### **Linear Perturbation Full Harmonic Analysis**

- [VM76](#) - Harmonic Response of a Guitar String

The *Technology Demonstration Guide* presents a series of analysis problems from a variety of engineering disciplines. Each problem description provides information about the nature and physical characteristics of the problem, specific modeling techniques, material properties, boundary conditions and loading, analysis details, and solution controls. The guide includes the following example problems that use linear perturbation analysis methods:

- [Centrifugal Impeller Analysis Using Cyclic Symmetry and Linear Perturbation](#)
- [Brake Squeal Analysis](#)

---

## Chapter 10: Gasket Joints Simulation

---

Gasket joints are essential components in most structural assemblies. Gaskets as sealing components between structural components are usually very thin and made of various materials, such as steel, rubber and composites.

From a mechanics perspective, gaskets act to transfer force between components. The primary deformation of a gasket is usually confined to one direction, namely, through thickness. The stiffness contributions from membrane (in plane) and transverse shear are much smaller in general compared to the through thickness. The stiffness contribution therefore is assumed to be negligible, although the **TB** command provides options to account for transverse shear.

A typical example of a gasket joint is in engine assemblies. A thorough understanding of the gasket joint is critical in engine design and operation. This includes an understanding of the behavior of gasket joint components themselves in an engine operation, and the interaction of the gasket joint with other components.

**Interface elements** (`INTER $n$` ) are used to model gaskets. By default, these elements account for both gasket through-thickness and transverse shear stiffness. However, you can modify the transverse shear stiffness by using the transverse shear option of the gasket material data table. You can also exclude the transverse shear stiffness via an element key option (KEYOPT) setting. For more information, see the **TB** command documentation and the documentation for each interface element.

The following topics concerning gasket joint simulation are available:

- [10.1. Performing a Gasket Joint Analysis](#)
- [10.2. Finite Element Formulation](#)
- [10.3. Interface Elements](#)
- [10.4. Material Definition](#)
- [10.5. Meshing Interface Elements](#)
- [10.6. Solution Procedure and Result Output](#)
- [10.7. Reviewing the Results](#)
- [10.8. Sample Gasket Element Verification Analysis \(Command or Batch Method\)](#)

### 10.1. Performing a Gasket Joint Analysis

A gasket joint analysis involves the same *overall* steps that are involved in any ANSYS nonlinear analysis procedure. Most of these steps however warrant special considerations for a gasket joint analysis. Presented below are the overall steps with the special considerations noted, along with links to applicable sections where more detailed information is included on that topic.

1. **Build or import the model.** There are no special considerations for building or importing the model for a gasket joint analysis. You perform this step as you would in any typical ANSYS analysis. See [Building the Model](#) in the *Basic Analysis Guide*. For further details on building the model, see the *Modeling and Meshing Guide*.
2. **Define element type.** To properly simulate gasket joints, you must define structural element types and corresponding interface element types. See [Interface Elements \(p. 319\)](#) in this chapter for more

details on this topic, and in particular, see [Element Selection \(p. 319\)](#) for a table of corresponding structural and interface elements.

- 3. Define material.** Use **TB,GASKET** to define the gasket joint material. You can use **TB,GASKET** to define four types of data input: general parameters, transverse shear stiffness, compression (loading), and tension (unloading). You specify the type using *TBOPT*. You then input the sets of data using the **TBDATA** and **TBPT** commands, as applicable. You can also plot most of the gasket data types using the **TBPLOT** command. See [Material Definition \(p. 320\)](#) in this chapter for more details on this topic.
- 4. Mesh the model.** Use the **AMESH** or **VMESH** commands to mesh the structural element types, and use the **IMESH** command to mesh the gasket layer. Special restrictions apply to the **IMESH** command in terms of matching the source and target. Also, the order in which you execute these commands is critical. You can also mesh interface layers using the **VDRAG** command, and can generate interface elements directly using the **EGEN** command. Each of these commands involve special considerations for interface elements. See [Meshing Interface Elements \(p. 327\)](#) in this chapter for more details on this topic.
- 5. Solve.** There are special solving considerations when you perform a gasket joint analysis. These are primarily concerned with the gasket element stiffness loss, and the gasket element's use with contact elements. See [Solution Procedure and Result Output \(p. 331\)](#) in this chapter for more details on this topic.
- 6. Review Results.** You can print or plot any of four gasket output items: stresses (also pressure), total closure, total inelastic closure, and thermal closure, using the **PRESOL**, **PRNSOL**, **PLESOL**, **PLNSOL**, or **ESOL** commands. You can also use these items with the **\*GET** command in POST1. See [Reviewing the Results \(p. 333\)](#) in this chapter for more details on this topic.

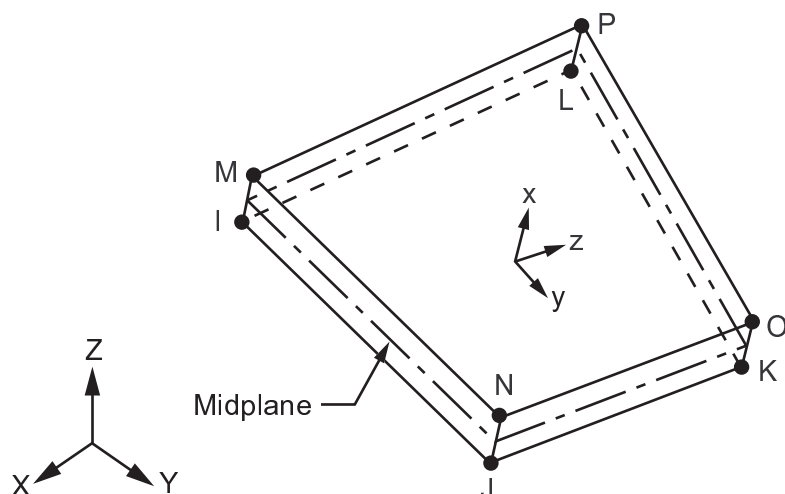
## 10.2. Finite Element Formulation

The primary deformation behavior of gasket joints is through-thickness deformation. It is therefore difficult to use solid continuum elements to effectively model gasket joints. The interface elements, which are based on the relative deformation of the top and bottom surfaces, offer a direct means to quantify through-thickness deformation of the gasket joints. Thus the pressure versus closure behavior can be directly applied to characterize the gasket material.

The element formulation is based on a corotational procedure. Refer to [Gasket Material](#) in the *Mechanical APDL Theory Reference* for further details.

### 10.2.1. Element Topologies

An interface element is composed of bottom and top surfaces. ANSYS provides several types of interface elements for the analysis of the gasket joints. [Figure 10.1: Element Topology of a 3-D 8-Node Interface Element \(p. 319\)](#) shows the geometry of a 3-D 8-node interface element available in ANSYS. An element midplane is created by averaging the coordinates of node pairs from the bottom and top surfaces of the elements. The numerical integration of interface elements is performed in the element midplane. The Gauss integration scheme is used for the numerical integration.

**Figure 10.1: Element Topology of a 3-D 8-Node Interface Element**

### 10.2.2. Thickness Direction

The thickness direction is defined as the normal direction of the mid plane of the element at the integration point, and computed inside of ANSYS. The positive direction is defined by the right-hand rule going around the nodes in the midplane. The through thickness deformation is quantified by the relative deformation of bottom and top surfaces along the thickness direction. The thickness direction is then noted as the X-direction according to the ANSYS notation convention. No **ESYS** coordinate system is allowed for the elements.

## 10.3. Interface Elements

Four types of elements are available to simulate gaskets. They are referred to as *interface elements* and are summarized as follows:

- **INTER192** - 2-D, 4-node, linear element.
- **INTER193** - 2-D, 6-node, quadratic element.
- **INTER194** - 3-D, 16-node, quadratic element.
- **INTER195** - 3-D, 8-node, linear element

The 2-D elements, **INTER192** and **INTER193**, use a KEYOPT to define various stress-state options.

### 10.3.1. Element Selection

The simulation of an entire gasket joint assembly, consisting of the gasket and the structural elements on either side of the gasket, involves choosing interface elements and structural elements that have the same characteristics. Use the following table as a guideline for choosing interface and structural elements that have the same characteristics:

For elements with these characteristics:	Use this interface element:	... with one of these structural elements:
2-D, linear	<b>INTER192</b>	<b>PLANE182</b>

For elements with these characteristics:	Use this interface element:	... with one of these structural elements:
2-D, quadratic	INTER193	PLANE183
3-D, quadratic	INTER194	SOLID96, SOLID186, SOLID187
3-D, linear	INTER195	SOLID65, SOLID185, SOLSH190, SOLID272, SOLID272, SOLID285

Proper element type is chosen based on the stress states of interest and structural element types used.

Element selection is done by the element type command, **ET**, for example,

```
ET,1,195
```

defines element type 1 as element INTER195.

### 10.3.2. Applications

In general, linear and quadratic elements are chosen for the following reasons:

- Fewer nodes produce a smaller model that runs faster with less computer resources.
- Quadratic elements are necessary if stress gradients are present in surrounding bodies.
- If elements are to follow a curved boundary closely, quadratic elements are ideal because their edges are arcs.
- With a free mesh (tetrahedral elements) the mid-node (quadratic) is required for an accurate solution.

When a surrounding structure can be considered as a 2-D structure, for example, plane stress / strain / axisymmetric, 2-D elements are the ideal choice. A good example of the use of 2-D element INTER192 or INTER193 is the gasket between the "flanged" ends of pipe line. In this case the gasket properties do not vary significantly with geometric location.

For a 3-D structure such as an internal combustion engine, 3-D element INTER194 or INTER195 is a good choice for simulating the gasket between the cylinder head and block. In this case there is no "nice" geometry because the gasket must fill in between two complicated surfaces, in between cylinders, and around other holes and passages. Also the gasket properties can vary in different zones. For example in a cylinder head, there is usually a much stiffer zone immediately around the cylinder to contain combustion pressure (called the "fire ring"). The remainder of the gasket is much softer.

## 10.4. Material Definition

### 10.4.1. Material Characteristics

The **TB** command option GASKET allows gasket joints to be simulated with the ANSYS interface elements, in which there is only one element through the thickness, and the through thickness deformation is decoupled from the in plane deformation. The gasket material is usually under compression. The material under compression is highly nonlinear. The gasket material also exhibits quite complicated unloading behavior when compression is released. The GASKET option allows you to directly input data for the experimentally measured complex pressure closure curve for the material model (compression curve), and also for several unloading pressure closure curves. When no unloading curves are defined, the material behavior follows the compression curve while it is unloaded.

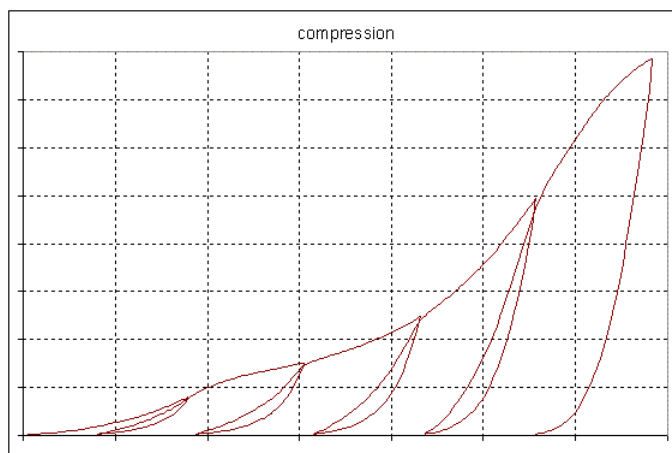


As it is a joint component, there often exists an initial gap or void. On the other hand, from a modeling point of view, it is a lot easier to fill the spaces or volumes between the adjacent components with the interface meshes, and then set an initial gap for the gasket material to account for it. As long as the closure is less than the initial gap, no pressure is acted on the gaskets. Also, when it is under tension loading, there will be an open gap. Therefore, gasket joints generally do not have tension pressure. A stress cap is used to restrict tension pressure in the gasket joint elements.

The GASKET material option must be used with interface elements [INTER192](#), [INTER193](#), [INTER194](#), and [INTER195](#).

[Figure 10.2: Pressure vs. Closure Behavior of a Gasket Material \(p. 321\)](#) shows the experimental pressure vs. closure (relative displacement of top and bottom gasket surfaces) data for a graphite composite gasket material. The sample was unloaded and reloaded 5 times along the loading path and then unloaded at the end of the test to determine the materials unloading stiffness.

**Figure 10.2: Pressure vs. Closure Behavior of a Gasket Material**



### 10.4.2. Input Format

You input gasket material data using **TB,GASKET**. The material data consists of 2 main parts: general parameters and pressure closure behaviors. The general parameters define initial gasket gap, stable stiffness for numerical stabilization, and stress cap for a gasket in tension. The pressure closure behavior includes gasket compression (loading) and tension data (unloading).

The **TB** command specification for defining a gasket material is:

**TB,GASKET,MAT,NTEMP,NPTS,TBOPT**

where *TBOPT* = one of the following types of gasket material data:

- PARA: gasket material general parameters.
- COMP: gasket compression data.
- LUNL: gasket linear unloading data.
- NUNL: gasket nonlinear unloading data.
- TSS: gasket transverse shear stiffness data.

You input the general parameters using the **TBDATA** command, then input the compression and unloading data using the **TBPT** command.

Presented in the following sections are examples of inputs for the various types of gasket data.

### 10.4.2.1. Define General Parameters

(*TBOPT* = PARA)

The gasket material general parameters include the initial gap, stable stiffness and the maximum tension stress cap. These parameters are defined as C1, C2, and C3 in the following example:

```
TB, GASKET, MAT, NTEMP, NPTS, PARA
TBDATA, 1, C1, C2, C3
```

Refer to [Gasket Materials](#) in the *Material Reference* for further details on these parameters.

### 10.4.2.2. Define Compression Load Closure Curve

(*TBOPT* = COMP)

The compression pressure closure curve gasket material definition option is defined as follows:

```
TB, GASKET, MAT, NTEMP, NPTS, COMP
TBPT, , x1, y1
TBPT, , x2, y2

TBPT, , xi, yi
```

where: xi, yi are pairs of closure and pressure values.

The following input listing is an example defining a compressive pressure vs. closure behavior of a gasket joint material with 10 data points.

```
TB, GASKET, 1, , 10, COMP           ! define compression data
TBPT, , 0.20000E-04, 0.54000E+08
TBPT, , 0.40000E-04, 0.15150E+09
TBPT, , 0.60000E-04, 0.24900E+09
TBPT, , 0.78000E-04, 0.30000E+09
TBPT, , 0.12000E-03, 0.37200E+09
TBPT, , 0.19000E-03, 0.47400E+09
TBPT, , 0.28600E-03, 0.58500E+09
TBPT, , 0.35800E-03, 0.67350E+09
TBPT, , 0.43200E-03, 0.78000E+09
TBPT, , 0.50500E-03, 0.89550E+09
```

### 10.4.2.3. Define Linear Unloading Data

(*TBOPT* = LUNL)

The linear unloading gasket material definition option is a simple way to define the gasket unloading behavior. Several unloading slopes can be defined to accommodate the comprehensive unloading behavior as follows:

```
TB, GASKET, MAT, NTEMP, NPTS, LUNL
TBPT, , x1, y1
TBPT, , x2, y2

TBPT, , xi, yi
```

where: *NPTS* is the number of unloading points; xi is the closure where unloading started, and yi is unloading slope.

The following input listing is an example showing the linear unloading behavior of a gasket joint material with 3 unloading points

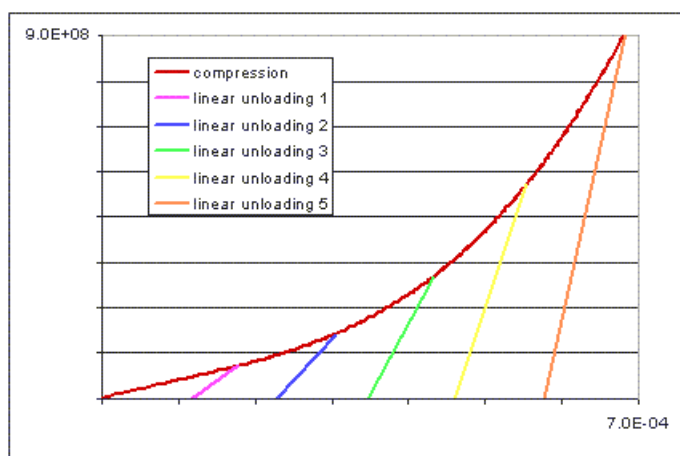
```

TB,GASKET,1, ,3,LUNL      ! define linear unloading data
TBPT,, 0.78000E-04, 0.25100E+12
TBPT,, 0.28600E-03, 0.25500E+12
TBPT,, 0.50500E-03, 0.10600E+13

```

A sample plot representing linear unloading curves is shown in [Figure 10.3: Gasket Material Input: Linear Unloading Curves \(p. 323\)](#).

**Figure 10.3: Gasket Material Input: Linear Unloading Curves**



#### 10.4.2.4. Define Nonlinear Unloading Data

(*TBOPT* = NUNL)

The nonlinear unloading gasket material definition option provides a more comprehensive way of defining gasket material unloading behavior. The input listing format is:

```

TB,GASKET,MAT,NTEMP,NPTS,NUNL
TBPT,,x1,y1
TBPT,,x2,y2

TBPT,,xi,yi

```

where:  $x_i, y_i$  are pairs of closure and pressure values. Several unloading curves can be defined.

An example showing the nonlinear unloading behavior of a gasket joint material with 3 unloading points is as follows:

```

TB,GASKET,1, ,5,NUNL      ! define first nonlinear unloading data
TBPT,, 0.78000E-04, 0.30000E+09
TBPT,, 0.66900E-04, 0.24750E+08
TBPT,, 0.63100E-04, 0.82500E+07
TBPT,, 0.54100E-04, 0.15000E+07
TBPT,, 0.00000E+00, 0.00000E+00

TB,GASKET,1, ,5,NUNL      ! define second nonlinear unloading data
TBPT,, 0.28600E-03, 0.58500E+09
TBPT,, 0.26400E-03, 0.22350E+08
TBPT,, 0.26100E-03, 0.90000E+07
TBPT,, 0.25600E-03, 0.15000E+07
TBPT,, 0.00000E+00, 0.00000E+00

TB,GASKET,1, ,5,NUNL      ! define third nonlinear unloading data
TBPT,, 0.50500E-03, 0.89550E+09
TBPT,, 0.47800E-03, 0.33900E+08

```

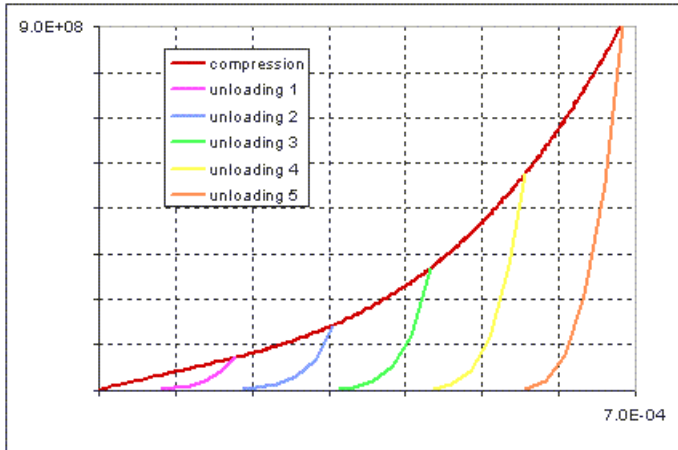
```

TBPT,, 0.47500E-03, 0.13500E+08
TBPT,, 0.46800E-03, 0.15000E+07
TBPT,, 0.00000E+00, 0.00000E+00

```

A sample plot representing nonlinear unloading curves is shown in [Figure 10.4: Gasket Material Input: Nonlinear Unloading Curves \(p. 324\)](#).

**Figure 10.4: Gasket Material Input: Nonlinear Unloading Curves**



### 10.4.3. Temperature Dependencies

Inputting temperature dependent gasket material properties follows the standard ANSYS procedure for inputting temperature dependent data for other materials. The following format shows this procedure.

```

TB, GASKET, MAT, NTEMP, NPTS, LUNL
TBTEMP, T1
TBPT, , x1, y1
TBPT, , x2, y2

TBTEMP, T2
TBPT, , x1, y1
TBPT, , x2, y2

```

ANSYS will automatically interpolate the temperature data to the material points using linear interpolation. When the temperature is out of the specified range, the closest temperature point is used.

The following is an example input listing defining a compressive pressure vs. closure behavior of a gasket joint material with 5 temperature points and up to 10 data points for each temperature point, and 3 nonlinear unloading curves with each curve having 5 temperatures and 5 data points.

```

TB, GASKET, 1, 5, 10, COMP      ! define compression data with 5 temperatures
TBTEMP, 100.000
TBPT, , 0.20000E-04, 0.54000E+08
TBPT, , 0.40000E-04, 0.15150E+09
TBPT, , 0.60000E-04, 0.24900E+09
TBPT, , 0.78000E-04, 0.30000E+09
TBPT, , 0.12000E-03, 0.37200E+09
TBPT, , 0.19000E-03, 0.47400E+09
TBPT, , 0.28600E-03, 0.58500E+09
TBPT, , 0.35800E-03, 0.67350E+09
TBPT, , 0.43200E-03, 0.78000E+09
TBPT, , 0.50500E-03, 0.89550E+09

TBTEMP, 200.000
TBPT, , 0.20000E-04, 0.18000E+08
TBPT, , 0.40000E-04, 0.50500E+08
TBPT, , 0.60000E-04, 0.83000E+08
TBPT, , 0.78000E-04, 0.10000E+09

```

```
TBPT,, 0.12000E-03, 0.12400E+09
TBPT,, 0.19000E-03, 0.15800E+09
TBPT,, 0.28600E-03, 0.19500E+09
TBPT,, 0.35800E-03, 0.22450E+09
TBPT,, 0.43200E-03, 0.26000E+09
TBPT,, 0.50500E-03, 0.29850E+09
```

```
TBTEMP, 300.000
```

```
TBPT,, 0.20000E-04, 0.90000E+07
TBPT,, 0.40000E-04, 0.25250E+08
TBPT,, 0.60000E-04, 0.41500E+08
TBPT,, 0.78000E-04, 0.50000E+08
TBPT,, 0.12000E-03, 0.62000E+08
TBPT,, 0.19000E-03, 0.79000E+08
TBPT,, 0.28600E-03, 0.97500E+08
TBPT,, 0.35800E-03, 0.11225E+09
TBPT,, 0.43200E-03, 0.13000E+09
TBPT,, 0.50500E-03, 0.14925E+09
```

```
TBTEMP, 400.000
```

```
TBPT,, 0.20000E-04, 0.36000E+07
TBPT,, 0.40000E-04, 0.10100E+08
TBPT,, 0.60000E-04, 0.16600E+08
TBPT,, 0.78000E-04, 0.20000E+08
TBPT,, 0.12000E-03, 0.24800E+08
TBPT,, 0.19000E-03, 0.31600E+08
TBPT,, 0.28600E-03, 0.39000E+08
TBPT,, 0.35800E-03, 0.44900E+08
TBPT,, 0.43200E-03, 0.52000E+08
TBPT,, 0.50500E-03, 0.59700E+08
```

```
TBTEMP, 500.000
```

```
TBPT,, 0.20000E-04, 0.18000E+07
TBPT,, 0.40000E-04, 0.50500E+07
TBPT,, 0.60000E-04, 0.83000E+07
TBPT,, 0.78000E-04, 0.10000E+08
TBPT,, 0.12000E-03, 0.12400E+08
TBPT,, 0.19000E-03, 0.15800E+08
TBPT,, 0.28600E-03, 0.19500E+08
TBPT,, 0.35800E-03, 0.22450E+08
TBPT,, 0.43200E-03, 0.26000E+08
TBPT,, 0.50500E-03, 0.29850E+08
```

```
TB,GASKET,1, 5,5,NUNL ! define first nonlinear unloading data with 5 temperatures
```

```
TBTEMP, 100.000
```

```
TBPT,, 0.78000E-04, 0.30000E+09
TBPT,, 0.66900E-04, 0.24750E+08
TBPT,, 0.63100E-04, 0.82500E+07
TBPT,, 0.54100E-04, 0.15000E+07
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 200.000
```

```
TBPT,, 0.78000E-04, 0.10000E+09
TBPT,, 0.66900E-04, 0.82500E+07
TBPT,, 0.63100E-04, 0.27500E+07
TBPT,, 0.54100E-04, 0.50000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 300.000
```

```
TBPT,, 0.78000E-04, 0.50000E+08
TBPT,, 0.66900E-04, 0.41250E+07
TBPT,, 0.63100E-04, 0.13750E+07
TBPT,, 0.54100E-04, 0.25000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 400.000
```

```
TBPT,, 0.78000E-04, 0.20000E+08
TBPT,, 0.66900E-04, 0.16500E+07
TBPT,, 0.63100E-04, 0.55000E+06
TBPT,, 0.54100E-04, 0.10000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

# Gasket Joints Simulation

---

```
TBTEMP, 500.000
TBPT,, 0.78000E-04, 0.10000E+08
TBPT,, 0.66900E-04, 0.82500E+06
TBPT,, 0.63100E-04, 0.27500E+06
TBPT,, 0.54100E-04, 0.50000E+05
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TB,GASKET,1, 5,5,NUNL ! define second nonlinear unloading data with 5 temperatures
```

```
TBTEMP, 100.000
TBPT,, 0.28600E-03, 0.58500E+09
TBPT,, 0.26400E-03, 0.22350E+08
TBPT,, 0.26100E-03, 0.90000E+07
TBPT,, 0.25600E-03, 0.15000E+07
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 200.000
TBPT,, 0.28600E-03, 0.19500E+09
TBPT,, 0.26400E-03, 0.74500E+07
TBPT,, 0.26100E-03, 0.30000E+07
TBPT,, 0.25600E-03, 0.50000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 300.000
TBPT,, 0.28600E-03, 0.97500E+08
TBPT,, 0.26400E-03, 0.37250E+07
TBPT,, 0.26100E-03, 0.15000E+07
TBPT,, 0.25600E-03, 0.25000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 400.000
TBPT,, 0.28600E-03, 0.39000E+08
TBPT,, 0.26400E-03, 0.14900E+07
TBPT,, 0.26100E-03, 0.60000E+06
TBPT,, 0.25600E-03, 0.10000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 500.000
TBPT,, 0.28600E-03, 0.19500E+08
TBPT,, 0.26400E-03, 0.74500E+06
TBPT,, 0.26100E-03, 0.30000E+06
TBPT,, 0.25600E-03, 0.50000E+05
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TB,GASKET,1, 5,5,NUNL ! define third nonlinear unloading data with 5 temperatures
```

```
TBTEMP, 100.000
TBPT,, 0.50500E-03, 0.89550E+09
TBPT,, 0.47800E-03, 0.33900E+08
TBPT,, 0.47500E-03, 0.13500E+08
TBPT,, 0.46800E-03, 0.15000E+07
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 200.000
TBPT,, 0.50500E-03, 0.29850E+09
TBPT,, 0.47800E-03, 0.11300E+08
TBPT,, 0.47500E-03, 0.45000E+07
TBPT,, 0.46800E-03, 0.50000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 300.000
TBPT,, 0.50500E-03, 0.14925E+09
TBPT,, 0.47800E-03, 0.56500E+07
TBPT,, 0.47500E-03, 0.22500E+07
TBPT,, 0.46800E-03, 0.25000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

```
TBTEMP, 400.000
TBPT,, 0.50500E-03, 0.59700E+08
TBPT,, 0.47800E-03, 0.22600E+07
TBPT,, 0.47500E-03, 0.90000E+06
TBPT,, 0.46800E-03, 0.10000E+06
TBPT,, 0.00000E+00, 0.00000E+00
```

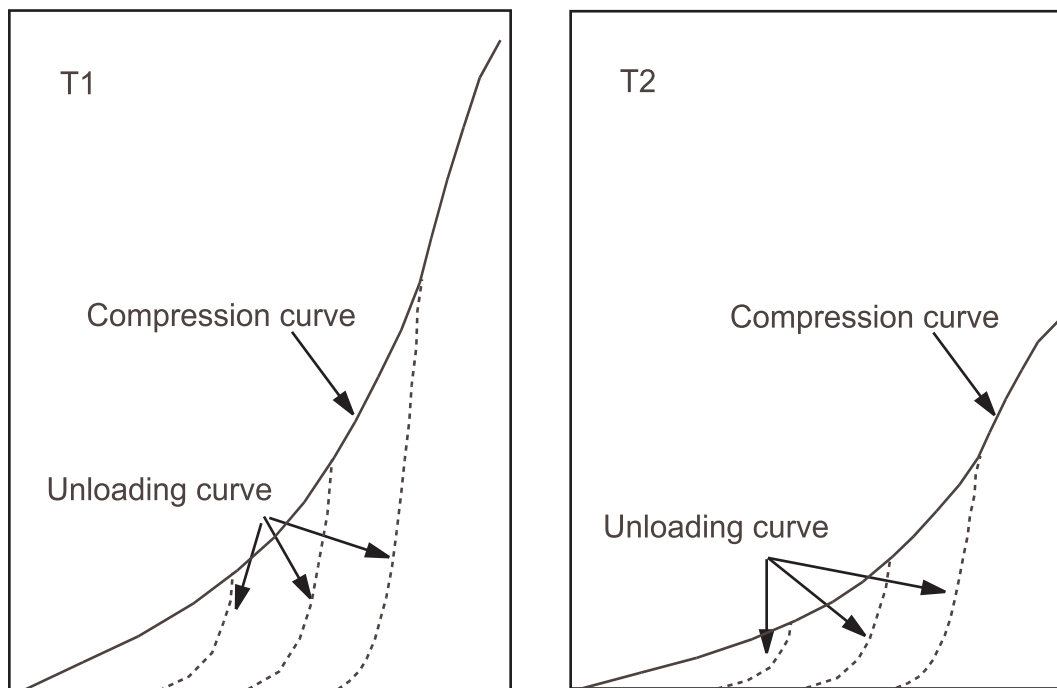
```

TBTEMP, 500.000
TBPT,, 0.50500E-03, 0.29850E+08
TBPT,, 0.47800E-03, 0.11300E+07
TBPT,, 0.47500E-03, 0.45000E+06
TBPT,, 0.46800E-03, 0.50000E+05
TBPT,, 0.00000E+00, 0.00000E+00

```

Sample plots of compression and unloading curves for gasket data to two temperatures is shown in Figure 10.5: Gasket Compression and Unloading Curves at Two Temperatures (p. 327).

**Figure 10.5: Gasket Compression and Unloading Curves at Two Temperatures**



#### 10.4.4. Plotting Gasket Data

You can plot gasket compression, linear unloading and nonlinear unloading data using the **TBPLOT** command. The use of this command to plot gasket data is as follows:

**TBPLOT**,GASKET,MAT,TBOPT,TEMP,SEGN

where *TBOPT* specifies the gasket material option to be plotted, *TEMP* specifies plotting either all of the temperature dependent data curves, or a curve at a specified temperature, and *SEGN* specifies whether or not to add the segment numbers to the curves.

### 10.5. Meshing Interface Elements

Three options are available for meshing interface elements:

- For meshing gasket layers as an area or volume, use the **IMESH** command.
- For meshing gasket layers by dragging an area mesh along a path, use the **VDRAG** command.
- For generating interface elements directly from a pattern, use the **EGEN** command.

There are special requirements for meshing interface elements. See [Generating an Interface Mesh for Gasket Simulations](#) in the *Modeling and Meshing Guide* for further details on this type of meshing.

The following example input listing shows the use of the **IMESH** command.

```

/batch,list
/title, Test to demonstrate the use of IMESH command
/com, *****
/com, *
/com, * This is a simple test to demonstrate the use of IMESH *
/com, * command to generate gasket elements. The model consists *
/com, * of two blocks with gasket elements (INTER194) defined *
/com, * between them. *
/com, * *
/com, *****

/prep7

!*+++++
!* Define Element Types
!*+++++
et,1,187          ! Solid tetrahedral element
et,2,194          ! Interface layer element

!*+++++
!* Define Parameters
!*+++++
EH=1.0
IH=0.1
DX=0
DY=0
DZ=IH
Z1=EH
Z2=Z1+IH
Z3=Z2+EH

!*+++++
!* Generate Keypoints
!*+++++
k,1,0,0
k,2,1,0
k,3,1,1
k,4,0,1
k,5,0,0,z1
k,6,1,0,z1
k,7,1,1,z1
k,8,0,1,z1
k,9,0,0,z2
k,10,1,0,z2
k,11,1,1,z2
k,12,0,1,z2
k,13,0,0,z3
k,14,1,0,z3
k,15,1,1,z3
k,16,0,1,z3

!*+++++
!* Generate First Volume
!*+++++
v,1,2,3,4,5,6,7,8

!*+++++
!* Generate Second Volume
!*+++++
v,9,10,11,12,13,14,15,16

!*+++++
!* Generate Middle Volume
!*+++++
v,5,6,7,8,9,10,11,12

```



```

!*+++++
!* Define Element Size
!*+++++
esize,,4      !

!*+++++
!* Mesh First Volume with Element Type 1
!*+++++
type,1
mat,1
vmesh,1

!*+++++
!* Generate Interface Layer with IMESH command
!* using Element Type 2 (INTER194)
!*+++++
type,2
mat,2
imesh,area,6,7,0,DX,DY,DZ,TOL

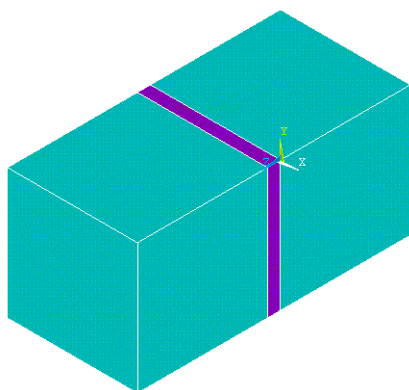
!*+++++
!* Mesh Second Volume with Element Type 1
!*+++++
type,1
mat,1
vmesh,2

!*+++++
!* Plot Elements
!*+++++
/view,1 ,1,1,1
eplot
finish

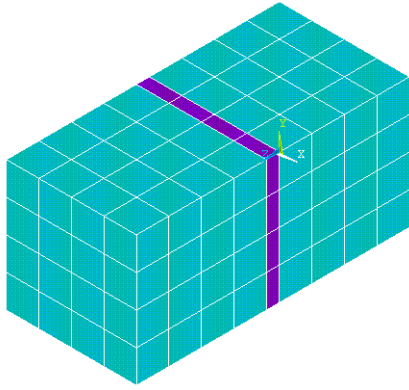
```

Figure 10.6: Gasket Finite Element Model Geometry (p. 329) shows the geometry of the finite element model, a thin interface layer between two block volumes. Figure 10.7: Whole Model Mesh with Brick Element (p. 330) shows the mesh with solid brick element, SOLID185, in top and bottom of block volumes, and Figure 10.8: Interface Layer Mesh (p. 330) shows the mesh of interface element, INTER195, in the interface layer between the two blocks. Figure 10.9: Whole Model Tetrahedral Mesh (p. 330) shows the mesh with solid tetrahedral element, SOLID187, in top and bottom of block volumes, and Figure 10.10: Interface Layer Mesh with Degenerated Wedge Elements (p. 331) shows the mesh of interface element (degenerated wedge), INTER194, in the interface layer between the two blocks.

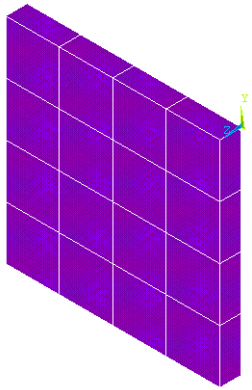
**Figure 10.6: Gasket Finite Element Model Geometry**



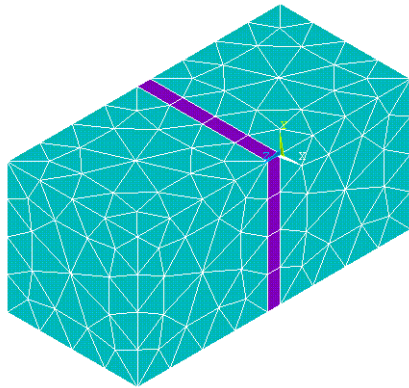
**Figure 10.7: Whole Model Mesh with Brick Element**

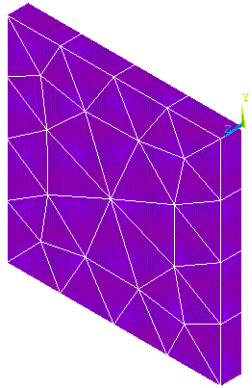


**Figure 10.8: Interface Layer Mesh**



**Figure 10.9: Whole Model Tetrahedral Mesh**



**Figure 10.10: Interface Layer Mesh with Degenerated Wedge Elements**

## 10.6. Solution Procedure and Result Output

Gasket material behavior is highly nonlinear. The full Newton-Raphson solution procedure (the standard ANSYS nonlinear method) is the default method for performing this type of analysis. Other solution procedures for gasket solutions are not recommended.

As with most nonlinear problems, convergence behavior of a gasket joint analysis depends on the problem type. ANSYS provides a comprehensive solution hierarchy; therefore, it is best to use the default solution options unless you are certain about the benefits of any changes.

Some special considerations for solving a gasket problem are as follows:

- By default, a zero stress cap is enforced on the gasket. When the element goes into tension, it loses its stiffness and sometimes causes numerical instability.
- It is always a good practice to place the lower and upper limit on the time step size (**DELTIM** or **NSUBST**). Start with a small time step, then subsequently ramp it up. This practice ensures that all modes and behaviors of interest are accurately included.
- When modeling gasket interfaces as *sliding contact*, it is usually necessary to include adequate gasket transverse shear stiffness. By default, the gasket elements account for a small transverse shear stiffness. You can modify the transverse shear stiffness if needed (**TB,GASKET,,,,TSS**) command. For better solution stability, use nodal contact detection.
- When modeling gasket interfaces via a *matching mesh method* (that is, with coincident nodes), it is better to *exclude* transverse shear stiffness to avoid unnecessary in-plane interaction between the gasket and mating components.

Like any other type of nonlinear analysis, the ANSYS program performs a series of linear approximations with corrections. A convergence failure can indicate a physical instability in the structure, or it can merely be the result of some numerical problem in the finite element model. The program printout gives you continuous feedback on the progress of these approximations and corrections. (The printout either appears directly on your screen, is captured on `Jobname. OUT`, or is written to some other file `[/OUTPUT]`.) You can examine some of this same information in POST1, using the **PRITER** command, or in POST26, using the **SOLU** and **PRVAR** commands. Understand the iteration history of your analysis before you accept the results. In particular, do not dismiss any program error or warning statements without fully understanding their meaning. A typical output listing with gasket nonlinearity only is shown in [Typical Gasket Solution Output Listing \(p. 332\)](#). When other types of nonlinearity such as contact or materials are included, additional information will be printed out.

## 10.6.1. Typical Gasket Solution Output Listing

### SOLUTION OPTIONS

PROBLEM DIMENSIONALITY. . . . .3-D  
 DEGREES OF FREEDOM. . . . . UX UY UZ  
 ANALYSIS TYPE . . . . .STATIC (STEADY-STATE)  
 PLASTIC MATERIAL PROPERTIES INCLUDED. . . . .YES  
 NEWTON-RAPHSON OPTION . . . . .PROGRAM CHOSEN

\*\*\* NOTE \*\*\* CP= 0.000 TIME= 00:00:00  
 Present time 0 is less than or equal to the previous time.  
 Time will default to 1.

\*\*\* NOTE \*\*\* CP= 0.000 TIME= 00:00:00

Nonlinear analysis, NROPT set to the FULL Newton-Raphson  
 solution procedure for ALL DOFs.

\*\*\* NOTE \*\*\* CP= 0.000 TIME= 00:00:00

The conditions for direct assembly have been met. No .emat or .erot  
 files will be produced.

### LOAD STEP OPTIONS

LOAD STEP NUMBER. . . . . 1  
 TIME AT END OF THE LOAD STEP. . . . . 1.0000  
 AUTOMATIC TIME STEPPING . . . . . ON  
 INITIAL NUMBER OF SUBSTEPS . . . . . 200  
 MAXIMUM NUMBER OF SUBSTEPS . . . . . 20000  
 MINIMUM NUMBER OF SUBSTEPS . . . . . 20  
 MAXIMUM NUMBER OF EQUILIBRIUM ITERATIONS. . . . . 15  
 STEP CHANGE BOUNDARY CONDITIONS . . . . . NO  
 TERMINATE ANALYSIS IF NOT CONVERGED . . . . .YES (EXIT)  
 CONVERGENCE CONTROLS. . . . .USE DEFAULTS  
 COPY INTEGRATION POINT VALUES TO NODE . . . . .YES, FOR ELEMENTS WITH  
 ACTIVE MAT. NONLINEARITIES  
 PRINT OUTPUT CONTROLS . . . . .NO PRINTOUT  
 DATABASE OUTPUT CONTROLS  
 ITEM FREQUENCY COMPONENT  
 ALL ALL  
 SVAR ALL

Range of element maximum matrix coefficients in global coordinates  
 Maximum= 4.326388889E+11 at element 0.  
 Minimum= 388758681 at element 0.

\*\*\* ELEMENT MATRIX FORMULATION TIMES  

TYPE	NUMBER	ENAME	TOTAL CP	AVE CP
1	2	SOLID185	0.000	0.000000
2	1	INTER195	0.000	0.000000

 Time at end of element matrix formulation CP= 0.

ALL CURRENT ANSYS DATA WRITTEN TO FILE NAME=  
 FOR POSSIBLE RESUME FROM THIS POINT  
 FORCE CONVERGENCE VALUE = 0.4200E+07 CRITERION= 0.2143E+05

SPARSE MATRIX DIRECT SOLVER.  
 Number of equations = 24, Maximum wavefront = 0  
 Memory available (MB) = 0.0, Memory required (MB) = 0.0

DISP CONVERGENCE VALUE = 0.1130E-04 CRITERION= 0.2000E-06  
 EQUIL ITER 1 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC= -0.4000E-05

FORCE CONVERGENCE VALUE = 0.1367E-08 CRITERION= 51.35 <<< CONVERGED  
 DISP CONVERGENCE VALUE = 0.3257E-20 CRITERION= 0.2000E-06 <<< CONVERGED  
 EQUIL ITER 2 COMPLETED. NEW TRIANG MATRIX. MAX DOF INC = -0.2234E-20  
 >>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 2

```

*** ELEMENT RESULT CALCULATION TIMES

TYPE  NUMBER  ENAME      TOTAL CP  AVE CP
     1      2  SOLID185      0.000    0.000000
     2      1  INTER195      0.000    0.000000

*** NODAL LOAD CALCULATION TIMES
TYPE  NUMBER  ENAME      TOTAL CP  AVE CP
     1      2  SOLID185      0.000    0.000000
     2      1  INTER195      0.000    0.000000

*** LOAD STEP      1  SUBSTEP      1  COMPLETED.    CUM ITER =      2
*** TIME = 0.500000E-02    TIME INC = 0.500000E-02
*** AUTO STEP TIME:  NEXT TIME INC = 0.500000E-02  UNCHANGED

FORCE CONVERGENCE VALUE = 0.7716E-09  CRITERION= 100.6
DISP CONVERGENCE VALUE  = 0.1951E-20  CRITERION= 0.2000E-06 <<< CONVERGED
EQUIL ITER 1 COMPLETED.  NEW TRIANG MATRIX.  MAX DOF INC= 0.1642E-20
FORCE CONVERGENCE VALUE = 0.4624E-09  CRITERION= 102.7 <<< CONVERGED
>>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 1

*** LOAD STEP      1  SUBSTEP      2  COMPLETED.    CUM ITER =      3
*** TIME = 0.100000E-01    TIME INC = 0.500000E-02
*** AUTO TIME STEP:  NEXT TIME INC = 0.750000E-02  INCREASED (FACTOR = 1.5000)

FORCE CONVERGENCE VALUE = 0.1964E-08  CRITERION= 176.1
DISP CONVERGENCE VALUE  = 0.3748E-20  CRITERION= 0.3000E-06 <<< CONVERGED
EQUIL ITER 1 COMPLETED.  NEW TRIANG MATRIX.  MAX DOF INC= -0.2744E-20
FORCE CONVERGENCE VALUE = 0.8307E-09  CRITERION= 179.7 <<< CONVERGED
>>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 1

*** LOAD STEP      1  SUBSTEP      3  COMPLETED.    CUM ITER =      4
*** TIME = 0.175000E-01    TIME INC = 0.750000E-02
*** AUTO TIME STEP:  NEXT TIME INC = 0.112500E-01  INCREASED (FACTOR = 1.5000)

FORCE CONVERGENCE VALUE = 0.3713E-08  CRITERION= 289.4
DISP CONVERGENCE VALUE  = 0.7198E-20  CRITERION= 0.4500E-06 <<< CONVERGED
EQUIL ITER 1 COMPLETED.  NEW TRIANG MATRIX.  MAX DOF INC= -0.4503E-20
FORCE CONVERGENCE VALUE = 0.1468E-08  CRITERION= 295.3 <<< CONVERGED
>>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 1

*** LOAD STEP      1  SUBSTEP      4  COMPLETED.    CUM ITER =      5
*** TIME = 0.287500E-01    TIME INC = 0.112500E-01
*** AUTO TIME STEP:  NEXT TIME INC = 0.168750E-01  INCREASED (FACTOR = 1.5000)

FORCE CONVERGENCE VALUE = 0.1833E-07  CRITERION= 459.2
DISP CONVERGENCE VALUE  = 0.3753E-19  CRITERION= 0.6750E-06 <<< CONVERGED
EQUIL ITER 1 COMPLETED.  NEW TRIANG MATRIX.  MAX DOF INC= 0.1800E-19
FORCE CONVERGENCE VALUE = 0.3550E-08  CRITERION= 468.6 <<< CONVERGED
>>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 1

*** LOAD STEP      1  SUBSTEP      5  COMPLETED.    CUM ITER =      6
*** TIME = 0.456250E-01    TIME INC = 0.168750E-01
*** AUTO TIME STEP:  NEXT TIME INC = 0.25313E-01  INCREASED (FACTOR = 1.5000)

FORCE CONVERGENCE VALUE = 0.2322E-07  CRITERION= 714.0
DISP CONVERGENCE VALUE  = 0.4656E-19  CRITERION= 0.1013E-05 <<< CONVERGED
EQUIL ITER 1 COMPLETED.  NEW TRIANG MATRIX.  MAX DOF INC= -0.2664E-19
FORCE CONVERGENCE VALUE = 0.6406E-08  CRITERION= 728.5 <<< CONVERGED
>>> SOLUTION CONVERGED AFTER EQUILIBRIUM ITERATION 1

*** LOAD STEP      1  SUBSTEP      6  COMPLETED.    CUM ITER =      7
*** TIME = 0.709375E-01    TIME INC = 0.253125E-01
*** AUTO TIME STEP:  NEXT TIME INC = 0.37969E-01  INCREASED (FACTOR = 1.5000)

```

## 10.7. Reviewing the Results

Results from a gasket joint analysis consist mainly of displacements, stresses, strains and reaction forces of the structural components and the gasket joint layer information (gasket pressure, closure, etc.). You can review these results in POST1 (the general postprocessor) or in POST26 (the time-history postpro-

cessor). See the Output Data sections of the element descriptions for any of the interface elements (for example [INTER192](#)) for a description of the available output components.

Note that in POST1, only one substep can be read in at a time, and that the results from that substep should have been written to `Jobname.RST`. (The load step option command **OUTRES** controls which substep results are stored on `Jobname.RST`.) A typical POST1 postprocessing sequence is described below.

### 10.7.1. Points to Remember

- To review results in POST1, the database must contain the same model for which the solution was calculated.
- The results file (`Jobname.RST`) must be available.

### 10.7.2. Reviewing Results in POST1

1. Verify from your output file (`Jobname.OUT`) whether or not the analysis converged at all load steps.
  - If not, you probably won't want to postprocess the results, other than to determine why convergence failed.
  - If your solution converged, then continue postprocessing.
2. Enter POST1. If your model is not currently in the database, issue **RESUME**.  
**Command(s): /POST1**  
**GUI: Main Menu> General Postproc**
3. Read in results for the desired load step and substep, which can be identified by load step and substep numbers or by time.  
**Command(s): SET**  
**GUI: Main Menu> General Postproc> Read Results> load step**
4. Display the results using any of the following options. Note that gasket results, such as pressure and closure, are always displayed and listed in the local coordinate system.

**Option: Display Deformed Shape**

**Command(s): PLDISP**

**GUI: Main Menu> General Postproc> Plot Results> Deformed Shape**

**Option: Contour Displays**

**Command(s): PLNSOL or PLESOL**

**GUI: Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu or Element Solu**

Use these options to display contours of stresses, strains, or any other applicable item. When displaying the gasket pressure distribution, if other structural mating components are not included, ANSYS will plot the geometry of those components in gray. To have a better visualization of a gasket pressure plot, it is better for you to select gasket elements only.

**Option: Tabular Listings**

**Command(s): PRNSOL** (nodal results) **PRESOL** (element-by-element results) **PRRSOL** (reaction data) **PRITER** (substep summary data), etc.

**GUI: Main Menu> General Postproc> List Results> Nodal Solution**

**Main Menu> General Postproc> List Results> Element Solution**  
**Main Menu> General Postproc> List Results> Reaction Solution**

### Option: Animation

You can also animate gasket results over time:

**Command(s): ANTIME**

**GUI: Utility Menu> PlotCtrls> Animate> Over Time**

### Other Capabilities

Many other postprocessing functions are available in POST1. See [The General Postprocessor \(POST1\)](#) in the *Basic Analysis Guide* for details. Load case combinations usually are not valid for nonlinear analyses.

## 10.7.3. Reviewing Results in POST26

You can also review the load-history response of a nonlinear structure using POST26, the time-history postprocessor. Use POST26 to compare one ANSYS variable against another. For instance, you might graph the gasket closure vs. gasket pressure, which should correspond to the material behavior defined by **TB,GASKET**. You might also graph the displacement at a node versus the corresponding level of applied load, or you might list the gasket pressure at a node and the corresponding *TIME* value. A typical POST26 postprocessing sequence for a gasket analysis is the same as the sequence for a typical nonlinear analysis. See steps 1 through 4 in [Reviewing Results in POST26 \(p. 253\)](#) included in [Nonlinear Structural Analysis \(p. 193\)](#).

## 10.8. Sample Gasket Element Verification Analysis (Command or Batch Method)

This is a simple finite element model created to demonstrate the gasket material simulation. Two block elements with element type **SOLID185** were generated as supporters and a gasket element **INTER195** was created. The whole system is fixed along one side of each axes to prevent rigid body motion. A prescribed displacement is placed on top of the supporter element. The gasket material is assumed to have a nonlinear compression behavior with 5 different linear unloading slopes. A pressure stress cap of  $1.0e-5$  is imposed so that no tension stress is generated. Two load steps were used.

The ANSYS commands are shown below.

```
/prep7

!*+++++
!* Define Element Types
!*+++++
et,1,185
et,2,195

!*+++++
!* Define Linear Elastic Material Type 1
!*+++++
mp,ex,1,2.1E12
mp,nuxy,1,0.0

!*+++++
!* Define Initial Gap, Stable Stiffness, Stress Cap
!*+++++
delta0 = 0.00e-3
stiff0 = 0.0e7
scap = 1.0e-5
tb,gasket,2,,para
```

```

tbdata, 1,delta0,stiff0,scap

!*+++++
!* Define Gasket Compression Curve
!*+++++
tb,gask,2,1,13,comp
tbpt, ,0.460800E-03, 0.161226E+07
tbpt, ,0.511600E-03, 0.520884E+07
tbpt, ,0.562400E-03, 0.113134E+08
tbpt, ,0.613200E-03, 0.200499E+08
tbpt, ,0.664000E-03, 0.259960E+08
tbpt, ,0.714800E-03, 0.290345E+08
tbpt, ,0.765600E-03, 0.357453E+08
tbpt, ,0.816400E-03, 0.440064E+08
tbpt, ,0.867200E-03, 0.563189E+08
tbpt, ,0.918000E-03, 0.748254E+08
tbpt, ,0.968800E-03, 0.967287E+08
tbpt, ,0.101960E-02, 0.129001E+09
tbpt, ,0.109326E-02, 0.157147E+09

!*+++++
!* Define Gasket Linear Unloading Curve
!*+++++
tb,gask,2,1,5,lunl
tbpt, ,0.460800E-03, 2.430000E+11
tbpt, ,0.714800E-03, 3.565000E+11
tbpt, ,0.816400E-03, 5.92300E+11
tbpt, ,0.968800E-03, 1.088000E+12
tbpt, ,0.109326E-02, 1.490000E+12

!*+++++
!* List Gasket Material Model
!*+++++
tblist,gask,all

!*+++++
!* Define Parameters
!*+++++
n1 = 20
n2 = n1*100
n3 = n1
dis1 = -0.0008
dis2 = -0.000001
pres = 1.0e7
pres2 = 10
pres3 = 1.4E8
dp = -2.0e7
elb = 1.0
elg = 0.1

!*+++++
!* Generate Nodes
!*+++++
n,1,
n,2,1.0
n,3,1.0,1.0
n,4,0.0,1.0
ngen,2,4,1,4,,0.0,0.0,elb
ngen,2,8,1,4,,0.0,0.0,elb+elg
ngen,2,12,1,4,,0.0,0.0,2*elb+elg

!*+++++
!* Generate Front and Back SOLID185 Element
!*+++++
et,1,185,,1
mat,1
e,1,2,3,4,5,6,7,8
e,9,10,11,12,13,14,15,16

!*+++++
!* Generate Middle INTER195 Element
!*+++++

```



```

et,2,195,,
type,2
mat,2
e,5,6,7,8,9,10,11,12

!*+++++
!* Define Boundary Condition
!*+++++
nselect,s,loc,z
d,all,uz
nselect,all
nselect,s,loc,x
d,all,ux
nselect,all
nselect,s,loc,y
d,all,uy
nselect,all
finish

/solu
!*+++++
!* Apply Displacement
!*+++++
nselect,s,loc,z,elb*2+elg
d,all,uz,dis1
nselect,all

!*+++++
!* Solve First Load Step, Compress the Elements
!*+++++
nsubst,n1,n2,n3
outres,all,all
outres,svar,all
solve

!*+++++
!* Solve Second Load Step, Open the Elements
!*+++++
nsubst,n1,n2,n3
outres,all,all
outres,svar,all
nselect,s,loc,z,elb*2+elg
d,all,uz,dis2
nall
solve
finish

!*+++++
!* Postprocess the Results
!*+++++
!/post1
!set,last
!pres,s
!pres,epel
!prns,epel
!finish

!*+++++
!* Post26, Print and Plot Gasket Element Results
!*+++++
/post26
esol,2,3,,s,x,press
esol,3,3,,epel,x,delta
add,4,2,,press,,-1, ! change sign for plotting
add,5,3,,delta,,-1, ! change sign for plotting
prvar,2,3,4,5
xvar,5
plvar,4

finish

```

Presented below is the POST26 output resulting from this analysis.

-----  
\*\*\*\*\* ANSYS POST26 VARIABLE LISTING \*\*\*\*\*

TIME	3 S X press	3 EPELX delta	4 ADD press	5 ADD delta
0.50000E-01	-139488.	-0.398672E-04	139488.	0.398672E-04
0.10000	-278977.	-0.797343E-04	278977.	0.797343E-04
0.15000	-418465.	-0.119601E-03	418465.	0.119601E-03
0.20000	-557953.	-0.159469E-03	557953.	0.159469E-03
0.25000	-697442.	-0.199336E-03	697442.	0.199336E-03
0.30000	-836930.	-0.239203E-03	836930.	0.239203E-03
0.35000	-976418.	-0.279070E-03	976418.	0.279070E-03
0.40000	-0.111591E+07	-0.318937E-03	0.111591E+07	0.318937E-03
0.45000	-0.125539E+07	-0.358804E-03	0.125539E+07	0.358804E-03
0.50000	-0.139488E+07	-0.398672E-03	0.139488E+07	0.398672E-03
0.55000	-0.153437E+07	-0.438539E-03	0.153437E+07	0.438539E-03
0.60000	-0.278389E+07	-0.477349E-03	0.278389E+07	0.477349E-03
0.65000	-0.557968E+07	-0.514686E-03	0.557968E+07	0.514686E-03
0.70000	-0.989280E+07	-0.550578E-03	0.989280E+07	0.550578E-03
0.75000	-0.152775E+08	-0.585450E-03	0.152775E+08	0.585450E-03
0.80000	-0.208613E+08	-0.620132E-03	0.208613E+08	0.620132E-03
0.85000	-0.250737E+08	-0.656120E-03	0.250737E+08	0.656120E-03
0.90000	-0.277640E+08	-0.693558E-03	0.277640E+08	0.693558E-03
0.95000	-0.310936E+08	-0.730387E-03	0.310936E+08	0.730387E-03
1.0000	-0.357955E+08	-0.765909E-03	0.357955E+08	0.765909E-03
1.0500	-0.227277E+08	-0.738405E-03	0.227277E+08	0.738405E-03
1.1000	-0.965988E+07	-0.710900E-03	0.965988E+07	0.710900E-03
1.1500	-0.100000E-04	-0.680150E-03	0.100000E-04	0.680150E-03
1.2000	-0.100000E-04	-0.640200E-03	0.100000E-04	0.640200E-03
1.2500	-0.100000E-04	-0.600250E-03	0.100000E-04	0.600250E-03
1.3000	-0.100000E-04	-0.560300E-03	0.100000E-04	0.560300E-03
1.3500	-0.100000E-04	-0.520350E-03	0.100000E-04	0.520350E-03
1.4000	-0.100000E-04	-0.480400E-03	0.100000E-04	0.480400E-03
1.4500	-0.100000E-04	-0.440450E-03	0.100000E-04	0.440450E-03
1.5000	-0.100000E-04	-0.400500E-03	0.100000E-04	0.400500E-03
1.5500	-0.100000E-04	-0.360550E-03	0.100000E-04	0.360550E-03
1.6000	-0.100000E-04	-0.320600E-03	0.100000E-04	0.320600E-03
1.6500	-0.100000E-04	-0.280650E-03	0.100000E-04	0.280650E-03
1.7000	-0.100000E-04	-0.240700E-03	0.100000E-04	0.240700E-03
1.7500	-0.100000E-04	-0.200750E-03	0.100000E-04	0.200750E-03
1.8000	-0.100000E-04	-0.160800E-03	0.100000E-04	0.160800E-03
1.8500	-0.100000E-04	-0.120850E-03	0.100000E-04	0.120850E-03
1.9000	-0.100000E-04	-0.809000E-04	0.100000E-04	0.809000E-04
1.9500	-0.100000E-04	-0.409500E-04	0.100000E-04	0.409500E-04
2.0000	-0.100000E-04	-0.100000E-05	0.100000E-04	0.100000E-05

---

## Chapter 11: Fracture Mechanics

---

Cracks and flaws occur in many structures and components for several reasons. The material may be inherently defective. Cracks may be introduced during the manufacturing stage, or later as a result of environmental conditions. The presence of such cracks or flaws can significantly degrade the structural integrity of a component under the action of applied loads and environmental conditions.

Fracture mechanics uses concepts from applied mechanics to develop an understanding of the stress and deformation fields around a crack tip when a crack is present in a structure. A sound knowledge of these stress and deformation fields helps in developing fail-safe and safe-life designs for structures. Such fracture-mechanics-based design concepts are widely used, but are not limited to, the fields of nuclear, aerospace, civil, and mechanical engineering.

The following topics concerning fracture analysis are available:

- [11.1. Introduction to Fracture](#)
- [11.2. Solving Fracture Mechanics Problems](#)
- [11.3. Numerical Evaluation of Fracture Mechanics Parameters](#)
- [11.4. Learning More About Fracture Mechanics](#)

### 11.1. Introduction to Fracture

Structural design concepts traditionally use a strength-of-material approach for designing a component. This approach does not anticipate the elevated stress levels due to the existence of cracks. The presence of such stresses can lead to catastrophic failure of the structure.

Fracture mechanics accounts for the cracks or flaws in a structure. The fracture mechanics approach to the design of structures includes flaw size as one important variable, and fracture toughness replaces strength of material as a relevant material parameter.

Fracture analysis is typically carried out either using the energy criterion or the stress-intensity-factor criterion. When the energy criterion is used, the energy required for a unit extension of the crack (the energy-release rate) characterizes the fracture toughness. When the stress-intensity-factor criterion is used, the critical value of the amplitude of the stress and deformation fields characterizes the fracture toughness. Under certain circumstances, the two criteria are equivalent.

The following additional topics concerning fracture are available:

- [11.1.1. Fracture Modes](#)
- [11.1.2. Fracture Mechanics Parameter Calculation](#)
- [11.1.3. Crack Growth Simulation](#)

#### 11.1.1. Fracture Modes

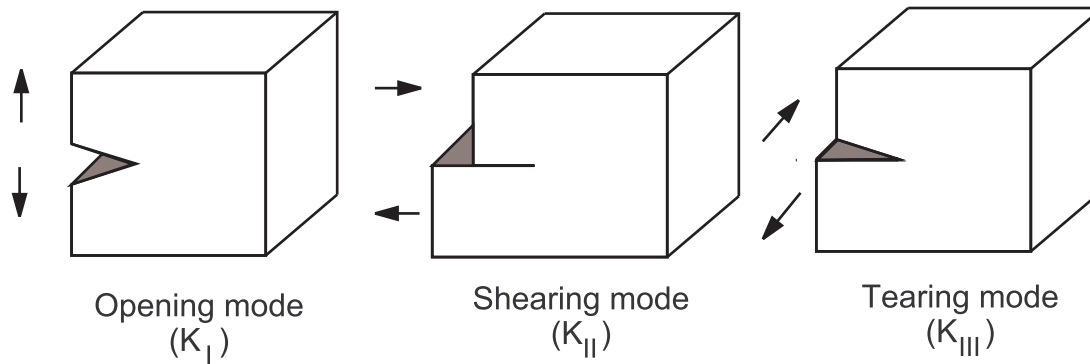
Depending on the failure kinematics (that is, the relative movement of the two surfaces of the crack), three fracture modes are distinguishable, as shown in [Figure 11.1: Schematic of the Fracture Modes \(p. 340\)](#):

- **Mode I** – Opening or tensile mode
- **Mode II** – Shearing or sliding mode

- **Mode III** – Tearing or out-of-plane mode

Fracture is generally characterized by a *combination* of fracture modes.

**Figure 11.1: Schematic of the Fracture Modes**



## 11.1.2. Fracture Mechanics Parameter Calculation

Typical fracture mechanics parameters describe either the energy-release rate or the amplitude of the stress and deformation fields ahead of the crack tip.

The following parameters are widely used in fracture mechanics analysis:

- 11.1.2.1. J-Integral
- 11.1.2.2. Energy-Release Rate
- 11.1.2.3. Stress-Intensity Factor
- 11.1.2.4. T-Stress
- 11.1.2.5. Material Force

### 11.1.2.1. J-Integral

J-Integral is one of the most widely accepted fracture mechanics parameters for linear plastic and nonlinear elastic-plastic materials. The J-Integral is defined as follows [2]:

$$J = \lim_{\Gamma \rightarrow 0} \int_{\Gamma_0} \left[ (w + T) \delta_{li} - \sigma_{ij} \frac{\partial u_j}{\partial x_l} \right] n_i d\Gamma$$

where  $W$  is the strain energy density,  $T$  is the kinematic energy density,  $\sigma$  represents the stresses,  $u$  is the displacement vector, and  $\Gamma$  is the contour over which the integration is carried out.

For a crack in a linear elastic material, the J-Integral represents the energy-release rate. Also, the amplitudes of the crack-tip stress and deformation fields are characterized by the J-Integral for a crack in a nonlinear elastic material.

For more information, see [J-Integral Calculation \(p. 351\)](#) and [J-Integral as a Stress-Intensity Factor \(p. 340\)](#).

#### 11.1.2.1.1. J-Integral as a Stress-Intensity Factor

Hutchinson [3] and Rice and Rosengren [4] independently showed that the J-Integral characterizes the crack-tip field in a nonlinear elastic material. They each assumed a power law relationship between

plastic strain and stress. If elastic strain is included, the relationship for uniaxial deformation is given as:

$$\frac{\varepsilon}{\varepsilon_0} = \frac{\sigma}{\sigma_0} + \alpha \left( \frac{\sigma}{\sigma_0} \right)^n$$

where  $\sigma_0$  is the reference stress (the yield stress of the material), and  $\varepsilon_0 = \sigma_0/E$ ,  $\alpha$  is a dimensionless constant, and  $n$  is the hardening component. They showed that, at a distance very close to the crack tip and well within the plastic zone, the crack-tip stress and strain ahead of crack tip can be expressed as:

$$\sigma_{ij} = f(\theta) \left( \frac{J}{r} \right)^{\frac{1}{n}}$$

and

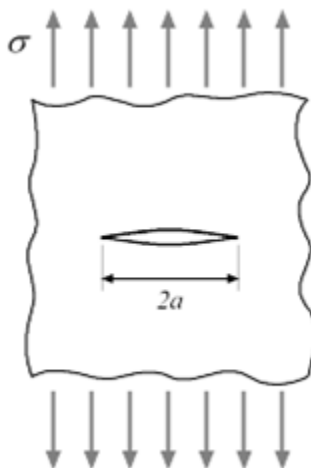
$$\varepsilon_{ij} = g(\theta) \left( \frac{J}{r} \right)^{\frac{n}{n+1}}$$

For elastic material,  $n = 1$  and the above equation predicts the  $1/\sqrt{r}$  singularity which is consistent with linear elastic fracture mechanics.

### 11.1.2.2. Energy-Release Rate

The energy-release rate, limited to linear elastic fracture mechanics, is based on the energy criterion for fracture proposed by Griffith and further development by Irwin. In this approach, the crack growth occurs when the energy available for crack growth is sufficient to overcome the resistance of the material.[1]

The energy-release rate  $G$  is defined in elastic materials as the rate of change of potential energy released from a structure when a crack opens. For example, the following illustration shows a crack of length  $2a$  in a large elastic body with modulus  $E$  subject to a tensile stress ( $\sigma$ ):



The energy-release rate is given by:

$$G = \frac{\pi \sigma^2 a}{E}$$

At the moment of fracture,  $G$  is equal to the critical energy-release rate  $G_c$ , a function of the fracture toughness. The value of  $G_c$  for a material can be determined via a relatively straightforward set of crack experiments.

For a single-fracture mode, the [stress-intensity factor](#) and the energy-release rate are related by:

$$G = \frac{K^2}{E'}$$

where  $G$  is the energy-release rate,  $E' = \frac{E}{1-\nu^2}$  for plane strain, and  $E' = E$  for plane stress. ( $E$  is the material Young's modulus, and  $\nu$  is the Poisson's ratio.)

For more information, see [VCCT Energy-Release Rate Calculation \(p. 356\)](#).

### 11.1.2.3. Stress-Intensity Factor

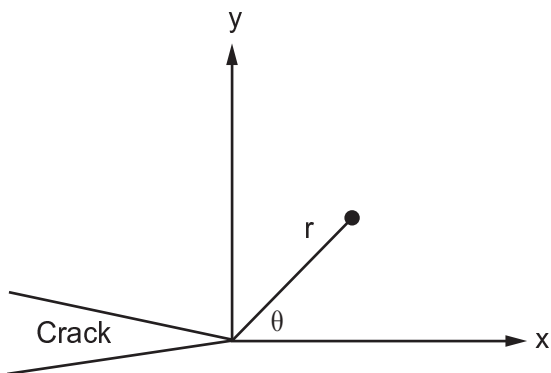
Limited to linear elastic material, the stress and strain fields ahead of the crack tip are expressed as:

$$\sigma_{ij} = -\frac{K}{\sqrt{r}} f_{ij}(\theta)$$

$$\varepsilon_{ij} = -\frac{K}{\sqrt{r}} g_{ij}(\theta)$$

where  $K$  is the stress-intensity factor,  $r$  and  $\theta$  are coordinates of a polar coordinate system (as shown in [Figure 11.2: Schematic of a Crack Tip \(p. 342\)](#)). These equations apply to any of the [three fracture modes](#).

**Figure 11.2: Schematic of a Crack Tip**



For a Mode I crack, the stress field is given as:

$$\sigma_x = \frac{K_I}{\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right) \left(1 - \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{3\theta}{2}\right)\right)$$

$$\sigma_y = \frac{K_I}{\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right) \left(1 + \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{3\theta}{2}\right)\right)$$

$$\sigma_{xy} = \frac{K_I}{\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right)$$

For more information, see [Stress-Intensity Factor Calculation](#) (p. 363).

#### 11.1.2.4. T-Stress

The asymptotic expansion of the stress field in the vicinity of the crack tip, expressed in the local polar coordinate system described in [Figure 11.2: Schematic of a Crack Tip](#) (p. 342), is represented as:

$$\tilde{A}_{ij} = \frac{K_I}{\sqrt{2\Delta r}} f_{ij}^I(\cdot) + \frac{K_{II}}{\sqrt{2\Delta r}} f_{ij}^{II}(\cdot) + \frac{K_{III}}{\sqrt{2\Delta r}} f_{ij}^{III}(\cdot) + T'_{1i'1j} + (\Delta T + E\mu_{33})'_{3i'3j} + O\left(r^{1/2}\right)$$

where the first singular terms of this eigen-expansion (the terms involving  $\frac{1}{\sqrt{r}}$ ) are the stress-intensity factors, and the first non-singular term (T) is the elastic T-stress.

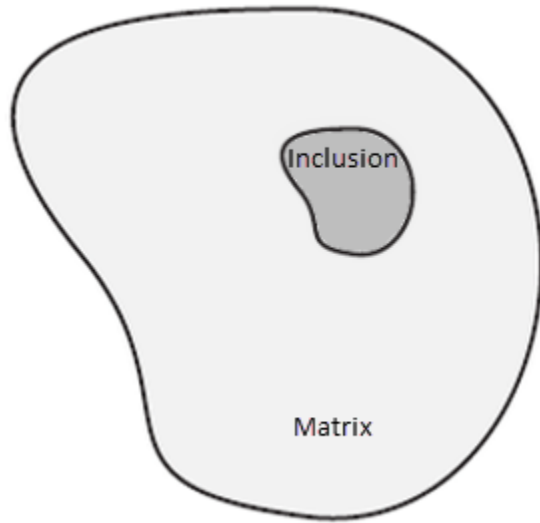
T-stress is the stress acting parallel to the crack faces. It is tightly linked to the level of crack-tip stress triaxiality, thus its sign and magnitude can substantially change the size and shape of the crack-tip plastic zone [14]. Negative T-stress values decrease the level of crack-tip triaxiality (leading to larger plastic zones), while positive values increase the level of triaxiality (leading to smaller plastic zones). A higher crack-tip triaxiality promotes fracture because the input of external work is dissipated less by the global plastic deformation and is thus available to augment local material degradation and damage [15].

T-stress also plays an important role in the stability of straight crack paths submitted to Mode I loading conditions. For a small amount of crack growth, cracks with  $T < 0$  have been shown to be stable, whereas cracks with  $T > 0$  tend to deviate from their initial propagation plane [16].

For more information, see [T-Stress Calculation](#) (p. 370).

#### 11.1.2.5. Material Force

Used primarily to analyze material defects such as dislocations, voids, interfaces and cracks, material force (also known as configurational force) can be understood by considering the presence of an inclusion in an elastic solid (matrix material), as shown in this figure:

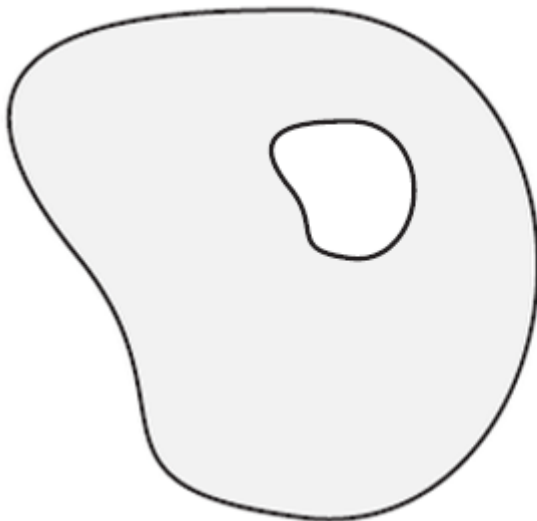
**Figure 11.3: Matrix with Inclusion**

The force exerted by the matrix on the inclusion is the material force. When an inclusion is incorporated into a stress-free elastic body, the entire body undergoes a deformation, resulting in a configurational change of the body (or matrix) from its original state. The change in the total energy due to the deformation is characterized by the material force. The material force is typically calculated by evaluating the energy-momentum tensor (or Eshelby [17] stress tensor).

When the inclusion undergoes a uniform deformation, both the matrix and the inclusion experience an elastic stress field. Now, consider the following figure:

**Figure 11.4: Thought Experiment Proposed by Eshelby**

- A. Isolate the inclusion from the matrix:

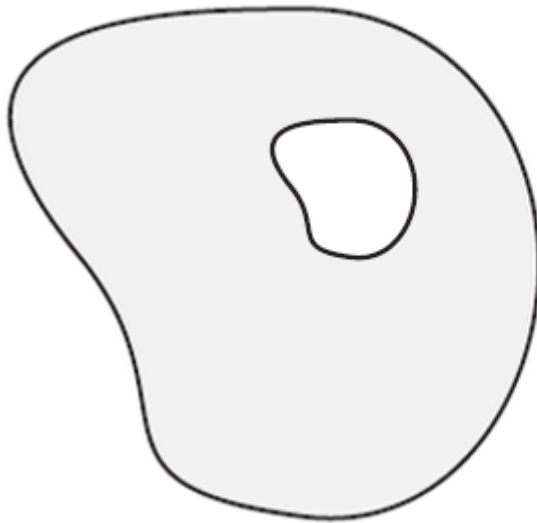


No forces are applied to the inclusion or matrix. Because the inclusion is now isolated, it undergoes a homogenous deformation. The strains experienced by the inclusion are called eigenstrains. The matrix remains stress- and strain-free.

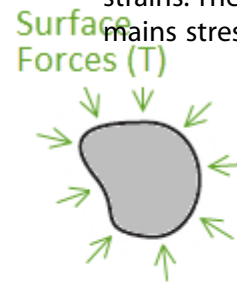
- B. Recover the original shape of inclusion by applying surface forces on the inclusion:

The elastic strains induced in the inclusion due to applied surface traction cancel out the eigen-



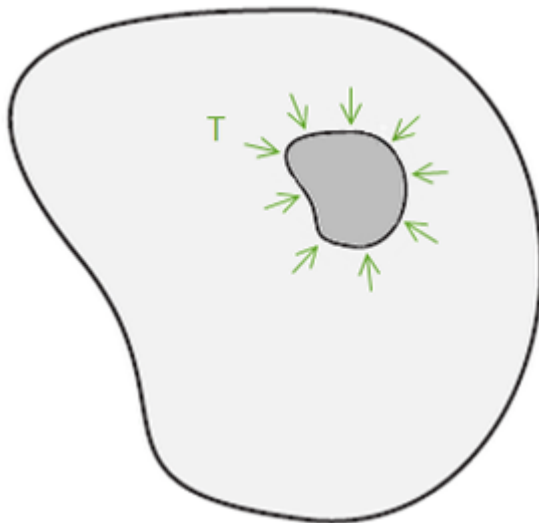


strains. The surrounding body remains stress- and strain-free.

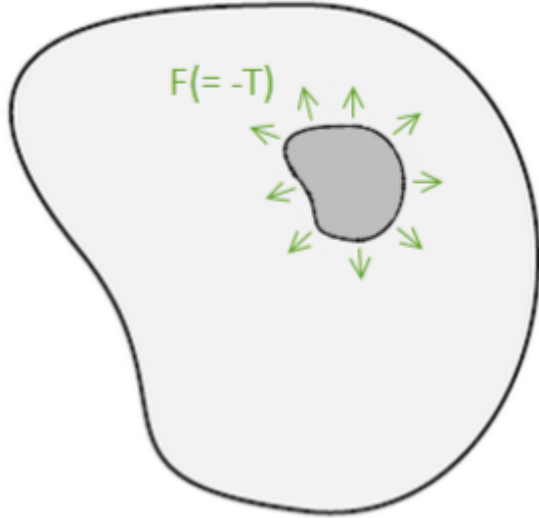


C. Replace the inclusion along with the applied surface forces:

No change of deformation occurs in either the inclusion or the matrix.



D. Remove the applied surface forces:



By removing the surface forces, we return to the original problem of a body with an inclusion, as shown in [Figure 11.3: Matrix with Inclusion](#) (p. 344).

The change from step C to step D is that a body force (or surface force on the hole in the matrix where the inclusion is inserted), equal and opposite to the surface forces in step B, is applied.

The body (or surface) force is the material force. Essentially, the presence of an inclusion creates a variation in the strain energy density in the matrix material, leading to the material force acting on the inclusion and being allowed to move through the material.

The material force method essentially determines a vectorial force-like quantity conjugate to the eigen-strain. As a general description, the material force approach is defined for elasticity as described in [Understanding the Material Force Approach](#) (p. 371). [18]

For a crack in a linear or nonlinear elastic material, the tangential component of the material force vector to the crack surface represents the energy-release rate. Also, the crack propagation direction and inhomogeneity, flaws, and mismatched mesh can be characterized by the material force vectors. In plasticity, the tangential component of the material force vector to the crack surface represents the crack-driving force. [19]

Material force calculations do not account for surface loads on crack faces.

For more information, see [Material Force Calculation](#) (p. 371).

### 11.1.3. Crack Growth Simulation

Fracture/crack growth is a phenomenon in which two surfaces are separated from each other, or material is progressively damaged under external loading. The following methods are available for simulating such failure:

[11.1.3.1. VCCT-Based Interface Element Method](#)

[11.1.3.2. Cohesive Zone Method](#)

[11.1.3.3. Gurson's Model Method](#)

#### 11.1.3.1. VCCT-Based Interface Element Method

This method uses interface elements (INTER $_{nnn}$ ) with VCCT to simulate the fracture by separating the interface elements between two materials with one or more user-specified fracture criteria. This approach applies to homogeneous material fracture as well as interfacial fracture in biomaterial systems. It is most

suitable for interface delamination of laminate composites with good numerical stability. For more information, see [VCCT-Based Crack Growth Simulation](#).

### 11.1.3.2. Cohesive Zone Method

This method uses interface (`INTERnnn`) or contact (`CONTAnnn`) elements to allow the separation of the surfaces and the cohesive material model to describe the separation behavior of the surfaces. This approach applies to both the simulation of fracture in a homogeneous material as well as interfacial delamination along the interface between two materials. For more information, see [Interface Delamination and Failure Simulation](#) (p. 381).

### 11.1.3.3. Gurson's Model Method

Gurson's model is a plasticity model (`TB,GURSON`) used to simulate ductile metal damage. The model is a micromechanics-based ductile damage model incorporating the void volume fraction into plasticity constitutive equation to represent the ductile damage process of void grow, void nucleation, and void coalescence. For more information, see [Gurson's Model](#) in the [Mechanical APDL Theory Reference](#) and the `TB,GURSON` command documentation.

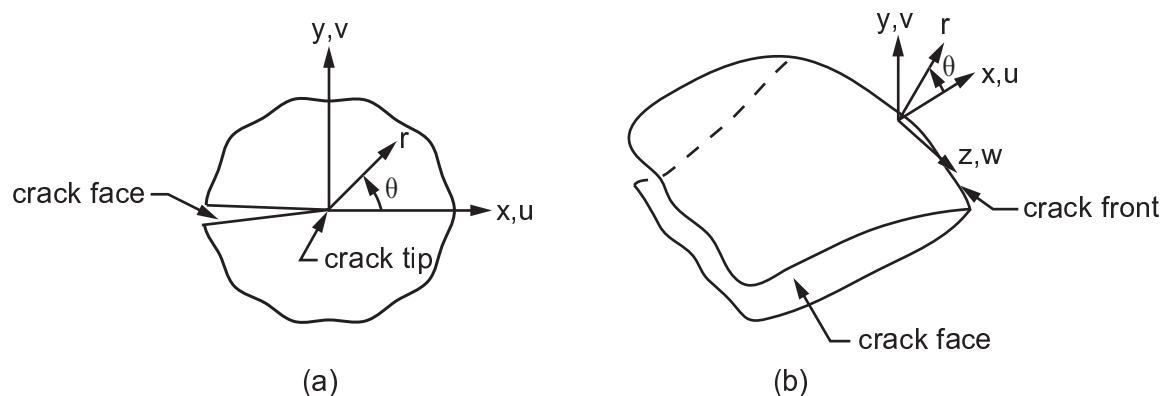
## 11.2. Solving Fracture Mechanics Problems

A fracture analysis is a combination of stress analysis and fracture mechanics parameter calculation. The stress analysis is a standard linear elastic or [nonlinear elastic plastic](#) analysis.

Because high stress gradients exist in the region around the crack tip, the finite element modeling of a component containing a crack requires special attention in that region.

The 2-D and 3-D representations of the region around the crack tip and the terminology used is outlined in the following section. The edge or tip of the crack is referred to as a *crack tip* in a 2-D model and *crack front* in a 3-D model, as illustrated in [Figure 11.5: Crack Tip and Crack Front](#) (p. 347).

**Figure 11.5: Crack Tip and Crack Front**



(a) 2-D models and (b) 3-D models

The following additional topics related to solving fracture problems are available :

[11.2.1. Modeling the Crack-Tip Region](#)

[11.2.2. Calculating Fracture Parameters](#)

## 11.2.1. Modeling the Crack-Tip Region

Stress and deformation fields around the crack tip generally have high gradients. The precise nature of these fields depends on the material, geometry, and other factors. To capture the rapidly varying stress and deformation fields, use a refined mesh in the region around the crack tip. For linear elastic problems,

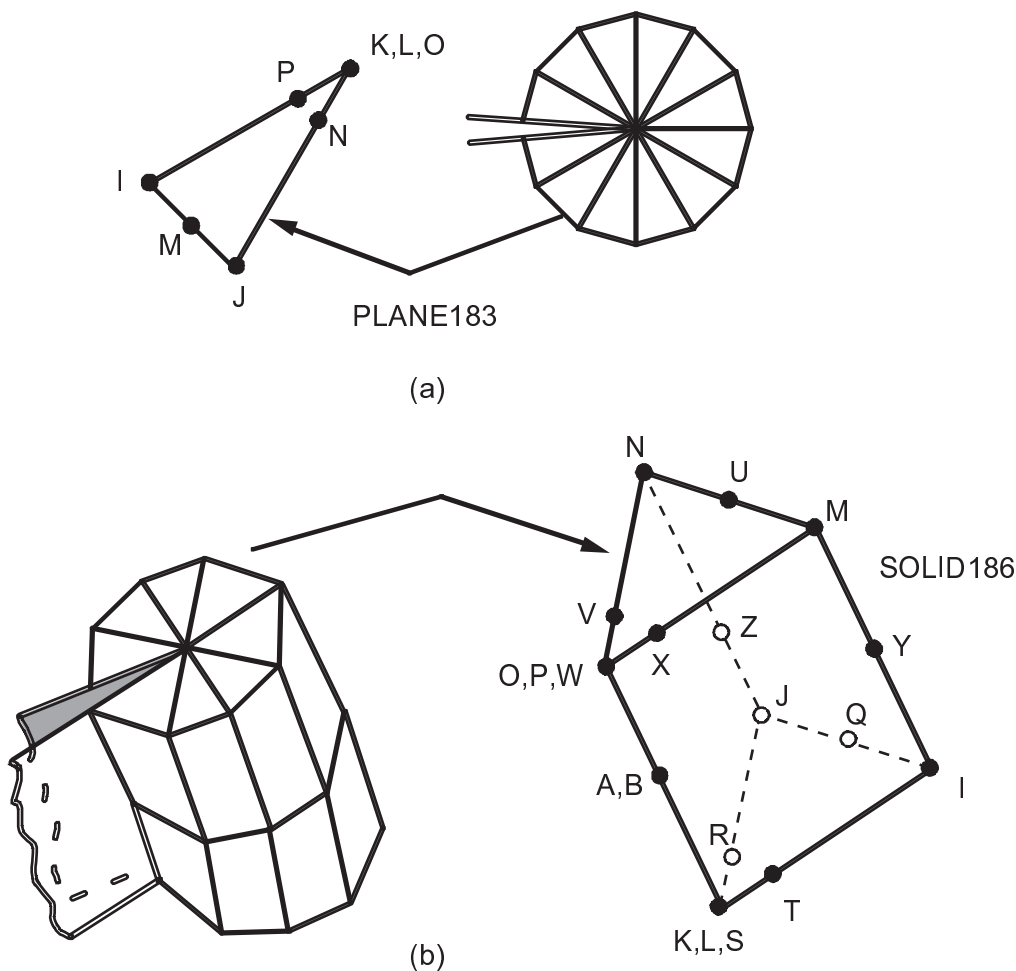
the displacements near the crack tip (or crack front) vary as  $\sqrt{r}$ , where  $r$  is the distance from the crack

tip. The stresses and strains are singular at the crack tip, varying as  $1/\sqrt{r}$ . To produce this singularity in stresses and strains, the crack tip mesh should have certain characteristics:

- The crack faces should be coincident.
- The elements around the crack tip (or crack front) should be quadratic, with the midside nodes placed at the quarter points. (Such elements are called *singular elements*.)

Figure 11.6: Singular Element Examples (p. 348) shows examples of singular elements for 2-D and 3-D fracture models.

**Figure 11.6: Singular Element Examples**



(a) 2-D models and (b) 3-D models

The following additional topics for modeling the crack-tip region are available:

### 11.2.1.1. Modeling 2-D Linear Elastic Fracture Problems

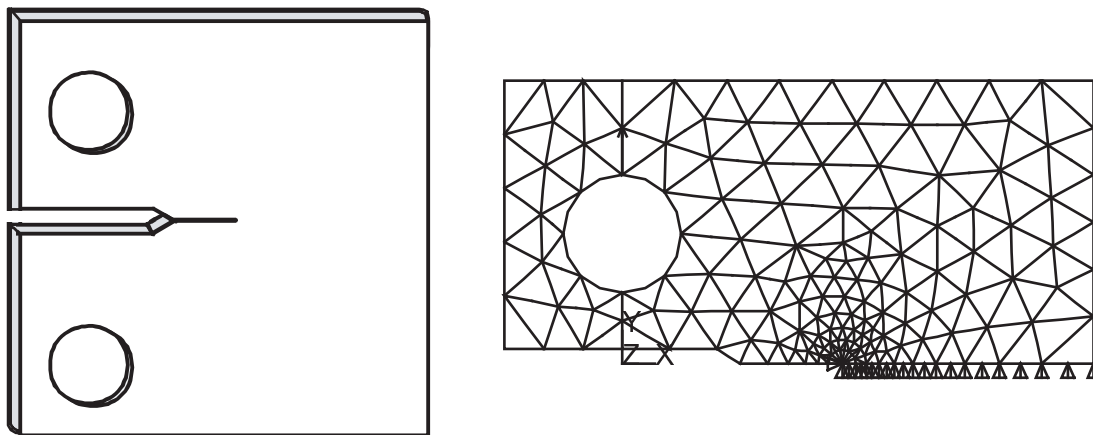
### 11.2.1.2. Modeling 3-D Linear Elastic Fracture Problems

#### 11.2.1.1. Modeling 2-D Linear Elastic Fracture Problems

The recommended element type for a 2-D fracture model is **PLANE183**, the 8-node quadratic solid. The first row of elements around the crack tip should be singular, as illustrated in [Figure 11.6: Singular Element Examples \(p. 348\)](#) (a).

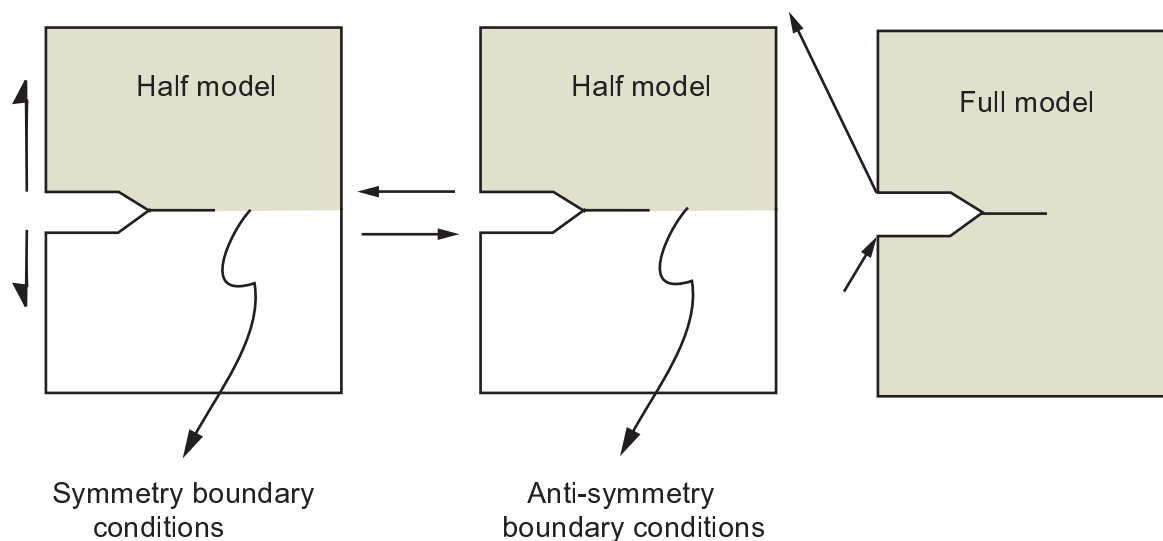
The PREP7 preprocessor's **KSCON** command (**Main Menu**> **Preprocessor**> **Meshing**> **Size Cntrls**> **Concentrat KPs**> **Create**), which assigns element division sizes around a keypoint, is particularly useful in a fracture model. It automatically generates singular elements around the specified keypoint. Other fields on the command allow you to control the radius of the first row of elements, the number of elements in the circumferential direction, and more. [Figure 11.7: Fracture Specimen and 2-D FE Model \(p. 349\)](#) shows a fracture model generated with the help of **KSCON**.

**Figure 11.7: Fracture Specimen and 2-D FE Model**



Take advantage of symmetry where possible. In many cases, you need to model only one half of the crack region, with symmetry or antisymmetry boundary conditions, as shown below.

**Figure 11.8: Using Symmetry to Your Advantage**



For reasonable results, the first row of elements around the crack tip should have a radius of approximately  $a / 8$  or smaller, where  $a$  is the crack length. In the circumferential direction, roughly one element every 30° or 40° is recommended.

The crack-tip elements should not be distorted and should take the shape of isosceles triangles.

### 11.2.1.2. Modeling 3-D Linear Elastic Fracture Problems

The recommended element type for 3-D models is **SOLID186**, the 20-node brick element. As shown in [Figure 11.6: Singular Element Examples \(p. 348\)](#) (b), the first row of elements around the crack front should be singular elements. Notice that the element is wedge-shaped, with the KLPO face collapsed into the line KO.

Generating a 3-D fracture model is considerably more involved than a 2-D model. The **KSCON** command is *not* available for 3-D modeling. You must verify that the crack front is along edge KO of the elements.

Element size recommendations are the same as for 2-D models. In addition, aspect ratios should not exceed approximately 4 to 1 in all directions.

For curved crack fronts, the element size along the crack front depends on the amount of local curvature. A general guideline is to have at least one element every 15° to 30° along a circular crack front.

All element edges should be straight, including the edge on the crack front.

### 11.2.2. Calculating Fracture Parameters

You can evaluate the following fracture parameters:

- [J-Integral](#)
- [Energy-release rate](#) ( $G_I$ ,  $G_{II}$ ,  $G_{III}$ )
- [Stress-intensity factors](#) ( $K_I$ ,  $K_{II}$ ,  $K_{III}$ )
- [T-stress](#)
- [Material force](#)

The J-Integral, energy-release rate, and stress-intensity factors are calculated during the solution phase of the analysis (**CINT**). The results are stored to the `.rst` results file for postprocessing.

The stress-intensity factors can also be calculated during (POST1) postprocessing (**KCALC**).

For more information, see [Numerical Evaluation of Fracture Mechanics Parameters](#).

## 11.3. Numerical Evaluation of Fracture Mechanics Parameters

Several tools are available for evaluating fracture mechanics parameters:

- The [J-Integral calculation](#) is based on the domain integral approach and is performed during the solution phase of the analysis (**CINT**).
- Direct [energy-release rate](#) calculation, based on the virtual crack closure technique (VCCT), is performed at solution (**CINT**).

- **Stress-intensity factors** calculation with the interaction integral approach during solution (**CINT**)
- **Stress-intensity factors** calculation with extrapolation during postprocessing (**KCALC**).
- **T-stress** calculation using the interaction integral approach during solution (**CINT**).
- **Material force** calculation is performed during the solution (**CINT**).

Most fracture calculations rely on the element nodal connectivity order and must conform to the pattern shown for each element described in the *Element Reference*.

### 11.3.1. J-Integral Calculation

The J-Integral evaluation is based on the domain integral method by Shih[5]. The domain integration formulation applies area integration for 2-D problems and volume integration for 3-D problems. Area and volume integrals offer much better accuracy than contour integral and surface integrals, and are much easier to implement numerically. The method itself is also very easy to use.

The following topics concerning J-Integral calculation are available:

11.3.1.1. [Understanding the Domain Integral Method](#)

11.3.1.2. [J-Integral Calculation](#)

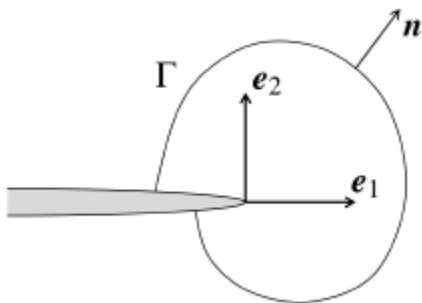
#### 11.3.1.1. Understanding the Domain Integral Method

For a 2-D problem, and in the absence of thermal strain, path dependent plastic strains, body forces within the integration of area, and pressure on the crack surface, the domain integral representation of the J-Integral is given by:

$$J = \int_A \left[ \sigma_{ij} \frac{\partial u_j}{\partial x_i} - w \delta_{ij} \right] \frac{\partial q}{\partial x_i} dA$$

where  $\sigma_{ij}$  is the stress tensor,  $u_j$  is the displacement vector,  $w$  is the strain energy density,  $\delta_{ij}$  is the Kronecker delta,  $x_i$  is the coordinate axis, and  $q$  is referred to as the crack-extension vector.

The direction of  $q$  is the simple x-axis of the local coordinate system ahead of the crack tip. The  $q$  vector is chosen as zero at nodes along the contour  $\Gamma$ , and is a unit vector for all nodes inside  $\Gamma$  except the midside nodes, if there are any, that are directly connected to  $\Gamma$ . The program refers to these nodes with a unit  $q$  vector as *virtual crack-extension nodes*.



The discretized form of the J-Integral is given by:

$$J = \sum_{ie=1}^{ne} \left[ \sigma_{ij} \frac{\partial u_j}{\partial x_1} - w \delta_{1i} \right] \frac{\partial q}{\partial x_i} w_{iw} A_{ie}$$

where  $ne$  is the number of elements to be integrated,  $w_{iw}$  is the weight function, and  $A_{ie}$  is the area of the element represented by  $ie$ .

For higher-order elements (such as [PLANE183](#) and [SOLID186](#)), the  $q$  vector at midside nodes takes the averaged values from the corresponding corner nodes.

If the thermal strains exist in the structure and the surface tractions act on crack faces, the J-Integral is expressed as:

$$J = \int_A \left[ \sigma_{ij} \frac{\partial u_j}{\partial x_1} - w \delta_{1i} \right] \frac{\partial q}{\partial x_i} dA + \int \alpha \sigma_{ii} \frac{\partial \theta}{\partial x_1} q_1 dA - \int_C t_j u_{j,1} q_1 ds$$

where  $\alpha$  is the thermal expansion coefficient,  $t_j$  is the crack face traction, and  $C$  is crack face upon which the tractions act.

For a 3-D problem, domain integral representation of the J-Integral becomes a volume integration, which again is evaluated over a group of elements. The implementation becomes more complicated; however, the principal is similar to the 2-D problem.

### 11.3.1.1.1. Virtual Crack-Extension Nodes and J-Integral Contours

Virtual crack-extension nodes are one of the most important input data elements required for J-Integral evaluation. It is also referred to as the crack-tip node component.

For a 2-D crack problem, the crack-tip node component usually contains one node which is also the crack-tip node. The first contour for the area integration of the J-Integral is evaluated over the elements associated with the crack-tip node component. The second contour for the area integration of the J-Integral is evaluated over the elements adjacent to the first contour of elements. This procedure is repeated for all contours. To ensure correct results, the elements for the contour integration should not reach the outer boundary of the model (with the exception of the crack surface).

For a 3-D crack problem, the crack-tip node component is comprised of the nodes along the crack front. The crack-tip node component is not required to be sorted. The 3-D J-Integral contour follows a procedure similar to that of the 2-D contour.

### 11.3.1.1.2. Element Selection and Material Behavior

J-Integral or [stress-intensity](#) evaluation (accessed via the [CINT](#) command) supports the following elements:

- [PLANE182](#)
- [PLANE183](#)
- [SOLID185](#)
- [SOLID186](#)
- [SOLID187](#)



J-Integral or stress-intensity evaluation supports the following material behavior:

- Linear isotropic elasticity
- Isotropic Plasticity

### 11.3.1.2. J-Integral Calculation

The program calculates the J-Integral at the solution phase of the analysis after a substep has converged, then stores the value to the results file.

The **CINT** command initiates the J-Integral calculation and also specifies the parameters necessary for the calculation.

Perform the J-Integral calculation as follows:

11.3.1.2.1. Step 1: Initiate a New J-Integral Calculation

11.3.1.2.2. Step 2: Define Crack Information

11.3.1.2.3. Step 3: Specify the Number of Contours to Calculate

11.3.1.2.4. Step 4: Define a Crack Symmetry Condition

11.3.1.2.5. Step 5: Specify Output Controls

#### 11.3.1.2.1. Step 1: Initiate a New J-Integral Calculation

To start a J-Integral calculation, use the **CINT** command's NEW option and provide a number to identify the input information for the J-Integral calculation. The command syntax is:

**CINT**,NEW,*n*

where *n* is the number identifying this J-Integral calculation.

For example:

```
CINT,NEW,1 ! initiate a new J-Integral calculation as # 1
```

#### 11.3.1.2.2. Step 2: Define Crack Information

The crack-tip node component and the crack-extension direction are both necessary for a J-Integral calculation. Two methods using the **CINT** command are available for specifying the values:

- **Define the crack-tip node component and the crack-plane normal.**

This approach applies for both 2-D crack geometry and 3-D flat crack surfaces. It offers a simple way to define a 3-D J-Integral calculation, as you need only define the crack-tip (front) node component and the normal of the crack plane. Use this method when the crack plane is flat.

- **Define the crack-extension node component and crack-extension direction.**

This approach applies for 3-D curve crack planes, where a unique normal may not exist. However, you must define the crack-extension node component and the crack-extension direction at each crack-tip node location. Use this method when the crack plane is not flat, or when a set of nodes form the crack tip, as in the case of a collapsed crack-tip mesh.

### 11.3.1.2.2.1. Define the Crack-Tip Node Component and Crack-Plane Normal

For 2-D crack geometry, define a crack-tip node component (usually a node located at the crack tip). You can also define a group of nodes around the crack tip, including the node at the crack tip. ANSYS uses this group of nodes as the starting nodes to form the necessary information for the contour integration automatically.

For 3-D flat crack geometry, you must define a crack-tip node component that includes all of the nodes along the crack front. At each node location, however, only one node can exist. All nodes in the crack-tip node component must be connectable, and they must form a line based on the element connectivity associated with it. This line is the crack front. ANSYS uses it to automatically determine the elements for the contour integration. The procedure is similar to 2-D crack geometry, and is done through all the nodes along the crack front.

The command syntax is:

```
CINT, CTNC, CMNAME
```

After the crack-tip node component is defined, use the **CINT** command's NORM option to define the normal of the crack plane. The program automatically converts it into the crack-extension vector  $q$ , based on the element information. The crack-extension vector is taken along the perpendicular direction to the plane formed by the crack-plane normal and the tangent direction of the crack-tip node, and is normalized to a unit vector.

The command syntax is:

```
CINT, NORM, par1, par2
```

where par1 is the coordinate system number and par2 is the axis of coordinate system par1.

#### Example 1

```
!local coordinate system
LOCAL,11,0,,,,
! select nodes located along the crack front and define it as crack front/tip node component
NSEL,S,LOC,X,Xctip
NSEL,R,LOC,Y,Yctip
CM,CRACK_TIP_NODE_CM
! Define a new J-Integral calculation
CINT,NEW,1
CINT,CTNC,CRACK_TIP_NODE_CM
CINT,NORM,11,2
```

#### Example 2

```
! select nodes located along the crack front and define it as crack front/tip node component
LSEL,,,,
NSLL
CM,CRACK_FRONT_NODE_CM,NODE
CINT,NEW,1
CINT,CTNC,CRACK_FRONT_NODE_CM,
```

### 11.3.1.2.2.2. Define the Crack-Extension Node Component and Crack-Extension Direction

Use this method when the crack plane is not flat, or when a set of nodes form the crack tip, as in the case of a collapsed crack-tip mesh:

1. **Define a node component consisting of one or more nodes forming the crack tip.** The node component can have one or more nodes.

Example: **CINT, CENC, CMNAME**

2. **Identify the crack-tip node separately if the node component has more than one node.** If a crack-tip node is not identified, then the first node of the node component is taken to be the first node.

*Example:* `CINT ,CENC ,CMNAME ,node1`

3. **Define the crack-extension direction.**

Identify the local coordinate system associated with the crack under consideration.

Identify the local axis (for the above CS) along which the crack is supposed to extend.

*Example:* `CINT ,CENC ,CMNAME ,node1 ,11 , 2`

Alternatively, define the crack-extension direction by directly specifying the global X Y and Z components of the crack-extension vector.

*Example:* `CINT ,CENC ,CMNAME ,node1 , , , compx ,compy ,compz`

Repeat this method for all node locations along the crack front.

### Example 1

```
! local coordinate systems
local,11,0,,,,
local,12,0,,,,
...
local,n,0,,,,

NSEL,S,LOC,X,Xctip1
NSEL,R,LOC,Y,Yctip1
NSEL,R,LOC,Z,Zctip1
CM,CRACK_FRONT_NODE_CM1
NSEL,S,LOC,X,Xctip2
NSEL,R,LOC,Y,Yctip2
NSEL,R,LOC,Z,Zctip2
CM,CRACK_FRONT_NODE_CM2
...
NSEL,S,LOC,X,Xctipn
NSEL,R,LOC,Y,Yctipn
NSEL,R,LOC,Z,Zctipn
CM,CRACK_FRONT_NODE_CMn

CINT,NEW,1
CINT,CENC,CRACK_FRONT_NODE_CM1,,11,2
CINT,CENC,CRACK_FRONT_NODE_CM2,,11,2
...
CINT,CENC,CRACK_FRONT_NODE_CM2,,n,2
```

### Example 2

```
! Crack-extension node component and
! crack-extension direction specification using vectors
NSEL,S,LOC,X,Xctip1
NSEL,R,LOC,Y,Yctip1
NSEL,R,LOC,Z,Zctip1
CM,CRACK_FRONT_NODE_CM1
NSEL,S,LOC,X,Xctip2
NSEL,R,LOC,Y,Yctip2
NSEL,R,LOC,Z,Zctip2
CM,CRACK_FRONT_NODE_CM2
...
NSEL,S,LOC,X,Xctipn
NSEL,R,LOC,Y,Yctipn
NSEL,R,LOC,Z,Zctipn
CM,CRACK_FRONT_NODE_CMn
```

```
CINT,NEW,1
CINT,CENC,CRACK_FRONT_NODE_CM1, , , ,Vx1,Vy1,Vz1
CINT,CENC,CRACK_FRONT_NODE_CM2, , , ,Vx2,Vy2,Vz2
...
CINT,CENC,CRACK_FRONT_NODE_CM2, , , ,Vxn,Vyn,Vzn
```

### 11.3.1.2.3. Step 3: Specify the Number of Contours to Calculate

Specify the number of contours for the J-Integral evaluation, as follows:

```
CINT,NCON,n
```

where  $n$  is the number of contours.

For 3-D crack geometry, every node along the crack front has the same number of contours.

#### Example

```
CINT,NEW,1
CINT,CTNC,CRACK_FRONT_NODE_SET,NODE
CINT,NCON,6
```

### 11.3.1.2.4. Step 4: Define a Crack Symmetry Condition

If the crack is located along a symmetry plane, and only a half model is created, define a symmetric condition so that ANSYS can account for it:

```
CINT,SYMM,ON
```

#### Example

```
CINT,NEW,1
CINT,SYMM,ON ! crack 1 is a symmetrical crack
```

### 11.3.1.2.5. Step 5: Specify Output Controls

Mechanical APDL calculates the J-Integral at the solution level and stores it to the results file for post-processing. J-Integral output uses all the defaults from the **OUTRES** command. The command **OUTRES,ALL** includes **CINT** command results. However, you can issue an **OUTRES,CINT** command to control the specific output for J-Integral results only.

#### Example

```
CINT,NEW,1
CINT,CTNC,CRACK_TIP_NODE_CM
CINT,SYMM,ON
CINT,NCON,5
OUTRES,CINT,10
! output J-Integral every 10 substeps
```

## 11.3.2. VCCT Energy-Release Rate Calculation

The approach for evaluating the energy-release rate is based on the virtual crack-closure technique (VCCT). The energy-release rate calculation occurs during the solution phase of the analysis and the results are saved for postprocessing.

The following energy-release rate calculation topics are available:

[11.3.2.1. Using VCCT for Energy-Release Rate Calculation](#)

[11.3.2.2. Process for Calculating the Energy-Release Rate](#)

### 11.3.2.1. Using VCCT for Energy-Release Rate Calculation

VCCT is based on the assumption that the energy needed to separate a surface is the same as the energy needed to close the same surface. The implementation described here uses the modified crack-closure method (a VCCT-based method) and assumes further that stress states around the crack tip do not change significantly when the crack grows by a small amount ( $\Delta a$ ).

#### 11.3.2.1.1. 2-D Crack Geometry

For 2-D crack geometry with a low-order element mesh, the energy-release rate is defined as:

$$G_I = -\frac{1}{2\Delta a} R_Y \Delta v$$

$$G_{II} = -\frac{1}{2\Delta a} R_X \Delta u$$

where:

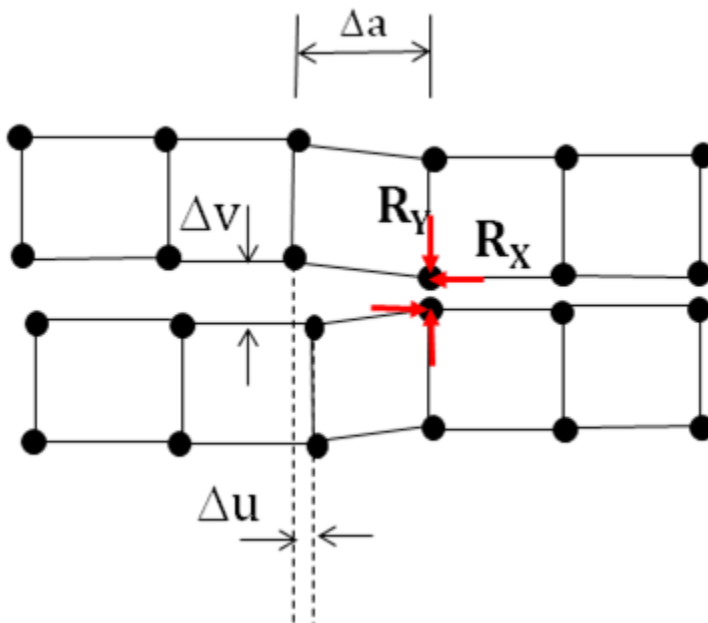
$G_I$  and  $G_{II}$  = mode I and II energy-release rate, respectively

$\Delta u$  and  $\Delta v$  = relative displacement between the top and bottom nodes of the crack face in local coordinates  $x$  and  $y$ , respectively

$R_x$  and  $R_y$  = reaction forces at the crack-tip node

$\Delta a$  = crack extension, as shown in the following figure:

**Figure 11.9: 2-D Crack Geometry Schematic**



#### 11.3.2.1.2. 3-D Crack Geometry

For 3-D crack geometry with a low-order element mesh, the energy-release rate is defined as:

$$G_I = -\frac{1}{2\Delta A} R_Y \Delta v$$

$$G_{II} = -\frac{1}{2\Delta A} R_x \Delta u$$

$$G_{III} = -\frac{1}{2\Delta A} R_z \Delta w$$

where:

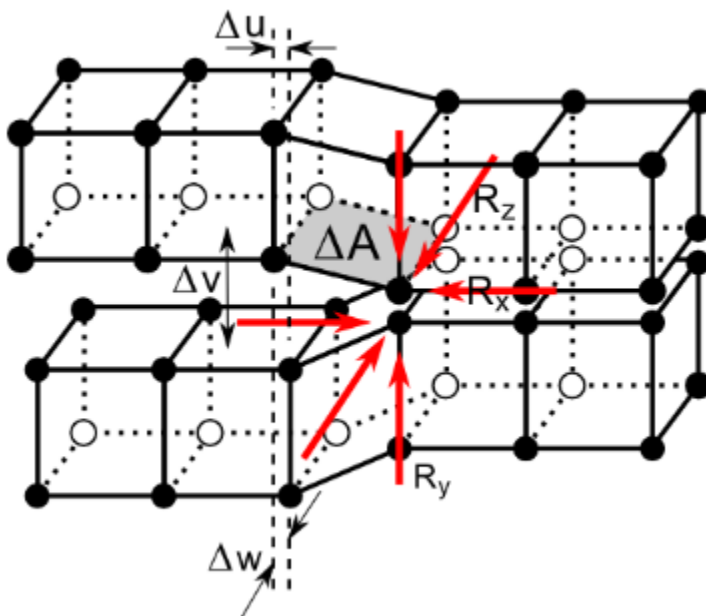
$G_I$ ,  $G_{II}$ , and  $G_{III}$  = mode I, II, and III energy-release rate, respectively

$\Delta u$ ,  $\Delta v$ , and  $\Delta w$  = relative displacement between the top and bottom nodes of the crack face in local coordinates  $x$ ,  $y$ , and  $z$ , respectively

$R_x$ ,  $R_y$ , and  $R_z$  = reaction forces at the crack-tip node

$\Delta A$  = crack-extension area, as shown in the following figure:

**Figure 11.10: 3-D Crack Geometry Schematic**



### 11.3.2.1.3. Element Support, Mesh and Material Behavior

The VCCT method for energy-release rate calculation (accessed via the **CINT** command) supports the following elements:

- PLANE182
- PLANE183
- SOLID185
- SOLID186

In most cases, ANSYS, Inc. recommends using linear elements including **PLANE182** and **SOLID185**.

The accuracy of the VCCT calculation depends on the meshes. To ensure the greatest accuracy, use equal element sizes ahead of and behind the crack-tip node. The mesh size affects the solution; therefore, it is helpful to examine mesh-size convergence prior to attempting the finite element solution.

The VCCT method for energy-release rate calculation supports the following material behaviors:

- Linear isotropic elasticity
- Orthotropic elasticity
- Anisotropic elasticity

### 11.3.2.2. Process for Calculating the Energy-Release Rate

The **CINT** command's VCCT option initiates the energy-release rate calculation.

Similar to the [J-integral calculation](#), **CINT** specifies the parameters necessary for the calculation.

For **CINT** element-type and material-behavior support, see [Element Selection and Material Behavior](#) (p. 352).

Following is the general process for calculating the energy-release rate:

[11.3.2.2.1. Step 1: Initiate a New Energy-Release Rate Calculation](#)

[11.3.2.2.2. Step 2: Define Crack Information](#)

[11.3.2.2.3. Step 3: Define a Crack Symmetry Condition](#)

[11.3.2.2.4. Step 4: Specify Output Controls](#)

#### 11.3.2.2.1. Step 1: Initiate a New Energy-Release Rate Calculation

Issue the **CINT** command twice, as shown:

```
CINT,NEW,n
CINT,TYPE,VCCT
```

where *n* is the identifier for this energy-release rate calculation (for example, 1).

#### 11.3.2.2.2. Step 2: Define Crack Information

Similar to the [J-integral calculation](#), the crack-tip node component and the crack-extension direction are both necessary for the energy-release rate calculation.

VCCT requires the finite element mesh to be in the crack-extension direction. To ensure the accuracy of the energy-release rate calculation, it is crucial that you correctly define the crack extension. How you do so depends upon whether the crack plane is flat or not:

[11.3.2.2.2.1. Specifying Crack Information When the Crack Plane Is Flat](#)

[11.3.2.2.2.2. Specifying Crack Information When the Crack Plane Is Not Flat](#)

##### 11.3.2.2.2.1. Specifying Crack Information When the Crack Plane Is Flat

This approach applies to both 2-D crack geometry and 3-D flat crack surfaces. It offers a simple way to define a 3-D energy-release rate calculation, as you need only define the crack-tip (front) node component and the normal of the crack plane.

#### 2-D Flat Crack Geometry

For 2-D crack geometry, define a crack-tip node component (usually a node located at the crack tip). You can also define a group of nodes around the crack tip, including the node at the crack tip. The

program uses this group of nodes to form the necessary information for the VCCT calculation automatically.

### 3-D Flat Crack Geometry

For 3-D flat crack surfaces, define a crack-tip node component that includes all of the nodes along the crack front. At each node location, however, only one node can exist.

All nodes in the crack-tip node component must be connectable, and they must form a line based on the element connectivity associated with it. This line is the crack front. The program uses it to determine the elements and the nodes needed for the VCCT calculation automatically.

VCCT is not applicable in the case of a collapsed crack-tip mesh.

The command syntax is:

```
CINT,CTNC,Par1,Par2,Par3
```

where CTNC specifies a crack-tip node component, *Par1* is the crack-tip node component name, *Par2* is the crack-extension direction calculation-assist node (any node on the open side of the crack), and *Par3* is the crack front's end-node crack-extension direction override.

The *Par1* and *Par2* values help to identify the crack-extension direction. Although the program automatically calculates the energy-release rate at the crack tip using the local coordinate system, it is usually best to use *Par2* to define a crack face node to help align the extension directions of the crack-tip nodes.

By default, the program uses the external surface to determine the crack-extension direction and normal when the crack-tip node hits the free surface. You can use *Par3* to override this default.

After the crack-tip node component is defined, define the normal of the crack plane. The program automatically converts it into the crack-extension vector  $q$ , based on the element information. The crack-extension vector is taken along the perpendicular direction to the plane formed by the crack-plane normal and the tangent direction of the crack-tip node, and is normalized to a unit vector.

The command syntax is:

```
CINT,NORM,Par1,Par2
```

where *Par1* is the coordinate system number and *Par2* is the coordinate system axis.

### Example 1:

```
! Local coordinate system
LOCAL,11,0,,,,
! select nodes located along the crack front and
! define it as crack front/tip node component
NSEL,S,LOC,X,Xctip
NSEL,R,LOC,Y,Yctip
CM,CRACK_TIP_NODE_CM

! Define a new the energy-release rate calculation
CINT,NEW,1
CINT,TYPE,VCCT
CINT,CTNC,CRACK_TIP_NODE_CM
CINT,NORM,11,2
```



**Example 2:**

```

!   Select nodes located along the crack front and
!       define it as crack front/tip node component
LSEL,,,
NSLL
CM,CRACK_FRONT_NODE_CM,NODE
CINT,NEW,1
CINT,TYPE,VCCT
CINT,CTNC,CRACK_FRONT_NODE_CM

```

**11.3.2.2.2. Specifying Crack Information When the Crack Plane Is Not Flat**

This approach applies to 3-D curved crack planes, where a unique normal may not exist. However, you must define the crack-extension node component and the crack-extension direction at each crack-tip node location:

**1. Define a node component consisting of one or more nodes forming the crack tip.**

The node component can have one or more nodes.

**Example:** `CINT,CENC,CompName`

**2. If the node component has more than one node, identify the crack-tip node separately.**

If a crack-tip node is not identified, the first node of the node component is used as the first node.

**Example:** `CINT,CENC,CompName,Node1`

**3. Define the crack-extension direction.**

Identify the local coordinate system associated with the crack under consideration, and identify the local axis along which the crack should extend.

**Example:** `CINT,CENC,CompName,Node1,1,2`

Alternatively, define the crack-extension direction by directly specifying the global X, Y, and Z components of the crack-extension vector.

**Example:** `CINT,CENC,CompName,Node1,,,compx,compy,compz`

Repeat this method for all node locations along the crack front. Although the program automatically calculates the local coordinate system at the crack tip to determine the energy-release rate, it is usually best to use the NORM option to help align the calculated normals of the crack-tip nodes.

**Example 1:**

```

!   Local coordinate systems
local,11,0,,,,
local,12,0,,,,
...
local,n,0,,,,

NSEL,S,LOC,X,Xctip1
NSEL,R,LOC,Y,Yctip1
NSEL,R,LOC,Z,Zctip1
CM,CRACK_FRONT_NODE_CM1
NSEL,S,LOC,X,Xctip2
NSEL,R,LOC,Y,Yctip2
NSEL,R,LOC,Z,Zctip2
CM,CRACK_FRONT_NODE_CM2
...

```

```

NSEL,S,LOC,X,Xctipn
NSEL,R,LOC,Y,Yctipn
NSEL,R,LOC,Z,Zctipn
CM,CRACK_FRONT_NODE_CMn

CINT,NEW,1
CINT,TYPE,VCCT
CINT,CENC,CRACK_FRONT_NODE_CM1,,11,2
CINT,CENC,CRACK_FRONT_NODE_CM2,,11,2
...
CINT,CENC,CRACK_FRONT_NODE_CMn,,n,2

```

**Example 2:**

```

! Crack-extension node component and
! crack-extension direction specification using vectors
NSEL,S,LOC,X,Xctip1
NSEL,R,LOC,Y,Yctip1
NSEL,R,LOC,Z,Zctip1
CM,CRACK_FRONT_NODE_CM1
NSEL,S,LOC,X,Xctip2
NSEL,R,LOC,Y,Yctip2
NSEL,R,LOC,Z,Zctip2
CM,CRACK_FRONT_NODE_CM2
...
NSEL,S,LOC,X,Xctipn
NSEL,R,LOC,Y,Yctipn
NSEL,R,LOC,Z,Zctipn
CM,CRACK_FRONT_NODE_CMn

CINT,NEW,1
CINT,TYPE,VCCT
CINT,CENC,CRACK_FRONT_NODE_CM1,, ,Vx1,Vy1,Vz1
CINT,CENC,CRACK_FRONT_NODE_CM2,, ,Vx2,Vy2,Vz2
...
CINT,CENC,CRACK_FRONT_NODE_CMn,, ,Vxn,Vyn,Vzn

```

**Local Crack-Tip Coordinate System**

The VCCT calculation is based on the local crack-tip coordinate systems. To ensure the accuracy of the energy-release rate calculation, it is crucial to have a local crack-tip coordinate system in which the local x-axis is pointed to the crack extension, the local y-axis is pointed to the normal of the crack surfaces or edges, and the local z-axis pointed to the tangential direction of the crack front.

Local coordinate systems must be consistent across all nodes along the crack front. A set of inconsistent coordinate systems results in irregular behavior of the energy-release rate distribution along crack front.

The program automatically calculates the local coordinate systems based on the input crack front nodes and the normal of the crack surface or extension directions. Because there may be not enough information to determine a set of consistent coordinate systems, however, ANSYS, Inc. recommends:

- Using the **CINT** command's CTNC option to define a crack-face node to help identify the coordinate systems, *or*
- Using the **CINT** command's NORM option, followed by the CENC option, to define the coordinate systems.

**11.3.2.2.3. Step 3: Define a Crack Symmetry Condition**

If the crack is located along a symmetry plane, and only a half model is created, define a symmetric condition so that the program can account for it. To do so, issue the following command:

**CINT**,SYMM,ON

**Example:**

```
CINT,NEW,1
CINT,TYPE,VCCT
CINT,SYMM,ON ! crack 1 is a symmetrical crack
```

### 11.3.2.2.4. Step 4: Specify Output Controls

Similar to the [J-Integral calculation](#), the program calculates the energy-release rate during the solution phase of the analysis and stores the results in the `.rst` file for postprocessing.

Energy-release rate output uses all of the defaults from the **OUTRES** command. The **OUTRES,ALL** command includes **CINT** command results. However, you can issue an **OUTRES,CINT** command to control the specific output for energy-release rate results only.

**Example:**

```
CINT,NEW,1
CINT,TYPE,VCCT
CINT,CTNCP,CRACK_TIP_NODE_CM
CINT,SYMM,ON
OUTRES,CINT,10 ! output CINT results every 10 substeps
```

## 11.3.3. Stress-Intensity Factor Calculation

Two approaches for evaluating stress-intensity factors are available:

- **Interaction integral method** -- ANSYS performs the stress-intensity factors calculation at solution and stores the results to the result file for postprocessing.
- **Displacement extrapolation method** -- ANSYS performs the stress-intensity factors calculation during postprocessing.

### 11.3.3.1. Calculating Stress-Intensity Factors via Interaction Integrals

Similar to the [domain integral method for J-Integral evaluation](#), the interaction integral method for stress-intensity factors calculation applies area integration for 2-D problems and volume integration for 3-D problems. In comparison to the traditional [displacement extrapolation method](#), the interaction integral method offers better accuracy, fewer mesh requirements, and ease of use.

The following topics are available:

- 11.3.3.1.1. Understanding Interaction Integral Formulation
- 11.3.3.1.2. Calculating the Stress-Intensity Factors

#### 11.3.3.1.1. Understanding Interaction Integral Formulation

The interaction integral is defined as

$$I = -\int_V q_{i,j} \left( \sigma_{kl} \epsilon_{kl}^{aux} \delta_{ij} - \sigma_{kj}^{aux} u_{k,i} - \sigma_{kj} u_{k,i}^{aux} \right) dV / \int_S \delta q_n ds$$

where:

$\sigma_{ij}, \epsilon_{ij}, u_i$  are the stress, strain and displacement,

$\sigma_{ij}^{aux}$ ,  $\epsilon_{ij}^{aux}$ ,  $u_i^{aux}$  are the stress, strain and displacement of the auxiliary field, and  $q_i$  is the crack-extension vector.

The interaction integral is associated with the stress-intensity factors as

$$I = \frac{2}{E^*} (K_1 K_1^{aux} + K_2 K_2^{aux}) + \frac{1}{\mu} K_3 K_3^{aux}$$

where:

$K_i$  ( $i = 1,2,3$ ) = mode I, II, and III stress-intensity factors

$K_i^{aux}$  ( $i = 1,2,3$ ) = auxiliary mode I, II and III stress-intensity factors

$E^* = E$  for plane stress and  $E^* = E / (1 - \nu^2)$  for plane strain

$E$  = Young's modulus

$\nu$  = Poisson's ratio

$\mu$  = shear modulus

### 11.3.3.1.2. Calculating the Stress-Intensity Factors

Mechanical APDL calculates the stress-intensity factors via interaction integral evaluation at the solution phase of the analysis, and then stores the values to the results file.

The **CINT** command's SIFS option initiates the stress-intensity factors calculations. Similar to the [J-integral calculation](#), **CINT** is also used to specify the parameters necessary for the calculation. For element type and material behavior support for **CINT**, see [Element Selection and Material Behavior](#) (p. 352).

Following is the general process for calculating the stress-intensity factors:

11.3.3.1.2.1. [Step 1: Initiate a New Stress-Intensity Factors Calculation](#)

11.3.3.1.2.2. [Step 2: Define Crack Information](#)

11.3.3.1.2.3. [Step 3: Specify the Number of Contours](#)

11.3.3.1.2.4. [Step 4: Define a Crack Symmetry Condition](#)

11.3.3.1.2.5. [Step 5: Specify Output Controls](#)

#### 11.3.3.1.2.1. Step 1: Initiate a New Stress-Intensity Factors Calculation

To start a stress-intensity factors calculation, issue the **CINT** command twice, as follows:

```
CINT,NEW,n
```

```
CINT,TYPE,SIFS
```

where *n* is the number identifying this stress-intensity factors calculation.

For example:

```
CINT,NEW,1      ! initiate a new calculation as #1
CINT,TYPE,SIFS ! specify stress-intensity factor calculations
```

#### 11.3.3.1.2.2. Step 2: Define Crack Information

Similar to the [J-integral calculation](#), the crack-tip node component and the crack-extension direction are both necessary for the stress-intensity factors calculation. Two methods, both involving the **CINT** command, are available for specifying the values:

- **Define the crack-tip node component and the crack-plane normal.**

This approach applies for both 2-D crack geometry and 3-D flat crack surfaces. It offers a simple way to define a 3-D stress-intensity-factors calculation, as you need only define the crack-tip (front) node component and the normal of the crack plane. Use this method when the crack plane is flat.

- **Define the crack-extension node component and crack-extension direction.**

This approach applies for 3-D curve crack planes, where a unique normal may not exist. However, you must define the crack-extension node component and the crack-extension direction at each crack-tip node location. Use this method when the crack plane is not flat, or when a set of nodes form the crack tip, as in the case of a collapsed crack-tip mesh.

The auxiliary crack-tip field is based on the crack-extension direction. To ensure the accuracy of the stress-intensity factors calculation, it is crucial that you correctly define the crack-extension definition.

### **Define the Crack-Tip Node Component and Crack-Plane Normal**

For 2-D crack geometry, define a crack-tip node component (usually a node located at the crack tip). You can also define a group of nodes around the crack tip, including the node at the crack tip. The program uses this group of nodes as the starting nodes to form the necessary information for the contour integration automatically.

For 3-D flat crack geometry, you must define a crack-tip node component that includes all of the nodes along the crack front. At each node location, however, only one node can exist. All nodes in the crack-tip node component must be connectable, and they must form a line based on the element connectivity associated with it. This line is the crack front. ANSYS uses it to automatically determine the elements for the contour integration. The procedure is similar to 2-D crack geometry, and is done through all the nodes along the crack front.

The command syntax is:

**CINT**,CTNC,*Par1*, *Par2*, *Par3*

where *Par1* is the crack-tip node component name, *Par2* defines the crack-extension direction calculation-assist node (any node on the open side of the crack), and *Par3* is crack front's end-node crack-extension direction-override flag.

The *Par2* and *Par3* values help to identify the crack-extension direction. Although the program automatically calculates the local coordinate system at crack tip for stress-intensity factors calculations, it is usually best to *Par2* to define a crack face node to help align the extension directions of the crack-tip nodes. By default, the program uses the external surface to determine the crack-extension direction and normal when the crack-tip node hits the free surface; however, you can use *Par3* to override this default with the calculated coordinate system.

After the crack-tip node component is defined, use the **CINT** command's NORM option to define the normal of the crack plane. The program automatically converts it into the crack-extension vector  $q$ , based on the element information. The crack-extension vector is taken along the perpendicular direction to the plane formed by the crack-plane normal and the tangent direction of the crack-tip node, and is normalized to a unit vector.

The command syntax is:

**CINT**,NORM,*Par1*, *Par2*

where *Par1* is the coordinate system number and *Par2* is the axis of coordinate system *Par1*.

### Example 1

```
! local coordinate system
LOCAL,11,0,,,
! select nodes located along the crack front and
! define it as crack front/tip node component
NSEL,S,LOC,X,Xctip
NSEL,R,LOC,Y,Yctip
CM,CRACK_TIP_NODE_CM
! Define a new stress-intensity factors calculation
CINT,NEW,1
CINT,TYPE,SIFS
CINT,CTNC,CRACK_TIP_NODE_CM
CINT,NORM,11,2
```

### Example 2

```
! select nodes located along the crack front and
! define it as crack front/tip node component
LSEL,,,
NSLL
CM,CRACK_FRONT_NODE_CM,NODE
CINT,NEW,1
CINT,TYPE,SIFS
CINT,CTNC,CRACK_FRONT_NODE_CM,
```

## Define the Crack-Extension Node Component and Crack-Extension Direction

Use this method when the crack plane is not flat, or when a set of nodes form the crack tip, as in the case of a collapsed crack-tip mesh:

### 1. Define a node component consisting of one or more nodes forming the crack tip.

The node component can have one or more nodes.

*Example: CINT, CENC, CMName*

### 2. Identify the crack-tip node separately if the node component has more than one node.

If a crack-tip node is not identified, the first node of the node component is taken to be the first node.

*Example: CINT, CENCP, CMName, node1*

### 3. Define the crack-extension direction.

Identify the local coordinate system associated with the crack under consideration.

Identify the local axis (for the above CS) along which the crack is supposed to extend.

*Example: CINT, CENC, CMName, node1, 11, 2*

Alternatively, define the crack-extension direction by directly specifying the global X Y and Z components of the crack-extension vector.

*Example: CINT, CENC, CMName, node1, , , compx, compy, compz*

Repeat this method for all node locations along the crack front. Although the program automatically calculates the local coordinate system at the crack tip to determine stress-intensity factors, it is usually best to use the NORM option to help align the calculated normals of the crack-tip nodes.

### Example 1

```
! local coordinate systems
local,11,0,,,,
local,12,0,,,,
...
local,n,0,,,,

NSEL,S,LOC,X,Xctip1
NSEL,R,LOC,Y,Yctip1
NSEL,R,LOC,Z,Zctip1
CM,CRACK_FRONT_NODE_CM1
NSEL,S,LOC,X,Xctip2
NSEL,R,LOC,Y,Yctip2
NSEL,R,LOC,Z,Zctip2
CM,CRACK_FRONT_NODE_CM2
...
NSEL,S,LOC,X,Xctipn
NSEL,R,LOC,Y,Yctipn
NSEL,R,LOC,Z,Zctipn
CM,CRACK_FRONT_NODE_CMn

CINT,NEW,1
CINT,TYPE,SIFS
CINT,CENC,CRACK_FRONT_NODE_CM1,,11,2
CINT,CENC,CRACK_FRONT_NODE_CM2,,11,2
...
CINT,CENC,CRACK_FRONT_NODE_CM2,,n,2
```

### Example 2

```
! Crack-extension node component and
! crack-extension direction specification using vectors
NSEL,S,LOC,X,Xctip1
NSEL,R,LOC,Y,Yctip1
NSEL,R,LOC,Z,Zctip1
CM,CRACK_FRONT_NODE_CM1
NSEL,S,LOC,X,Xctip2
NSEL,R,LOC,Y,Yctip2
NSEL,R,LOC,Z,Zctip2
CM,CRACK_FRONT_NODE_CM2
...
NSEL,S,LOC,X,Xctipn
NSEL,R,LOC,Y,Yctipn
NSEL,R,LOC,Z,Zctipn
CM,CRACK_FRONT_NODE_CMn

CINT,NEW,1
CINT,TYPE,SIFS
CINT,CENC,CRACK_FRONT_NODE_CM1,, ,Vx1,Vy1,Vz1
CINT,CENC,CRACK_FRONT_NODE_CM2,, ,Vx2,Vy2,Vz2
...
CINT,CENCOMP,CRACK_FRONT_NODE_CM2,, ,Vxn,Vyn,Vzn
```

## Local Crack-Tip Coordinate System

The auxiliary crack-tip field is based on the local crack-tip coordinate systems. To ensure the accuracy of the stress-intensity factors calculation, it is crucial to have a correct local crack-tip coordinate system in which the local x-axis is pointed to the crack extension, the local y-axis is pointed to the normal of the crack surfaces or edges, and the local z-axis pointed to the tangential direction of the crack front.

The local coordinate systems must be consistent across all nodes along the crack front. A set of inconsistent coordinate systems results in no path dependency of the calculated stress-intensity factors and

irregular behavior of the stress-intensity factor distribution along crack front. The program automatically calculate the local coordinate systems based on the input crack front nodes and the normal of the crack surface or extension directions; however, because there may be not enough information to determine a set of consistent coordinate systems, ANSYS, Inc. recommends using the **CINT** command's CTNC option to define a crack-face node to help identify the coordinate systems, or use the NORM option with the CENC option to define the coordinate systems.

### 11.3.3.1.2.3. Step 3: Specify the Number of Contours

The interaction integral, and thus the stress-intensity factors, have the same contour-independent behavior as the J-Integral. You can specify the number of contours for the interaction integral evaluation, as follows:

```
CINT,NCON,n
```

where *n* is the number of contours.

For 3-D crack geometry, every node along the crack front has the same number of contours.

#### Example

```
CINT,NEW,1
CINT,TYPE,SIFS
CINT,CTNC,CRACK_FRONT_NODE_SET,NODE
CINT,NCON,6
```

### 11.3.3.1.2.4. Step 4: Define a Crack Symmetry Condition

In general, there is no crack symmetry condition for a mixed-mode crack. When a crack symmetry condition is defined, however, stress-intensity factor  $K_{II}$  and  $K_{III}$  are both set to zero. The calculated interaction integral is timed by a factor of two and is used to calculate the stress-intensity factor  $K_I$ .

The command to define a symmetric condition is as follows:

```
CINT,SYMM,ON
```

#### Example

```
CINT,NEW,1
CINT,TYPE,SIFS
CINT,SYMM,ON ! crack #1 is a symmetrical crack
```

### 11.3.3.1.2.5. Step 5: Specify Output Controls

Similar to the J-Integral calculation, Mechanical APDL calculates the stress-intensity factors during the solution phase of the analysis and stores it to the results file for postprocessing. Stress-intensity factors output uses all the defaults from the **OUTRES** command. The command **OUTRES,ALL** includes CINT command results. However, you can issue an **OUTRES,CINT** command to control the specific output for stress-intensity factors results only.

#### Example

```
CINT,NEW,1
CINT,TYPE,SIFS
CINT,CTNC,CRACK_TIP_NODE_CM
CINT,SYMM,ON
CINT,NCON,5
OUTRES,CINT,10 ! output CINT results every 10 substeps
```



### 11.3.3.2. Calculating Stress-Intensity Factors via Displacement Extrapolation

The POST1 ppostprocessing **KCALC** command (**Main Menu**> **General Postproc**> **Nodal Calcs**> **Stress Int Factr**) calculates the mixed-mode stress-intensity factors  $K_I$ ,  $K_{II}$ , and  $K_{III}$ . The command is limited to linear elastic problems with a homogeneous, isotropic material near the crack region.

To calculate stress-intensity factors using the displacement extrapolation method, follow these steps within the POST1 postprocessor:

11.3.3.2.1. Step 1: Define a Local Crack-Tip or Crack-Front Coordinate System

11.3.3.2.2. Step 2: Define a Path Along the Crack Face

11.3.3.2.3. Step 3: Calculate  $K_I$ ,  $K_{II}$ , and  $K_{III}$

#### 11.3.3.2.1. Step 1: Define a Local Crack-Tip or Crack-Front Coordinate System

The X axis must be parallel to the crack face (perpendicular to the crack front in 3-D models) and the Y axis perpendicular to the crack face (as shown in [Figure 11.5: Crack Tip and Crack Front \(p. 347\)](#)).

This coordinate system must be the active model coordinate system (**CSYS**) and results coordinate system (**RSYS**) when **KCALC** executes.

**Command(s): LOCAL** (or **CLOCAL**, **CS**, **CSKP**, etc.)

**GUI: Utility Menu**> **WorkPlane**> **Local Coordinate Systems**> **Create Local CS**> **At Specified Loc**

#### 11.3.3.2.2. Step 2: Define a Path Along the Crack Face

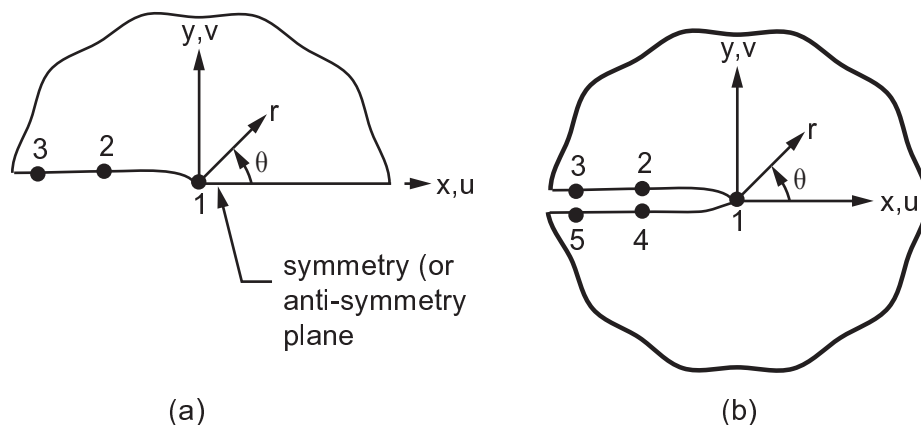
The first node on the path should be the crack-tip node. For a half-crack model, two additional nodes are required, both along the crack face. For a full-crack model, where both crack faces are included, four additional nodes are required: two along one crack face and two along the other.

**Command(s): PATH** (or **PPATH**)

**GUI: Main Menu**> **General Postproc**> **Path Operations**> **Define Path**

The following figure illustrates the two cases for a 2-D model.

**Figure 11.11: Typical Crack Face Path Definitions**



(a) half-crack model and (b) full-crack model

#### 11.3.3.2.3. Step 3: Calculate $K_I$ , $K_{II}$ , and $K_{III}$

The KPLAN field on the **KCALC** command specifies whether the model is plane-strain or plane-stress. Except for the analysis of thin plates, the asymptotic or near-crack-tip behavior of stress is usually thought to be that of plane strain. The KCSYM field specifies whether the model is a half-crack model

with symmetry boundary conditions, a half-crack model with antisymmetry boundary conditions, or a full-crack model.

**Command(s):**

*Content to be provided.*

**KCALC**

**GUI: Main Menu > General Postproc > Nodal Calcs > Stress Int Factr**

### 11.3.4. T-Stress Calculation

The following topics related to T-stress calculation are available:

11.3.4.1. T-Stress Interaction Integral Formulation

11.3.4.2. Element Selection and Material Behavior

11.3.4.3. Calculating the T-Stress

#### 11.3.4.1. T-Stress Interaction Integral Formulation

The T-stress parameter is calculated using an interaction integral similar to the one used for the [stress-intensity factors calculation](#). The auxiliary solution used is the solution of a line load of magnitude  $f$  applied along the crack front in the direction of the crack plane [12][13]:

$$\begin{aligned}\tilde{A}_{11}^L &= \frac{f}{\tilde{A}_r} \cos^3 \theta, & \tilde{A}_{22}^L &= \frac{f}{\tilde{A}_r} \cos \theta \sin^2 \theta, & \tilde{A}_{12}^L &= \frac{f}{\tilde{A}_r} \sin \theta \cos^2 \theta, \\ \tilde{A}_{33}^L &= \frac{f}{\tilde{A}_r} \nu \cos \theta, & \tilde{A}_{13}^L &= \tilde{A}_{23}^L = 0\end{aligned}$$

In practice, the magnitude of  $f$  is typically 1.

The T-stress itself is extracted from the interaction integral result  $I$  as follows:

$$T = \frac{E}{(1-\nu^2)} \left\{ \frac{I}{f} + \nu \epsilon_{33} \right\}$$

where:

$T$  = obtained T-stress value

$E$  = Young's modulus

$\nu$  = Poisson's ratio

$\epsilon_{33}$  = extensional strain at the crack front in the direction tangential to the front

#### 11.3.4.2. Element Selection and Material Behavior

T-stress evaluation (accessed via the **CINT** command) supports the following elements:

- PLANE182
- PLANE183
- SOLID185
- SOLID186

- [SOLID187](#)

T-stress evaluation supports the following material behavior:

- Linear isotropic elasticity
- Isotropic plasticity

Even if isotropic plasticity behavior is supported, stress-intensity factors and T-stress are strictly valid under the small-scale yielding assumption; therefore, the crack tip plastic zone must be small. In practice, the crack tip plastic zone must always be smaller than the last **CINT** integration contour so that the interaction integral is calculated on elements that still exhibit a linear elastic behavior.

T-stress evaluation supports 2-D plane strain and plane stress and 3-D analysis. Pressure on the crack faces, and axisymmetry, are not supported.

### 11.3.4.3. Calculating the T-Stress

To start a stress-intensity factors calculation, issue the **CINT** command twice, as follows:

```
CINT,NEW,n
```

```
CINT,TYPE,TSTRESS
```

where *n* is the T-stress calculation ID number.

All **CINT** command options available for stress-intensity factor calculation are also available for T-stress. For example, the following example input is a full setup of a T-stress evaluation using the node component `CRACK_TIP_NODE_CM` to define the crack front, a symmetry condition, and five integration contours:

```
CINT,NEW,1
CINT,TYPE,TSTRESS
CINT,CTNC,CRACK_TIP_NODE_CM
CINT,SYMM,ON
CINT,NCON,5
```

For more information about the calculation setup procedure, see [Calculating the Stress-Intensity Factors](#) (p. 364).

## 11.3.5. Material Force Calculation

The **material force** method determines the vectorial force-like quantities conjugated to the configurational change; that is, the method evaluates the material node point forces corresponding to the Eshelby stress and the material body forces.

The following topics concerning material force calculation are available:

[11.3.5.1. Understanding the Material Force Approach](#)

[11.3.5.2. Calculating Material Force](#)

### 11.3.5.1. Understanding the Material Force Approach

For a general 2-D problem (in the absence of body forces, thermal strain and dynamic loads), the nodal material forces are defined as:

$$F^{\text{mat}} = \prod_{ie=1}^{ne} \int_2 (\boldsymbol{\varepsilon} \cdot \nabla \mathbf{N}) w_{iw} A_{ie}$$

where:

$ne$  = number of elements to be integrated

$w_{iw}$  = weight function

$A_{ie}$  = area of the element represented by  $ie$

$\Sigma$  = Eshelby stress

$\mathbf{N}$  = shape functions

$\nabla \mathbf{N}$  = gradients of the shape functions

The Eshelby stress ( $\Sigma$ ) is introduced in small-strain elasticity as:

$$\mathcal{E}_{ij} = w'_{ij} - \nabla u_{ij} \tilde{A}_{ij}$$

where :

$\tilde{A}_{ij}$  = stress tensor

$\nabla u_{ij}$  = gradient of the displacement

$w$  = strain energy density

$'_{ij}$  = Kronecker delta

In finite strain elasticity, the Eshelby stress yields:

$$\mathcal{E}_{ij} = w_0'_{ij} - F_{ij} P_{ij}$$

where :

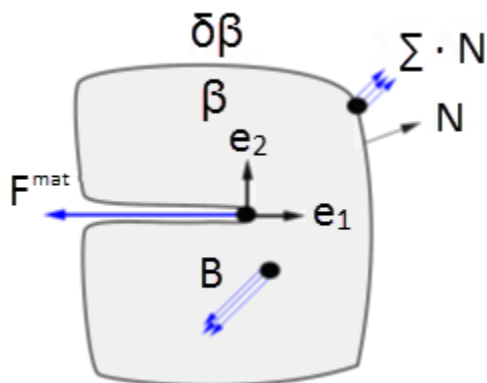
$P_{ij}$  = first Piola-Kirchoff stress tensor

$F_{ij}$  = deformation gradient

$w_0$  = material strain energy density

In the following figure, the directions of  $e_1$  and  $e_2$  correspond to the local coordinate system at the crack tip:

**Figure 11.12: Domain Enhancement via Nodal Vectors**



Here,  $e_1$  is the tangential component of the local coordinate system with respect to the crack surface, while  $e_2$  represents the normal component of the local coordinate system. The tangential component of the material force vector  $F^{\text{mat}}$ , which is in the direction of  $e_1$ , represents the scalar crack-driven force. In the numerical evaluation of material force, the material force is calculated based on the resultant of all the material force vectors in a user-defined domain  $\beta$  surrounding the crack tip.

If the plastic deformations exist in the structure, the material body forces acting on the domain are expressed as:

$$F^{\text{mat}} = \prod_{e=1}^{\text{numel}} \int_{\beta} (\Sigma \cdot \nabla N - NB) w_{iw} A_{ie}$$

where  $B$  is the material body forces:

$$B = B^p + B^t$$

Here,  $B^p$  is the material body forces via plasticity, whereas  $B^t$  is the material body forces via thermal stress. Then, the material body forces via plasticity are expressed as:

$$B^p = \tilde{A} \cdot \nabla \mu^p - q \nabla \pm$$

where :

$\pm$  = internal variable coefficient

$\mu^p$  = plastic strain

$q$  = derivation of the strain energy density in relation to the internal variable  $\pm$

If thermal strains exist in the structure, the nodal material body force vectors are:

$$B^t = \pm \tilde{A} \cdot \nabla_{,j}$$

where :

$\pm$  = thermal coefficient

$\nabla_{,j}$  = gradients of the temperature

For a 3-D problem, integral representation of the nodal material forces becomes a volume integration, which again is evaluated over a group of elements. The principal is similar to the 2-D problem. After nodal material forces are evaluated, however, they are divided by a distance quantity through the thickness.

### 11.3.5.1.1. Virtual Crack-Extension Nodes and Material Force Contours

Virtual crack-extension nodes are critical input data elements required for material force evaluation. The program uses virtual crack-extension node input to evaluate tangential (crack-driven force) and non-tangential components to the crack surface of the material force vectors. The crack-extension nodes are typically grouped together as crack-tip node components.

For a 2-D crack problem, the crack-tip node component typically contains one node, which is also the crack-tip node. The first contour for the area integration of the material force is evaluated over the elements associated with the crack-tip node component. The first contour gives nodal material force. The second contour for the area integration of the material force approach is evaluated over the elements

adjacent to the first contour of elements. The procedure is repeated for all contours. To ensure correct results, the elements for the contour integration should not reach the outer boundary of the model (with the exception of the crack surface).

For a 3-D crack problem, the crack-tip node component is comprised of the nodes along the crack front. It is not necessary for the crack-tip node component to be sorted. The 3-D material force contour uses a procedure similar to that of the 2-D contour.

### 11.3.5.1.2. Element Selection and Material Behavior

The following elements support material force evaluation (accessed via the **CINT** command):

- PLANE182
- PLANE183
- SOLID185
- SOLID186
- SOLID187

Material force evaluation supports the following material behavior:

- Linear isotropic elasticity
- Nonlinear isotropic elasticity
- Isotropic hardening plasticity
- Kinematic hardening plasticity

### 11.3.5.2. Calculating Material Force

The program calculates the material forces during the solution phase of the analysis after a substep has converged, then stores the values to the results file.

The **CINT** command initiates the material force calculation and specifies the necessary parameters.

Following is the general process for applying material force:

- 11.3.5.2.1. Step 1: Initiate a New Material Force Calculation
- 11.3.5.2.2. Step 2: Define Crack Information
- 11.3.5.2.3. Step 3: Specify the Number of Contours to Calculate
- 11.3.5.2.4. Step 4: Define a Crack Symmetry Condition
- 11.3.5.2.5. Step 5: Specify Output Controls

#### 11.3.5.2.1. Step 1: Initiate a New Material Force Calculation

To start a material force calculation, issue the **CINT** command twice, as follows:

```
CINT,NEW,n
```

```
CINT,TYPE,MFOR
```

where *n* is the material force calculation ID number.

**Example 11.1: Initiating a New Material Force Calculation**

```
CINT,NEW,1           ! initiate a new material force calculation as #1
CINT,TYPE,MFOR      ! set calculation type to material force
```

**11.3.5.2.2. Step 2: Define Crack Information**

The crack-tip node component and the local crack direction are both necessary for a material force calculation. Two methods using the **CINT** command are available for specifying the values:

- **Define the crack-tip node component and crack-plane normal.**

This approach applies to both 2-D crack geometry and 3-D flat crack surfaces. It offers a simple way to define a 3-D material force calculation, as you need only define the crack-tip (front) node component and the normal of the crack plane.

Use this method when the crack plane is flat.

- **Define the crack-extension node component and crack-extension direction.**

This approach applies to 3-D curve crack planes, where a unique normal may not exist. However, you must define the crack-extension node component and the crack-extension direction at each crack-tip node location. Use this method when the crack plane is not flat, or when a set of nodes form the crack tip, as in the case of a collapsed crack-tip mesh.

Use this method when the crack plane is not flat, or when a set of nodes form the crack tip, as in the case of a collapsed crack-tip mesh.

**11.3.5.2.2.1. Define the Crack-Tip Node Component and Crack-Plane Normal**

For 2-D crack geometry, you can define a crack-tip node component (usually a node located at the crack tip). You can also define a group of nodes around the crack tip, including the node at the crack tip. The program uses this group of nodes as the starting nodes to form the necessary information for the contour integration automatically.

For 3-D flat crack geometry, you must define a crack-tip node component that includes all of the nodes along the crack front. At each node location, however, only one node can exist. All nodes in the crack-tip node component must be connectable, and they must form a line based on the element connectivity associated with it. This line is the crack front. The program uses it to automatically determine the elements for the contour integration. The procedure is similar to 2-D crack geometry, and is performed through all nodes along the crack front.

The command syntax is:

```
CINT,CTNC,CompName
```

where *CompName* is the crack-tip node component name.

After defining the crack-tip node component, define the normal of the crack plane:

```
CINT,NORM,Par1,Par2
```

where *Par1* is the coordinate system number and *Par2* is the axis of that coordinate system.

The program automatically converts the crack plane normal into the local crack coordinate system, based on the element information. The local coordinate system is evaluated along the perpendicular

direction to the plane formed by the crack-plane normal and the tangent direction of the crack-tip node, and is normalized to a unit vector.

### Example 11.2: Defining the Crack-Tip Node Component and Crack-Plane Normal

```
! Local coordinate system
LOCAL,11,0,,,,
! Select nodes located along the crack front and define it as
! crack front/tip node component
NSEL,S,LOC,X,Xctip
NSEL,R,LOC,Y,Yctip
CM,CRACK_TIP_NODE_CM
! Define a new material force calculation
CINT,NEW,1
CINT,CTNC,CRACK_TIP_NODE_CM
CINT,NORM,11,2
```

### Example 11.3: Defining the Crack-Tip Node Component and Crack-Plane Normal

```
! Select nodes located along the crack front and define it as crack
! front/tip node component
LSEL,,,,
NSLL
CM,CRACK_FRONT_NODE_CM,NODE
CINT,NEW,1
CINT,CTNC,CRACK_FRONT_NODE_CM,
```

#### 11.3.5.2.2. Define the Crack-Extension Node Component and Crack-Extension Direction

Use this method when the crack plane is not flat, or when a set of nodes form the crack tip, as in the case of a collapsed crack-tip mesh:

1. **Define a node component consisting of one or more nodes forming the crack tip.** The node component can have one or more nodes.

Example: **CINT**,CENC,CompName

2. **Identify the crack-tip node separately if the node component has more than one node.** If a crack-tip node is not identified, then the first node of the node component is taken to be the first node.

Example: Example: **CINT**,CENC,CompName,NODE1

3. **Define the crack-extension direction.**

Identify the local coordinate system associated with the crack under consideration, and the local axis for that coordinate system along which the crack is supposed to extend.

Example: **CINT**,CENC,CompName,NODE1,11,2

Alternatively, define the crack-extension direction by directly specifying the global X, Y, and Z components of the crack-extension vector.

Example: **CINT**,CENC,CompName,NODE1,,, CompX, CompY, CompZ

Repeat this method for all node locations along the crack front.

### Example 11.4: Defining the Crack-Extension Node Component and Crack-Extension Direction

```
! Local coordinate systems
local,11,0,,,,
local,12,0,,,,
...
```



```

local,n,0,,,

NSEL,S,LOC,X,Xctip1
NSEL,R,LOC,Y,Yctip1
NSEL,R,LOC,Z,Zctip1
CM,CRACK_FRONT_NODE_CM1
NSEL,S,LOC,X,Xctip2
NSEL,R,LOC,Y,Yctip2
NSEL,R,LOC,Z,Zctip2
CM,CRACK_FRONT_NODE_CM2
...
NSEL,S,LOC,X,Xctipn
NSEL,R,LOC,Y,Yctipn
NSEL,R,LOC,Z,Zctipn
CM,CRACK_FRONT_NODE_CMn

CINT,NEW,1
CINT,CENC,CRACK_FRONT_NODE_CM1,,11,2
CINT,CENC,CRACK_FRONT_NODE_CM2,,11,2
...
CINT,CENC,CRACK_FRONT_NODE_CM2,,n,2

```

### Example 11.5: Defining the Crack-Extension Node Component and Crack-Extension Direction

```

! Crack-extension node component and
! crack-extension direction specification using vectors
NSEL,S,LOC,X,Xctip1
NSEL,R,LOC,Y,Yctip1
NSEL,R,LOC,Z,Zctip1
CM,CRACK_FRONT_NODE_CM1
NSEL,S,LOC,X,Xctip2
NSEL,R,LOC,Y,Yctip2
NSEL,R,LOC,Z,Zctip2
CM,CRACK_FRONT_NODE_CM2
...
NSEL,S,LOC,X,Xctipn
NSEL,R,LOC,Y,Yctipn
NSEL,R,LOC,Z,Zctipn
CM,CRACK_FRONT_NODE_CMn

CINT,NEW,1
CINT,CENC,CRACK_FRONT_NODE_CM1,, , , ,Vx1,Vy1,Vz1
CINT,CENC,CRACK_FRONT_NODE_CM2,, , , ,Vx2,Vy2,Vz2
...
CINT,CENC,CRACK_FRONT_NODE_CM2,, , , ,Vxn,Vyn,Vzn

```

#### 11.3.5.2.3. Step 3: Specify the Number of Contours to Calculate

Specify the number of contours for the material force evaluation:

```
CINT,NCON,n
```

where  $n$  is the number of contours.

For 3-D crack geometry, every node along the crack front has the same number of contours.

### Example 11.6: Specifying the Number of Contours to Calculate

```

CINT,NEW,1
CINT,CTNC,CRACK_FRONT_NODE_SET,NODE
CINT,NCON,6

```

#### 11.3.5.2.4. Step 4: Define a Crack Symmetry Condition

If the crack is located along a symmetry plane and only a half model is created, define a symmetric condition so that the program can account for it:

**CINT**,SYMM,ON**Example 11.7: Defining a Crack Symmetry Condition**

```
CINT,NEW,1
CINT,SYMM,ON      ! crack 1 is a symmetrical crack
```

**11.3.5.2.5. Step 5: Specify Output Controls**

The program calculates the nodal material forces at the solution level and stores it to the results file for postprocessing.

The output for the material force approach uses all **OUTRES** command defaults. The command **OUTRES,ALL** includes **CINT** command results. However, you can issue an **OUTRES,CINT** command to control the specific output for material force results only.

**Example 11.8: Specifying Output Controls**

```
CINT,NEW,1
CINT,CTNC,CRACK_TIP_NODE_CM
CINT,SYMM,ON
CINT,NCON,5
OUTRES,CINT,10      ! output material forces every 10 substeps
```

**11.4. Learning More About Fracture Mechanics**

A considerable body of literature exists concerning fracture mechanics. The following list of resources offers a wealth of information but is by no means exhaustive:

1. Anderson, T. L. *Fracture Mechanics -- Fundamentals and Applications*. 2nd ed., Boca Raton: CRC, 1994.
2. Rice, J. R. "A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notched and Cracks." *Journal of Applied Mathematics*. 35 (1968): 379-386.
3. Hutchinson, J. W. "Singular Behavior at the End of a Tensile Crack in a Hardening Material." *Journal of the Mechanics and Physics of Solids*. 16, 1 (1968): 13-31.
4. Rice, J. R. and G. F. Rosengren. "Plane Strain Deformation Near a Crack Tip in a Power Law Hardening Material." *Journal of the Mechanics and Physics of Solids*. 16 (1968): 1-12.
5. Shih, C. F., B. Moran, and T. Nakamura. "Energy Release Rate Along a Three-Dimensional Crack Front in a Thermally Stressed Body." *International Journal of Fracture*. 30, 2 (1986): 79-102.
6. Rybicki, E.F. and M.F. Kanninen. "A Finite Element Calculation of Stress-Intensity Factors by a Modified Crack Closure Integral." *Engineering Fracture Mechanics*. 9 (1977): 931-938.
7. Benzeggagh, M. and M. Kenane. "Measurement of Mixed-Mode Delamination Fracture Toughness of Unidirectional Glass/Epoxy Composites with Mixed-Mode Bending Apparatus." *Composite Science and Technology*. 56 (1996): 439-449.
8. Reeder, J.R. "A Bilinear Failure Criterion for Mixed-Mode Delamination in Composite Materials." *Testing and Design*. ASTM STP 1206. 11 (1993): 303-322.
9. Reeder, J., S. Kyongchan, P. B. Chunchu, and D. R.. Ambur. *Postbuckling and Growth of Delaminations in Composite Plates Subjected to Axial Compression*. 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. Denver. 1746 (2002): 10.

10. Wu, E. M. and R. C. Reuter Jr. *Crack Extension in Fiberglass Reinforced Plastics*. University of Illinois. T/AM Report. 275 (1965).
11. Krueger, R. *The Virtual Crack Closure Technique: History, Approach and Applications*. ICASE Report No. 2002-10, 2002.
12. Toshio, N. and D. M. Parks. "Determination of Elastic T-Stress Along Three-Dimensional Crack Fronts Using an Interaction Integral." *International Journal of Solids and Structures* . 29, 13 (1992): 1597-1611.
13. Zhao, L. G., J. Tong, and J. Byrne. "Stress-Intensity Factor K and the Elastic T-Stress for Corner Cracks." *International Journal of Fracture*. 109, 2 (2001): 209-225.
14. Carlsson, A. J. "Influence of Non-Singular Stress Terms and Specimen Geometry on Small-Scale Yielding at Crack Tips in Elastic-Plastic Materials." *Journal of the Mechanics and Physics of Solids*. 21, 4 (1973): 263-277.
15. Hackett, E. M. and K. H. Schwalbe. "Constraint Effects in Fracture." *ASTM International -- Committee E08 on Fatigue and Fracture*. 1171 (1993).
16. Cotterell, B. and J. Rice. "Slightly Curved or Kinked Cracks." *International Journal of Fracture*. 16, 2 (1980): 155-169.
17. Eshelby, J. D. "The Determination of the Elastic Field of an Ellipsoidal Inclusion and Related Problems." *Proceedings of the Royal Society of London, Series A. Mathematical and Physical Sciences*. 241, 1226 (1957): 376-396.
18. Braun, M. (1997). "Configurational forces induced by finite-element discretization." *Proceedings of the Estonian Academy of Sciences, Physics and Mathematics*. 35 (1997): 379-386.
19. Näser, B., M. Kaliske and R. Müller. "Material Forces for Inelastic Models at Large Strains: Application to Fracture Mechanics." *Computational Mechanics*. 40, 6 (2007): 1005-1013.



---

## Chapter 12: Interface Delamination and Failure Simulation

---

An interface exists anywhere two materials are joined together. The interface between the layers of a composite structure is of special interest, because when this type of structure is subjected to certain types of external loading, the failure process (delamination) takes on a unique character. Interface delamination is traditionally simulated using fracture mechanics methods, such as nodal release technique.

Because cracks generally grow along the interfaces, VCCT-based crack growth simulation has become a widely used technique for simulation of interface delamination of laminate composite. This technique is also well suited for modeling the fracture process in a homogeneous medium, as fracture can be considered a separation process between two surfaces. For more information, see [VCCT-Based Crack Growth Simulation](#) (p. 381).

Alternatively, you can use the cohesive zone model to simulate interface delamination and other fracture phenomena. This approach introduces failure mechanisms by using the hardening-softening relationships between the separations and incorporating the corresponding tractions across the interface. Similarly, this technique is also well suited for modeling the fracture process in a homogeneous medium. An interface delamination and failure simulation is performed by first separating the model into two components or groups of elements, then defining a cohesive zone between the two groups. Two options are available for modeling the interface:

- [Modeling Interface Delamination with Interface Elements](#) (p. 397)
- [Modeling Interface Delamination with Contact Elements](#) (p. 404)

### 12.1. VCCT-Based Crack Growth Simulation

The virtual crack closure technique (VCCT) was initially developed to calculate the energy-release rate of a cracked body [6]. It has since been widely used in the interfacial crack growth simulation of laminate composites, with the assumption that crack growth is always along a predefined path, specifically the interfaces [7][8][9][10].

VCCT-based crack growth simulation is available with current-technology linear elements such as [PLANE182](#) and [SOLID185](#).

A VCCT-based crack growth simulation involves the following assumptions:

- Crack growth occurs along a predefined crack path.
- The path is defined via interface elements.
- The analysis is quasi-static and does not account for transient effects.
- The material is linear elastic and can be isotropic, orthotropic or anisotropic.

The crack can be located in a material or along the interface of the two materials. The fracture criteria are based on energy-release rates calculated using VCCT. Several fracture criteria are available, including a user-defined option. Multiple cracks can be defined in an analysis.

A VCCT-based crack growth simulation uses:

- Interface elements [INTER202](#) (2-D) and [INTER205](#) (3-D).
- The **CINT** command to calculate the [energy-release rate](#).
- The **CGROW** command to define the crack growth set, fracture criterion, crack growth path, and solution control parameters.

### 12.1.1. VCCT Crack Growth Simulation Process

A VCCT-based crack growth simulation is assumed to be quasi-static. Following is the general process for performing the simulation:

12.1.1.1. Step 1. Create a Finite Element Model with a Predefined Crack Path

12.1.1.2. Step 2. Perform the Energy-Release Rate Calculation

12.1.1.3. Step 3. Perform the Crack Growth Calculation

12.1.1.4. Example: Crack Growth Set Definition

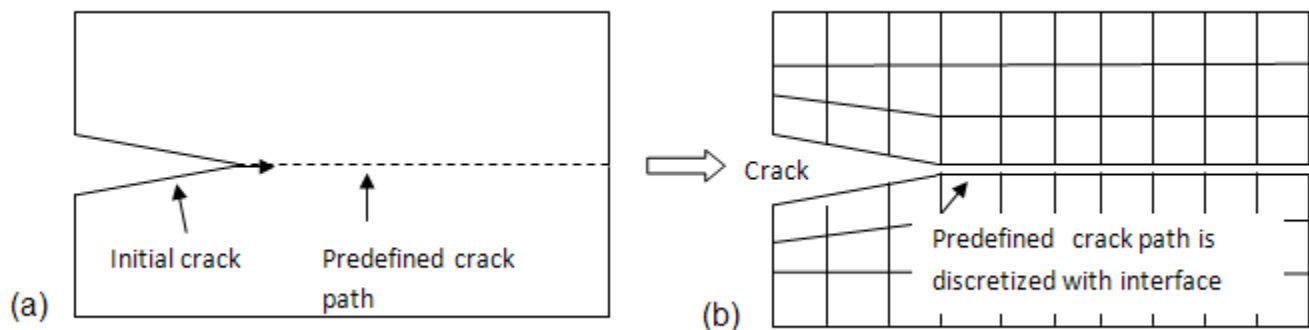
Crack growth simulation is a nonlinear structural analysis. The analysis details presented here emphasize features specific to crack growth.

#### 12.1.1.1. Step 1. Create a Finite Element Model with a Predefined Crack Path

Standard nonlinear solution procedures apply for creating a finite element model with proper solution-control settings, loadings and boundary conditions.

The [predefined crack path](#) is discretized with interface elements and grouped as an element component, as shown in the following figure:

**Figure 12.1: Crack Path Discretized with Interface Elements**



The interface elements can be meshed via **CZMESH** or meshed by a third-party tool that generates interface elements.

The element MPC constraint option ( $KEYOPT(2) = 1$ ) bonds the potential crack faces together before cracks begin to grow. The MPC constraints are subsequently released when the fracture criterion is met, thus growing the cracks.

In a 2-D problem, one interface element behind the crack tip may open if it meets the fracture criterion at a given substep. In a 3-D problem, *all* interface elements behind the crack front may open if they meet the fracture criterion.

Differences in the size of the elements ahead of and behind the crack tip/front affect the accuracy of the energy-release rate calculation. While the program uses a correction algorithm, it may be inadequate to produce an accurate solution. Instead, use equal sized meshes for elements along the predefined crack path. For more information, see [Numerical Evaluation of Fracture Mechanics Parameters \(p. 350\)](#).

### 12.1.1.2. Step 2. Perform the Energy-Release Rate Calculation

For VCCT-based crack-growth simulation, it is necessary to perform the [energy-release rate calculation](#) first.

To calculate the energy-release rates, issue the **CINT**,TYPE,VCCT command. Issue subsequent **CINT** commands to specify other options such as the crack tip node component and crack plane/edge normal.

The VCCT calculation uses the following assumptions:

- The strain energy released when a crack advances by a small amount is the same as the energy required to close the crack by the same amount.
- The crack tip field/deformation at the crack tip/front location is similar to when the crack extends by a small amount.

The assumptions do not apply when crack growth approaches the boundary or when the two cracks approach each other; therefore, use the VCCT calculation with care and examine the analysis results.

For further information, see [VCCT Energy-Release Rate Calculation \(p. 356\)](#).

### 12.1.1.3. Step 3. Perform the Crack Growth Calculation

The crack growth calculation occurs in the solution phase after stress calculation. To perform the crack growth calculation, you must define a crack growth set, then specify the crack path, fracture criterion, and crack growth solution controls. The solution command **CGROW** defines all necessary crack growth calculation parameters.

Perform the crack growth calculation as follows:

12.1.1.3.1. Step 3a. Initiate the Crack Growth Set

12.1.1.3.2. Step 3b. Specify the Crack Path

12.1.1.3.3. Step 3c. Specify the Crack-Calculation ID and Fracture Criterion

12.1.1.3.4. Step 3d: Specify Solution Controls for Crack Growth

#### 12.1.1.3.1. Step 3a. Initiate the Crack Growth Set

To define a [crack growth set](#), issue the **CGROW**,NEW,*n* command, where *n* is the crack growth set number.

#### 12.1.1.3.2. Step 3b. Specify the Crack Path

To define the crack path, issue the **CGROW**,CPATH,*cmname* command, where *cmname* is the component name for the interface elements.

#### 12.1.1.3.3. Step 3c. Specify the Crack-Calculation ID and Fracture Criterion

Specify the crack-calculation ID via the **CGROW**,CID,*n* command, where *n* is the crack-calculation (**CINT**) ID for energy-release rate calculation with VCCT. (The **CINT** command defines parameters associated with fracture parameter calculations.)

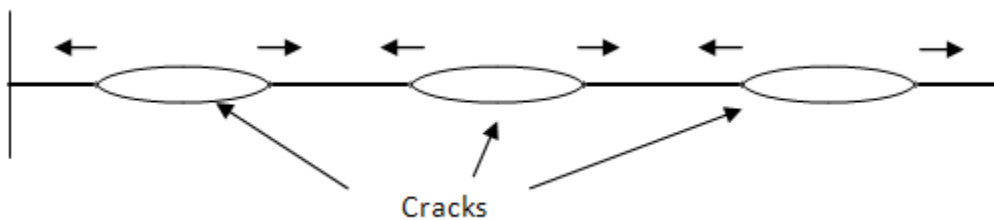
For a simple fracture criterion such as the [critical energy-release rate](#), you can specify it by issuing the **CGROW**,FCOPTION,GC,value command, where *value* is the critical energy-release rate.

For a more complex fracture criterion, you can specify the fracture criteria via a material data table. Issue the **CGROW**,FCOPTION,MTAB,matid command, where *matid* is the material ID for the material table. Several fracture criterion options are available (such as linear, bilinear, B-K, modified B-K, Power Law, and user-defined).

For more information, see the **TB**,CGCR command and [Fracture Criteria](#) (p. 386).

For each crack growth set, you can specify only one fracture criterion, and one element component for crack growth. You can define multicrack growth sets with different cracks and fracture criteria. Multiple cracks can grow simultaneously and independently from each other. Cracks can merge to a single crack when they are on the same interface, as shown in the following figure:

**Figure 12.2: Crack Growth and Merging**



You can also define the same crack with different fracture criteria in a separate crack growth set. The cracks can grow based on different criteria (according to which criterion is met), and are independent from each other. This technique is useful for comparing failure mechanisms.

#### 12.1.1.3.4. Step 3d: Specify Solution Controls for Crack Growth

Issue the **CGROW** command to specify solution controls, as follows:

To specify this solution control...	Issue this <b>CGROW</b> command:
Fracture criterion ratio ( $f_c$ )	<b>CGROW</b> ,FCRAT,value, where <i>value</i> is the ratio
Initial time step when crack growth initiates	<b>CGROW</b> ,DTIME,value, where <i>value</i> is initial time step  To avoid over-predicting the load-carrying capacity, specify a small initial time step.
Minimum time step for subsequent crack growth	<b>CGROW</b> ,DTMIN,value, where <i>value</i> is the minimum time step size
Maximum time step for subsequent crack growth	<b>CGROW</b> ,DTMAX,value, where <i>value</i> is the maximum time step size
Maximum crack extension allowed at any crack front nodes	<b>CGROW</b> ,STOP,CEMX,value,



To specify this solution control...	Issue this <b>CGROW</b> command:
	<p>where <i>value</i> is the maximum crack extension</p> <p>Because crack growth simulation can be time-consuming, use this command to stop the analysis when the specified crack extension of interest has been reached.</p>

When a crack extends rapidly (for example, in cases of unstable crack growth), use smaller DTMAX and DTMIN values to allow time for load rebalancing. When a crack is not growing, the specified time-stepping controls are ignored and the solution adheres to standard time-stepping control.

### 12.1.1.4. Example: Crack Growth Set Definition

The following input example defines a **crack growth set**:

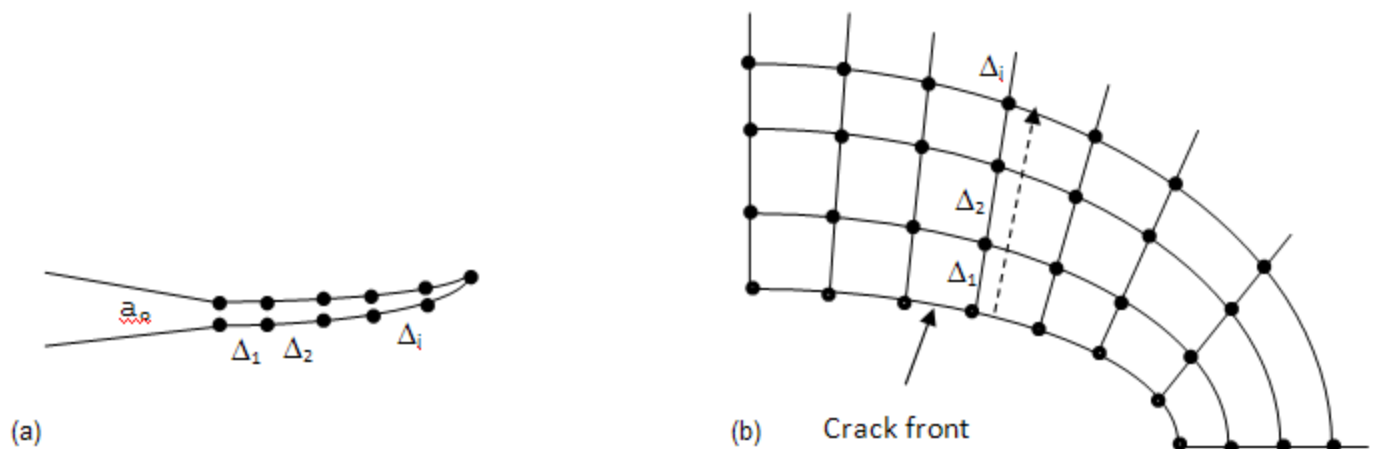
```
CGROW,NEW,1
CGROW,CPATH,cpath1
CGROW,FCOPTION,MTAB,5
CGROW,DTIME,1.0e-4
CGROW,DTMIN,1.0e-5
CGROW,DTMAX,2.0e-3
...
```

### 12.1.2. Crack Extension

In a crack growth simulation, a quantity of interest is the amount of crack extension. VCCT measures the crack extension based on the length of the interface elements that have opened, as expressed by the following equation and in the subsequent figure:

$$\Delta a = \sum \Delta_i$$

**Figure 12.3: 2-D and 3-D Crack Extension**



For 2-D crack problems, the crack extension is the summation of length of interface elements that are currently open (a). For 3-D problems, the crack extension is measured at each crack front node and is the summation of the length of the interface element edges that follow the crack extension direction (b).

Crack extension  $\Delta a$  is available as CEXT as part of the crack solution variable associated with the crack-calculation ID, and can be postprocessed similar to energy release rates via POST1 and POST26 postprocessing commands (such **PRCINT**, **PLCINT**, and **CISOL**).

### 12.1.3. Fracture Criteria

To model the crack growth, it is necessary to define a fracture criterion for crack onset and the subsequent crack growth. For linear elastic fracture mechanics (LEFM) applications, the fracture criterion is generally assumed to be a function of Mode I (GI), Mode II (GII), and Mode III (GIII) critical energy-release rates, expressed as:

$$f = f(G_I^C, G_{II}^C, G_{III}^C, G_I, G_{II}, G_{III}, \dots)$$

Other parameters may be necessary for some models.

Fracture occurs when the fracture criterion index is met, expressed as:

$$f \geq f_c$$

where  $f_c$  is the fracture criterion ratio. The recommended ratio is 0.95 to 1.05. The default is 1.0.

The following fracture criteria are available:

12.1.3.1. Critical Energy-Release Rate Criterion

12.1.3.2. Linear Fracture Criterion

12.1.3.3. Bilinear Fracture Criterion

12.1.3.4. B-K Fracture Criterion

12.1.3.5. Modified B-K Fracture Criterion

12.1.3.6. Power Law Fracture Criterion

12.1.3.7. User-Defined Fracture Criterion

The user-defined option requires a [subroutine that you provide](#) to define your own fracture criterion.

To initiate a fracture criterion table without the critical energy-release rate criterion, issue the **TB,CGCR** command.

#### 12.1.3.1. Critical Energy-Release Rate Criterion

The critical energy-release rate criterion uses total energy-release rate ( $G_T$ ) as fracture criterion. The total energy-release rate is summation of the Mode I (GI), Mode II (GII), and Mode III (GIII) energy-release rates, expressed as:

$$f = \frac{G_T}{G_T^C} \quad G_T = G_I + G_{II} + G_{III}$$

where  $G_T^C$  is the critical energy-release rate.

For Mode I fracture, the fracture criterion reduces to:

$$f = \frac{G_I}{G_I^C}$$

The critical energy-release rate option is the simplest fracture criterion and is suitable for general 2-D and 3-D crack growth simulation.

### Example 12.1: Critical Energy Release Rate Input

```
gtcval=10.0
CGROW,FCOPTION,GTC,gtcval
```

#### 12.1.3.2. Linear Fracture Criterion

The linear option assumes that the fracture criterion is a linear function of the Mode I ( $G_I$ ), Mode II ( $G_{II}$ ), and Mode III ( $G_{III}$ ) energy-release rates, expressed as:

$$f = \frac{G_I}{G_I^C} + \frac{G_{II}}{G_{II}^C} + \frac{G_{III}}{G_{III}^C}$$

where  $G_I^C$ ,  $G_{II}^C$ , and  $G_{III}^C$  are the Mode I, Mode II, and Mode III critical energy-release rates, respectively. The three values are input via the **TBDATA** command, as follows:

Constant	TBDATA Input	Comments
$G_I^C$	C1	Critical Mode I energy-release rate, $G_I^C \geq 0$
$G_{II}^C$	C2	Critical Mode II energy-release rate, $G_{II}^C \geq 0$
$G_{III}^C$	C3	Critical Mode III energy-release rate, $G_{III}^C \geq 0$

### Example 12.2: Linear Criterion Input

```
g1c=10.0
g2c=20.0
g3c=25.0
TB,CGCR,1,, ,LINEAR
TBDATA,1,g1c,g2c,g3c
```

The three constants cannot all be zero. If a constant is set to zero, the corresponding term is ignored.

When all three critical energy-release rates are equal, the linear fracture criterion reduces to the [critical energy-release rate criterion](#).

The linear fracture criterion is suitable for 3-D mixed-mode fracture simulation where distinct Mode I, Mode II, and Mode III critical energy-release rates exist.

### 12.1.3.3. Bilinear Fracture Criterion

The bilinear fracture option [8] assumes that the fracture criterion is a linear function of the Mode I ( $G_I$ ) and Mode II ( $G_{II}$ ) energy-release rates, expressed as:

$$f = \begin{cases} \frac{G_I - \xi G_{II}}{G_I^C} & \text{for } \frac{G_{II}}{G_I} < \frac{\frac{1}{\zeta} G_I^C + G_{II}^C}{G_I^C + \xi G_{II}^C} \\ \frac{\zeta G_{II} - G_I}{\zeta G_{II}^C} & \text{for } \frac{G_{II}}{G_I} > \frac{\frac{1}{\zeta} G_I^C + G_{II}^C}{G_I^C + \xi G_{II}^C} \end{cases}$$

where  $G_I^C$  and  $G_{II}^C$  are the Mode I and Mode II critical energy-release rates, respectively, and  $\xi$  and  $\zeta$  are the two material constants. All four values can be defined as a function of temperature and are input via the **TBDATA** command, as follows:

Constant	TBDATA Input	Comments
$G_I^C$	C1	Critical Mode I energy-release rate, $G_I^C > 0$
$G_{II}^C$	C2	Critical Mode II energy-release rate, $G_{II}^C > 0$
$\xi$	C3	$\xi > 0$
$\zeta$	C4	$\zeta > 0$

#### Example 12.3: Bilinear Criterion Input

```
g1c=10.0
g2c=20.0
x=2
y=2

TB,CGCR,1,, ,BILINEAR
TBDATA,1,g1c,g2c,x,y
```

The bilinear fracture criterion is suitable for 2-D mixed-mode fracture simulation.

### 12.1.3.4. B-K Fracture Criterion

The B-K [7] option is expressed as:

$$f = \frac{G_T}{G_I^C + (G_{II}^C - G_I^C) \left( \frac{G_{II} + G_{III}}{G_T} \right)^\eta}$$

where  $G_I^C$  and  $G_{II}^C$  are the Mode I and Mode II critical energy-release rates, respectively, and  $\eta$  is the material constant. All three values can be defined as a function of temperature and are input via the **TBDATA** command, as follows:

Constant	TBDATA Input	Comments
$G_I^C$	C1	Critical Mode I energy-release rate, $G_I^C > 0$
$G_{II}^C$	C2	Critical Mode II energy-release rate, $G_{II}^C > 0$
$\eta$	C3	$\eta > 0$

The B-K criterion is intended for composite interfacial fracture and is suitable for 3-D mixed-mode fracture simulation.

#### Example 12.4: B-K Criterion Input

```
g1c=10.0
g2c=20.0
h=2
```

```
TB,CGCR,1,,BK
TBDATA,1,g1c,g2c,h
```

#### 12.1.3.5. Modified B-K Fracture Criterion

The modified B-K (or Reeder) [9] option, is expressed as:

$$f = \frac{G_T}{G_I^C + \left( \left( G_{II}^C - G_I^C \right) \frac{G_{II}}{G_T} + \left( G_{III}^C - G_I^C \right) \frac{G_{III}}{G_T} \right) \left( \frac{G_{II} + G_{III}}{G_T} \right)^{\eta-1}}$$

where  $G_I^C$ ,  $G_{II}^C$ , and  $G_{III}^C$  are Mode I, Mode II, and Mode III critical energy-release rates, respectively, and  $\eta$  is the material constant. All four values can be defined as a function of temperature and are input via the **TBDATA** command, as follows:

Constant	TBDATA Input	Comments
$G_I^C$	C1	Critical Mode I energy-release rate, $G_I^C > 0$
$G_{II}^C$	C2	Critical Mode II energy-release rate, $G_{II}^C > 0$
$G_{III}^C$	C3	Critical Mode III energy-release rate, $G_{III}^C > 0$
$\eta$	C4	$\eta > 0$

When  $G_{II}^C = G_{III}^C$ , the modified B-K criterion reduces to the **B-K criterion**.

The modified B-K criterion is intended for composite interfacial fracture to account for distinct Mode II and Mode III critical energy-release rates, and is suitable for 3-D mixed-mode fracture simulation.

### Example 12.5: Modified B-K Criterion Input

```
g1c=10.0
g2c=20.0
g3c=25.0
h=2

TB,CGCR,1,, ,MBK
TBDATA,1,g1c,g2c,g3c,h
```

#### 12.1.3.6. Power Law Fracture Criterion

The power law [10] option assumes that the fracture criterion is a power function of the Mode I ( $G_I$ ), Mode II ( $G_{II}$ ), and Mode III ( $G_{III}$ ) energy-release rates, expressed as:

$$f = \left( \frac{G_I}{G_I^C} \right)^{n_1} + \left( \frac{G_{II}}{G_{II}^C} \right)^{n_2} + \left( \frac{G_{III}}{G_{III}^C} \right)^{n_3}$$

where  $G_I^C$ ,  $G_{II}^C$ , and  $G_{III}^C$  are Mode I, Mode II, and Mode III critical energy-release rates, respectively, and  $n_1$ ,  $n_2$ , and  $n_3$  are power exponents and are also constants. All six values can be defined as a function of temperature and are input via the **TBDATA** command, as follows:

Constant	TBDATA Input	Comments
$G_I^C$	C1	Critical Mode I energy-release rate, $G_I^C$ > 0
$G_{II}^C$	C2	Critical Mode II energy-release rate, $G_{II}^C$ > 0
$G_{III}^C$	C3	Critical Mode III energy-release rate, $G_{III}^C$ > 0
$n_1$	C4	$n_1 > 0$
$n_2$	C5	$n_2 > 0$
$n_3$	C6	$n_3 > 0$

The three critical energy-release rates cannot all be zero. If a constant is set to zero, the corresponding term is ignored.

When power exponents  $n_1$ ,  $n_2$ , and  $n_3$  are set to 1, the power law criterion is reduced to the [linear fracture criterion](#).

The power law fracture criterion is suitable for 3-D mixed-mode fracture simulation where distinct Mode I, Mode II, and Mode III critical energy-release rates exist.

**Example 12.6: Power Law Criterion Input**

```

g1c=10.0
g2c=20.0
g3c=25.0
n1=2
n2=2
n3=3

TB,CGCR,1,,POWERLAW
TBDATA,1,g1c,g2c,g3c,n1,n2,n3

```

**12.1.3.7. User-Defined Fracture Criterion**

A custom fracture criterion that you define is expressed as:

$$f = f(G_I, G_{II}, G_{III}, G_T, \dots)$$

where the fracture criterion is a function of the Mode I ( $G_I$ ), Mode II ( $G_{II}$ ), and Mode III ( $G_{III}$ ) energy-release rates, and the material constant(s). All values are input via the **TBDATA** command.

A subroutine that you provide is necessary. For more information, see the *Programmer's Reference*.

Following is an example subroutine defining a linear fracture criterion:

```

*deck,user_cgfcrit      optimize
      SUBROUTINE user_cgfcrit (cgi,  cid,  kct,
&                               nprop, prop, fcscl,
&                               var1, var2, var3, var4)
c*****
c
c *** primary function:
c       compute fracture criterion for crack growth
c       user fracture criterion example
c
c *** notice: this routine contains sasi confidential information ***
c
#include "impcom.inc"
#include "ansysdef.inc"
c
c   input arguments
c   =====
c   cgi      (int,sc  , in)   CGROW set id
c   cid      (int,sc  , in)   CINT ID to be used
c   kct      (int,sc  , in)   Current crack tip node
c   nprop    (int,sc  , in)   number of properties
c   prop     (dp ,ar(*), in)  property array
c
c   Output arguments
c   =====
c   fcscl    (dp, sc  , ou)   fracture criterion
c                                       a return value of one or bigger
c                                       indicates fracture
c
c   Misc. arguments
c   =====
c   var1     ( , , , )      not used
c   var2     ( , , , )      not used
c   var3     ( , , , )      not used
c   var4     ( , , , )      not used
c
c*****
c
c *** subroutines/function
c ***   get_cgfparr      : API to access fracture data
c ***   wrinqr          : ansys standard io function
c ***

```

```

        external      get_cgfparr
        external      wrinqr
        integer       wrinqr

c *** argument
c
        INTEGER      cgi, cid, kct, nprop
        double precision fcscl,
        &              var1, var2, var3, var4
        double precision prop(nprop)
c
c *** local variable
c
        integer      debugflag, iott
        integer      nn
        double precision g1c, g2c, g3c, g1, g2, g3
        double precision gs(4),da(1)
c
c *** local parameters
        DOUBLE PRECISION ZERO, ONE
        parameter    (ZERO = 0.0d0,
        &              ONE  = 1.0d0)
c
c*****
c *** initialization
        fcscl = ZERO

c *** retrieve energy-release rates
c *** for crack cid and crack tip node kct
c *** gs(1:3) will be returned as G1, G2, G3

c *** get energy-release rates
        nn = 3
        gs(1:nn) = ZERO
        call get_cgfparr ('GS ', cid, kct, 0, nn, gs(1))

c *** get crack extension
        nn = 1
        da(1) = ZERO
        call get_cgfparr ('DA ', cid, kct, 0, nn, da(1))

c *** energy-release rates
        g1 =abs(gs(1))
        g2 =abs(gs(2))
        g3 =abs(gs(3))

c *** input property from TBDATA,1,c1,c2,c3
        g1c = prop(1)
        g2c = prop(2)
        g3c = prop(3)

c *** linear fracture criterion
        fcscl = ZERO
        if (g1c .gt. TINY) fcscl = fcscl + g1/g1c
        if (g2c .gt. TINY) fcscl = fcscl + g2/g2c
        if (g3c .gt. TINY) fcscl = fcscl + g3/g3c

c *** user debug output
        debugflag = 1
        if (debugflag .gt. 0) then
            iott = wrinqr (WR_OUTPUT)
            write(iott, 1000) cgi, cid, kct, da(1), fcscl, gs(1:3)
1000 format (5x,'user fracture criterion:'/
        & 5x, 'crack growth set ID      =',i5/
        & 5x, 'crack ID                  =',i5/
        & 5x, 'crack tip node             =',i5/
        & 5x, 'crack extension            =',g11.5/
        & 5x, 'calculated fracture parameter =',g11.5/
        & 5x, 'energy-release rates Gs(1:3) =',3g12.5)
        end if

```



```

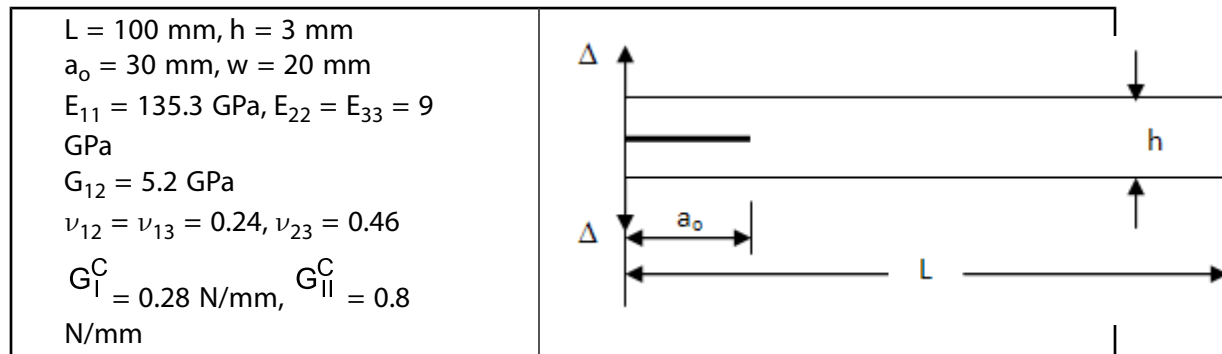
return
end

```

### 12.1.4. Example Crack Growth Simulation

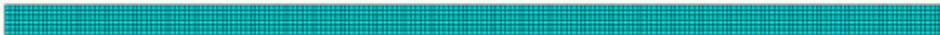
This example uses a double-cantilever beam with an edge crack at one end. Equal displacements with opposite directions are applied to the end of the beam about and below the crack in order to open up the crack, as shown in this figure:

**Figure 12.4: Crack Growth of a Double-Cantilever Beam**



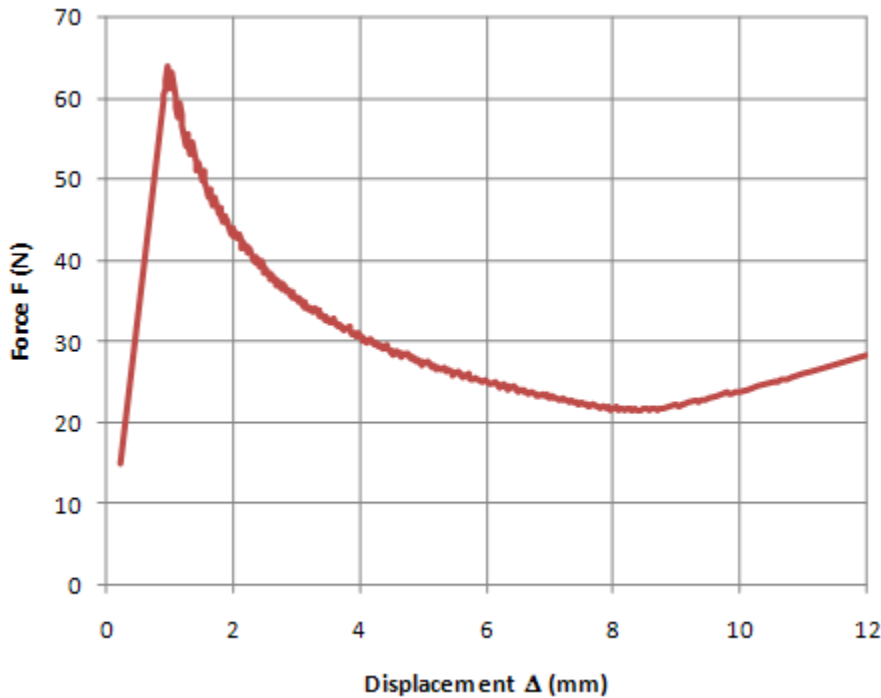
The following figure shows the finite element mesh:

**Figure 12.5: Double-Cantilever Beam Mesh**



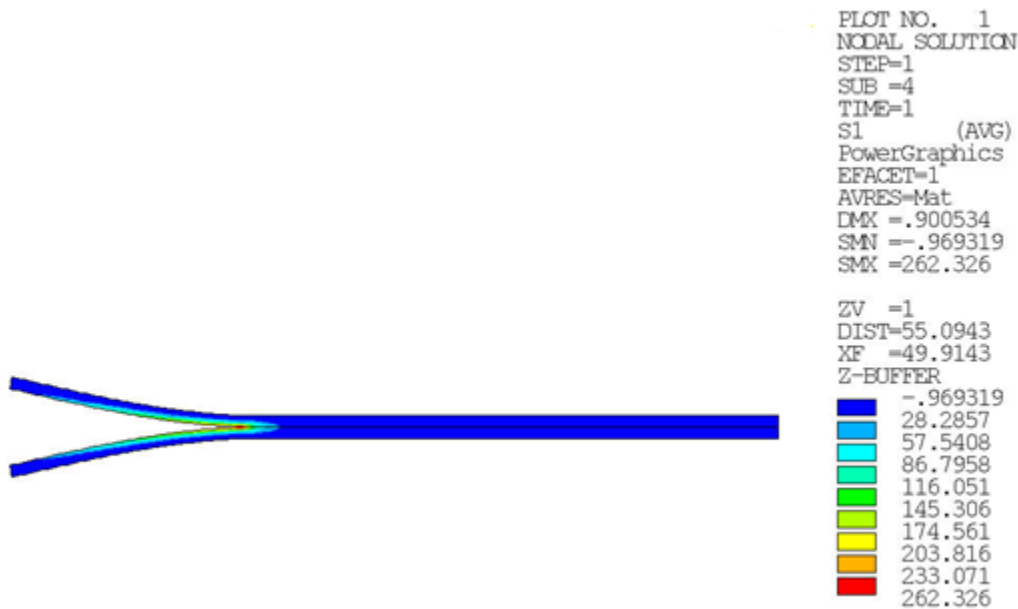
**PLANE182** with the enhanced strain option (**KEYOPT(1) = 2**) is used to model the solid part of the model. **INTER202** is used to model the crack path. A plane strain condition is assumed. In the vertical direction, the model uses 6 elements, and in the horizontal direction are 200 elements.

The following figure shows the predicted load-deflection curve:

**Figure 12.6: Double-Cantilever Beam Load-Deflection Curve**

The force increases with the applied displacement and peaks quickly before the crack begins to grow. The reaction force then decreases rapidly at the initial phase of crack growth, then slows down with the subsequent crack growth. The results match very well with the reference results [11].

The contour plot of maximum principal stress is shown in the following figure:

**Figure 12.7: Double-Cantilever Beam Contour Plot**

Following is the input file used for the crack growth simulation of the double-cantilever beam:

```

/prep7
dis1=0.9
dis2=12.0
n1=1000
n2=1000
n3=10
dl=100
dh=3
a0=30
nel=200
neh=6
toler=0.1e-5

et,1,182          !* 2d 4-node structural solid element
keyopt,1,1,2     !* enhance strain formulation
keyopt,1,3,2     !* plane strain
et,2,182
keyopt,2,1,2
keyopt,2,3,2

et,3,202         !* 2d 4-node cohesive zone element
!keyopt,3,2,2    !* element free option
keyopt,3,3,2     !* plane strain

mp,ex,1,1.353e5  !* e11 = 135.3 gpa
mp,ey,1,9.0e3    !* e22 = 9.0 gpa
mp,ez,1,9.0e3    !* e33 = 9.0 gpa
mp,gxy,1,5.2e3   !* g12 = 5.2 gpa
mp,prxy,1,0.24
mp,prxz,1,0.24
mp,pryz,1,0.46

g1c=0.28         !* critical energy-release rate
g2c=0.80
g3c=0.80
tb,cgcr,1,,3,linear !* linear fracture criterion
tbdata,1,g1c,g2c,g3c

! fe model
rectng,0,dl,dh/2 !* define areas
rectng,0,dl,0,-dh/2
lsl,s,line,,2,8,2 !* define line division
lesize,all,dh/neh
lsl,inve
lesize,all,, , nel
allsel,all
type,1           !* mesh area 2
mat,1
local,11,0,0,0,0
esys,11
amesh,2
csys,0
type,2          !* mesh area 1
esys,11
amesh,1
csys,0
nsl,s,loc,x,a0-toler,dl
nummrg,nodes
esln
type,3
mat,5
czmesh,, ,1,y,0 !* generate interface elements
allsel,all
nsl,s,loc,x,dl !* apply constraints
d,all,all
nsl,all

!
esel,s,ename,,202 !* select interface element to
cm,cpath,elem     !* define crack growth path

```

```

nsle
nlist
nsel,s,loc,x,a0
nsel,r,loc,y,0
nlist
esln
elist
cm,crack1,node          !* define crack tip node component
nlist
alls
finish

/solu
resc,,none
esel,s,type,,2
nsle,s
nsel,r,loc,x
nsel,r,loc,y,dh/2      !* apply displacement loading on top
d,all,uy,dis1
nsel,all
esel,all
esel,s,type,,1
nsle,s
nsel,r,loc,x
nsel,r,loc,y,-dh/2     !* apply displacement loading on bottom
d,all,uy,-dis1
nsel,all
esel,all
autots,on
time,1

cint,new,1             !* crack id
cint,type,vcct         !* vcct calculation
cint,ctnc,crack1      !* crack tip node component
cint,norm,0,2

! crack growth simulation set
cgrow,new,1           !* crack growth set
cgrow,cid,1           !* cint id for vcct calculation
cgrow,cpath,cpath     !* crack path
cgrow,fcop,mtab,1     !* fracture criterion
cgrow,dtime,2e-3
cgrow,dtmin,2e-3
cgrow,dtmax,2e-3
cgrow,fcra,0.98
nsub,4,4,4
allsel,all
outres,all,all
solve
time,2
esel,s,type,,2
nsle,s
nsel,r,loc,x
nsel,r,loc,y,dh/2     !* apply displacement loading on top
d,all,uy,dis2
nsel,all
esel,all
esel,s,type,,1
nsle,s
nsel,r,loc,x
nsel,r,loc,y,-dh/2   !* apply displacement loading on bottom
d,all,uy,-dis2
nsel,all
esel,all

nsubst,n1,n2,n3
outres,all,all
solve                 !* perform solution
finish

/post1
set,last

```

```

prci,1
!
set,1,4
plnsol,s,1
!
finish

/post26
nset,s,loc,y,dh/2
nset,r,loc,x,0
*get,ntop,node,0,num,max
nset,all
nsol,2,ntop,u,y,uy
rforce,3,ntop,f,y,fy
prod,4,3,,rf,,20
/title,,dcb: reaction at top node verses prescribed displacement
/axlab,x,disp u (mm)
/axlab,y,reaction force r (n)
/yrange,0,60
xvar,2
prvar,uy,rf
prvar,2,3,4
/com,
finish

```

### 12.1.5. VCCT Crack Growth Simulation Assumptions

VCCT-based crack growth simulation is available only with current-technology linear elements [PLANE182](#) and [SOLID185](#).

The following assumptions apply to VCCT-based crack growth simulation:

- The material is linearly elastic, and can be isotropic, orthotropic, or anisotropic.
- The analysis is assumed to be quasi-static. Although a transient analysis is possible, the fracture calculations do not account for the transient effects.
- The VCCT-based mixed-mode energy-release rates calculation assumes that the crack tip field / deformation at the crack tip/front location is similar to when the crack extends by a small amount. This assumption does not apply when crack growth approaches the boundary or when two cracks are close; therefore, use the VCCT calculation with care and examine the analysis results.

## 12.2. Modeling Interface Delamination with Interface Elements

A set of four interface elements is available for modeling interface delamination at the interface of two materials. The elements are capable of representing the cohesive zone between the interface and can account for the separation across the interface.

The following related topics are available:

- [12.2.1. Analyzing Interface Delamination](#)
- [12.2.2. Interface Elements](#)
- [12.2.3. Material Definition](#)
- [12.2.4. Meshing and Boundary Conditions](#)
- [12.2.5. Solution Procedure and Result Output](#)
- [12.2.6. Reviewing the Results](#)

## 12.2.1. Analyzing Interface Delamination

An interface delamination analysis with interface elements involves the same general steps that are involved in any nonlinear analysis procedure. Most of these steps, however, warrant special consideration with regard to behavior at the cohesive zone.

Following is the general procedure, with special considerations indicated, along with links to applicable sections where more detailed information is available:

1. **Build or import the model.** There are no special considerations for building or importing the model for an interface delamination analysis. You perform this step as you would in any typical analysis. See [Building the Model](#) in the *Basic Analysis Guide*. For further details on building the model, see the *Modeling and Meshing Guide*.
2. **Define element type.** To properly simulate the cohesive zone, you must define structural element types and corresponding interface element types. See [Element Selection \(p. 399\)](#) in this chapter for more details on this topic.
3. **Define material.** Use **TB,CZM** with  $TBOPT = EXPO$  or  $BILI$  to define the cohesive zone material that characterizes the separation behavior at the interface. You then input the sets of data using the **TBDATA** commands, as applicable.
4. **Mesh the model.** Use the **AMESH** or **VMESH** commands to mesh the structural elements, and use the **CZMESH** command to mesh the cohesive zone element interface along the layers. Special restrictions apply to the **CZMESH** command in terms of matching the source and target. Also, the order in which you execute these commands is critical. You can only use **CZMESH** after the underlying solid model has been meshed. You can also generate interface elements directly using the **EGEN** command. Each of these commands involves special consideration for interface elements. See [Meshing and Boundary Conditions \(p. 401\)](#) in this chapter for more details on this topic.
5. **Solve.** There are special solving consideration when you perform an interface delamination analysis. These are primarily concerned with the interface element stiffness loss or softening. Care should be taken to avoid the numerical instability that may be caused by the delamination and failure of the interface.
6. **Review Results.** You can print or plot your cohesive zone output items using the **PRESOL**, **PRNSOL**, **PLESOL**, **PLNSOL**, or **ESOL** commands. See [Reviewing the Results](#) in this chapter for more details on this topic.

## 12.2.2. Interface Elements

Four element types are available for simulating interface delamination and failure:

- **INTER202** - 2-D, 4-node, linear element.
- **INTER203** - 2-D, 6-node, quadratic element.
- **INTER204** - 3-D, 16-node, quadratic element.
- **INTER205** - 3-D, 8-node, linear element

The 2-D elements, **INTER202** and **INTER203**, use a **KEYOPT** to define various stress state options.

### 12.2.2.1. Element Definition

An interface element is composed of bottom and top surfaces. [Figure 205.1: INTER205 Geometry](#) in the [Element Reference](#) shows the geometry of a 3-D 8-node interface element. The element midplane can be created by averaging the coordinates of node pairs from the bottoms and top surfaces of the element. The numerical integration of the interface elements is performed in the element midplane. The Gauss integration scheme is used for the numerical integrations.

### 12.2.2.2. Element Selection

The simulation of an entire assembly, consisting of the cohesive zone and the structural elements on either side of the cohesive zone, requires that the interface elements and structural elements have the same characteristics. When you issue the **CZMESH** command, the appropriate interface element(s) will be selected automatically, depending on the adjacent structural elements. You can also manually specify your interface elements. Use the following table as a guideline for choosing interface and structural elements that have the same characteristics:

For elements with these characteristics:	... use this interface element:	... with one of these structural elements:
2-D, linear	<a href="#">INTER202</a>	<a href="#">PLANE182</a>
2-D, quadratic	<a href="#">INTER203</a>	<a href="#">PLANE183</a>
3-D, quadratic	<a href="#">INTER204</a>	<a href="#">SOLID186</a> , <a href="#">SOLID187</a>
3-D, linear	<a href="#">INTER205</a>	<a href="#">SOLID65</a> , <a href="#">SOLID185</a> , <a href="#">SOLSH190</a> , <a href="#">SOLID272</a> , <a href="#">SOLID273</a> , <a href="#">SOLID285</a>

Proper element type is chosen based on the stress states of interest and structural element types used.

Element selection is done by the element type command, **ET**, for example,

```
ET,1,205
```

defines element type 1 as element [INTER205](#).

## 12.2.3. Material Definition

The following material-definition topics are available for modeling interface delamination with interface elements:

[12.2.3.1. Material Characteristics](#)

[12.2.3.2. Material Constants -- Exponential Law](#)

[12.2.3.3. Material Constants -- Bilinear Law](#)

[12.2.3.4. Viscous Regularization for Cohesive Cone Material \(CZM\)](#)

### 12.2.3.1. Material Characteristics

The **TB,CZM** command defines interface separation behavior with interface elements. The interface is represented by a single element set of these elements. The interface deformation is characterized by a traction separation law (see below), with the deformation occurring only within the interface elements (the cohesive zone).

The tension or shear deformations within this zone are of primary interest. The surface behavior of the material depends on the **type of CZM model specified** (TB,CZM,,,,*TBOPT*, where *TBOPT* = EXPO or BILI). Unloading behavior is not addressed in the CZM with exponential law (TB,CZM,,,,EXPO).

The surface behavior of the material is highly nonlinear in either case, and the resulting softening or loss of stiffness changes character rapidly as the element separation increases. Unloading behavior is not addressed in this configuration.

### 12.2.3.2. Material Constants -- Exponential Law

The cohesive zone model (TB,CZM,,,,EXPO) uses a traction-separation law, defined as:

$$T_n = e\sigma_{\max}\Delta_n e^{-\Delta_n} e^{-\Delta_t^2}$$

for normal traction at the interface, and

$$T_t = 2e\sigma_{\max} \frac{\bar{\delta}_n}{\bar{\delta}_t} \Delta_t (1 + \Delta_n) e^{-\Delta_n} e^{-\Delta_t^2}$$

for shear traction at the interface, where:

$$\mathbf{T} = \frac{\partial\phi(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}}$$

The material constants  $\sigma_{\max}$ ,  $\bar{\delta}_n$ , and  $\bar{\delta}_t$  are input as C1, C2, and C3. The input format is

```
TB, CZM, , , ,
TBDATA, 1, C1, C2, C3
```

This CZM material option must be used with interface elements [INTER202](#), [INTER203](#), [INTER204](#), and [INTER205](#).

### 12.2.3.3. Material Constants -- Bilinear Law

The cohesive zone model (TB,CZM,,,,BILI) uses bilinear traction-separation laws, defined as:

$$T_n = K_n \delta_n (1 - D_m)$$

$$T_t = K_t \delta_t (1 - D_m)$$

$$\lambda = \sqrt{\left(\frac{\delta_n}{\delta_n^c}\right)^2 + \beta^2 \left(\frac{\delta_t}{\delta_t^c}\right)^2}$$

where



$$K_n = \text{normal cohesive stiffness} \frac{T_n^{\max}}{\delta_n^*}$$

$$T_n^{\max} = \text{maximum normal traction } \sigma_{\max}$$

$$\delta_n^* = \text{normal displacement jump at maximum normal traction}$$

$$\delta_n^c = \text{normal displacement jump at the completion of debonding}$$

$$K_t = \text{tangential cohesive stiffness} \frac{T_t^{\max}}{\delta_t^*}$$

$$T_t^{\max} = \text{maximum tangential traction } \tau_{\max}$$

$$\delta_t^* = \text{tangential displacement jump at maximum tangential cohesive traction}$$

$$\delta_t^c = \text{tangential displacement jump at the completion of debonding}$$

$$D_m = \text{Damage parameter } D_m(\lambda)$$

$$\alpha = \text{ratio of } \delta_n^* \text{ to } \delta_n^c, \text{ or ratio of } \delta_t^* \text{ to } \delta_t^c$$

$$\beta = \text{non-dimensional weighting parameter}$$

The material constants are input via the **TBDATA** command:

Con-stant	Mean-ing	Property
C1	$\sigma_{\max}$	Maximum normal traction
C2	$u_n^c$	Normal displacement jump at the completion of debonding
C3	$\tau_{\max}$	Maximum tangential traction
C4	$u_t^c$	Tangential displacement jump at the completion of debonding
C5	$\alpha$	Ratio of $\delta_n^*$ to $\delta_n^c$ , or ratio of $\delta_t^*$ to $\delta_t^c$
C6	$\beta$	Non-dimensional weighting parameter

For more information about defining a cohesive zone material in an interface delamination analysis, see [Bilinear Cohesive Zone Material for Interface Elements](#) in the *Material Reference*.

#### 12.2.3.4. Viscous Regularization for Cohesive Cone Material (CZM)

The [cohesive zone material \(CZM\)](#) model supports [viscous regularization \(TB,CZM,,,,VREG\)](#) for stabilizing interface delamination. For more information, see [Cohesive Material Law](#) in the *Material Reference* and [Viscous Regularization](#) in the *Mechanical APDL Theory Reference*.

### 12.2.4. Meshing and Boundary Conditions

#### 12.2.4.1. Meshing

There are three options available for meshing interface elements:

- Use the **CZMESH** command to generate the interface. You must either define the model into two components or groups of elements (between which the cohesive zone interface elements will reside), or specify a coordinate value for the line or plane that will divide the model.
- Use the **E** command to directly generate interface elements from a set of nodes.
- For generating interface elements directly from a pattern, use the **EGEN** command.

### 12.2.4.2. Boundary Conditions

The interface delamination and failure process involves the stiffness softening and complete loss of the interface stiffness, which in turn will cause numerical instability of the solution. You should therefore apply your constraints as boundary conditions. Using forces or pressures will generally cause rigid body motion after the fracture, and will result in other solution difficulties.

### 12.2.5. Solution Procedure and Result Output

Interface traction-separation behavior is highly nonlinear. The full Newton-Raphson solution procedure (the standard ANSYS nonlinear method), is the default method for performing this type of analysis. Other solution procedures for interface analyses are not recommended.

Like most nonlinear problems, convergence behavior of an interface delamination analysis depends strongly on the particular problem to be solved. ANSYS has provided a comprehensive solution control strategy, therefore it is always recommended that you use the ANSYS default solution options, unless you are sure about the benefits of any changes.

Some special considerations for solving an interface delamination problem:

- When the element breaks apart under external loading, it will lose its stiffness and may cause numerical instability.
- It is always a good practice to place the lower and upper limit on the time step size using the **DELTIM** or **NSUBST** commands, and to start with a small time step, then subsequently ramp it up. This ensures that all of the modes and behaviors of interest will be accurately included and that the problem is solved effectively.
- When interface elements are under tension, the normal stiffness is exponentially related to the separation. That is, the greater the separation, the lower the normal stiffness of the elements.
- When interface elements are under compression, you can align contact elements with the interface elements to obtain better penetration control.

A convergence failure can indicate a physical instability in the structure, or it can merely be the result of some numerical problem in the finite element model.

### 12.2.6. Reviewing the Results

Results from an interface delamination analysis consist mainly of displacements, stresses, strains and reaction forces of the structural components and the cohesive zone layer information (interface tension, separation, etc.). You can review these results in POST1, the general postprocessor, or in POST26, the time-history postprocessor. See the Output Data sections of the element descriptions for any of the interface elements (for example **INTER202**) for a description of the available output components.

Note that in POST1, only one substep can be read in at a time, and that the results from that substep should have been written to `Jobname.RST`. (The load step option command **OUTRES** controls which substep results are stored on `Jobname.RST`.) A typical POST1 postprocessing sequence is described below.

### 12.2.6.1. Points to Remember

- To review results in POST1, the database must contain the same model for which the solution was calculated.
- The results file (`Jobname.RST`) must be available.

### 12.2.6.2. Reviewing Results in POST1

1. Verify from your output file (`Jobname.OUT`) whether or not the analysis converged at all load steps.
  - If not, you probably won't want to postprocess the results, other than to determine why convergence failed.
  - If your solution converged, then continue postprocessing.

2. Enter POST1. If your model is not currently in the database, issue **RESUME**.

**Command(s): /POST1**

**GUI: Main Menu> General Postproc**

3. Read in results for the desired load step and substep, which can be identified by load step and substep numbers or by time.

**Command(s): SET**

**GUI: Main Menu> General Postproc> Read Results> load step**

4. Display the results using any of the following options. Note that cohesive zone element results, such as tension and separation, are always displayed and listed in the local coordinate system.

**Option: Display Deformed Shape**

**Command(s): PLDISP**

**GUI: Main Menu> General Postproc> Plot Results> Deformed Shape**

**Option: Contour Displays**

**Command(s): PLNSOL or PLESOL**

**GUI: Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu or Element Solu**

Use these options to display contours of stresses, strains, or any other applicable item. When displaying the interface tension distribution, if other structural mating components are not included, ANSYS will plot the geometry of those components in gray. To have a better visualization of an interface tension plot, it is better for you to select the interface elements only.

**Option: Tabular Listings**

**Command(s): PRNSOL** (nodal results) **PRESOL** (element-by-element results) **PRRSOL** (reaction data) **PRITER** (substep summary data), etc.

**GUI: Main Menu> General Postproc> List Results> Nodal Solution**

**Main Menu> General Postproc> List Results> Element Solution**

**Main Menu> General Postproc> List Results> Reaction Solution**

**Option: Animation**

You can also animate interface results over time:

**Command(s):** **ANTIME**

**GUI:** Utility Menu > PlotCtrls > Animate > Over Time

**Other Capabilities**

Many other postprocessing functions are available in POST1. See [The General Postprocessor \(POST1\)](#) in the *Basic Analysis Guide* for details. Load case combinations usually are not valid for nonlinear analyses.

**12.2.6.3. Reviewing Results in POST26**

You can also review the load-history response of a nonlinear structure using POST26, the time-history postprocessor. Use POST26 to compare one ANSYS variable against another. For instance, you might graph the interface separation vs. interface tension, which should correspond to the material behavior defined by **TB,CZM**. You might also graph the displacement at a node versus the corresponding level of applied load, or you might list the interface tension at a node and the corresponding *TIME* value. A typical POST26 postprocessing sequence for an interface delamination analysis is the same as the sequence for a typical nonlinear analysis. See steps 1 through 4 in [Reviewing Results in POST26 \(p. 253\)](#) included in [Nonlinear Structural Analysis \(p. 193\)](#).

**12.3. Modeling Interface Delamination with Contact Elements**

Interface delamination with contact elements is referred to as *debonding*. Debonding is modeled with contact elements which are bonded and have a cohesive zone material model defined. There are several advantages to using debonding to model interface delamination. Existing models with contact definitions can be easily modified to include debonding, and standard contact and debonding can be simulated with the same contact definitions.

**12.3.1. Analyzing Debonding**

Modeling debonding with contact elements involves the same steps as any other contact analysis. (For a detailed discussion on how to set up a contact analysis, see [Surface-to-Surface Contact](#).) Therefore, if you are familiar with setting up a contact analysis you can easily include debonding in your model. You simply add a bonded contact option and a cohesive zone material model for the contact elements.

**12.3.2. Contact Elements**

Debonding can be defined in any model that includes the following types of contact:

- Surface-to-surface ([CONTA171](#) through [CONTA174](#))
- Node-to-surface ([CONTA175](#))
- Line-to-line ([CONTA176](#))
- Line-to-surface ([CONTA177](#)) contact

The contact elements must use bonded contact ( $KEYOPT(12) = 2, 3, 4, 5$  or  $6$ ) with the augmented Lagrangian method or pure penalty method ( $KEYOPT(2) = 0$  or  $1$ ). Debonding is activated by associating a cohesive zone material model (input with **TB,CZM**) with the contact elements.

## 12.3.3. Material Definition

### 12.3.3.1. Material Characteristics

ANSYS provides two cohesive zone material models with bilinear behavior to represent debonding. The material behavior, defined in terms of contact stresses (normal and tangential) and contact separation distances (normal gap and tangential sliding), is characterized by linear elastic loading followed by linear softening. The slope of the curve depends on contact stiffness and a debonding parameter which is defined in terms of material constants. Debonding allows three modes of separation:

- Mode I debonding for normal separation
- Mode II debonding for tangential separation
- Mixed mode debonding for normal and tangential separation

Debonding is also characterized by convergence difficulties during material softening. Artificial damping is provided to overcome these problems. An option for tangential slip under compressive normal contact stress for mixed mode debonding is also provided.

After debonding is completed, the surface interaction is governed by standard contact constraints for normal and tangential directions. Frictional contact is used if friction is specified for the contact elements.

### 12.3.3.2. Material Constants

The cohesive zone material model with bilinear behavior (**TB,CZM** with *TBOPT* = CBDD or CBDE ) is defined as:

$$P = K_n u_n (1 - d)$$

$$\tau_y = K_t u_y (1 - d)$$

and

$$\tau_z = K_t u_z (1 - d)$$

where:

- P = normal contact stress (tension)
- $\tau_y$  = tangential contact stress in y direction
- $\tau_z$  = tangential contact stress in z direction
- $K_n$  = normal contact stiffness
- $K_t$  = tangential contact stiffness
- $u_n$  = contact gap
- $u_y$  = contact slip distance in y direction
- $u_z$  = contact slip distance in z direction
- d = debonding parameter

To model bilinear material behavior with tractions and separation distances, use **TB,CZM** with *TBOPT* = CBDD. You also input the following material constants with the **TBDATA** command:

Constant	Symbol	Meaning
C1	$\sigma_{\max}$	maximum normal contact stress [1]
C2	$\delta_n^c$	contact gap at the completion of debonding
C3	$\tau_{\max}$	maximum equivalent tangential contact stress [1]
C4	$\delta_t^c$	tangential slip at the completion of debonding
C5	$\eta$	artificial damping coefficient
C6	$\beta$	flag for tangential slip under compressive normal contact stress; must be 0 (off) or 1 (on)

1. For contact elements using the force-based model (see the description of KEYOPT(3) for [CONTA175](#), [CONTA176](#), and [CONTA177](#)), input a contact force value for this quantity.

To model bilinear material behavior with tractions and critical fracture energies, use **TB,CZM** with *TBOPT* = CBDE. You also input the following material constants with the **TBDATA** command:

Constant	Symbol	Meaning
C1	$\sigma_{\max}$	maximum normal contact stress [1]
C2	$G_{cn}$	critical fracture energy density (energy/area) for normal separation [2]
C3	$\tau_{\max}$	maximum equivalent tangential contact stress [1]
C4	$G_{ct}$	critical fracture energy density (energy/area) for tangential slip [2]
C5	$\eta$	artificial damping coefficient
C6	$\beta$	flag for tangential slip under compressive normal contact stress; must be 0 (off) or 1 (on)

1. For contact elements using the force-based model (see the description of KEYOPT(3) for [CONTA175](#), [CONTA176](#), and [CONTA177](#)), input a contact force value for this quantity.
2. For contact elements using the force-based model (see the description of KEYOPT(3) for [CONTA175](#), [CONTA176](#), and [CONTA177](#)), this quantity is critical fracture energy.

The following is an example of how to define a cohesive zone material with the **TB** and **TBDATA** commands:

```
TB,CZM,,,CBDD      ! bilinear behavior with tractions and separation distances
TBDATA,1,C1,C2,C3,C4,C5,C6
```

For more information on defining a cohesive zone material in a contact analysis, see [Cohesive Zone Material for Contact Elements](#) in the *Material Reference*.

## 12.3.4. Result Output

All applicable output quantities for contact elements are also available for debonding:

Output Quantities	Symbol	Meaning
PRES	P	normal contact stress [1]
SFRIC	$\tau_t$	tangential constant stress [1]
TAUR and TAUS	$\tau_y$ and $\tau_z$	components (tangential constant stress) [1]
GAP	$u_n$	contact gap
SLIDE	$u_t$	tangential slip
TASR and TASS	$u_y$ and $u_z$	components (tangential slip)

1. For contact elements using the force-based model (see the description of KEYOPT(3) for [CONTA175](#), [CONTA176](#), and [CONTA177](#)), this output item is reported as a force quantity.

Additionally, debonding specific output quantities are also available. These are output as NMISC data:

Output Quantities	Symbol	Meaning
DTSTART	(no symbol)	debonding time history
DPARAM	$d_n$ , $d_t$ , or $d_m$	debonding parameter
DENERI and DENERII	$G_n$ and $G_t$	fracture energies





---

## Chapter 13: Composites

---

Composite materials have been used in structures for centuries. In recent times, composite parts have been used extensively in aircraft structures, automobiles, sporting goods, and many consumer products.

Composite materials are those containing more than one bonded material, each with different structural properties. The main advantage of composite materials is the potential for a high ratio of stiffness to weight. Composites used for typical engineering applications are advanced fiber or laminated composites, such as fiberglass, glass epoxy, graphite epoxy, and boron epoxy.

ANSYS allows you to model composite materials with specialized elements called *layered elements*. After you build your model using these elements, you can perform any structural analysis (including nonlinearities such as large deflection and stress stiffening).

The following topics related to composites are available:

- [13.1. Modeling Composites](#)
- [13.2. The FiberSIM-ANSYS Interface](#)

### 13.1. Modeling Composites

Composites are somewhat more difficult to model than an isotropic material such as iron or steel. Because each layer may have different orthotropic material properties, you must exercise care when defining the properties and orientations of the various layers.

The following composite modeling topics are available:

- [13.1.1. Selecting the Proper Element Type](#)
- [13.1.2. Defining the Layered Configuration](#)
- [13.1.3. Specifying Failure Criteria for Composites](#)
- [13.1.4. Composite Modeling and Postprocessing Tips](#)

#### 13.1.1. Selecting the Proper Element Type

The following element types are available to model layered composite materials: [SHELL181](#), [SHELL281](#), [SOLSH190](#), [SOLID185](#) Layered Solid, and [SOLID186](#) Layered Solid.

The element you select depends upon your application and the type of results that must be calculated. See the individual element descriptions to determine if a specific element can be used in your ANSYS product. All layered elements allow failure criterion calculations.

##### [SHELL181](#) -- Finite Strain Shell

A 4-node 3-D shell element with 6 degrees of freedom at each node. The element has full nonlinear capabilities including large strain and allows 255 layers. The layer information is input using the section commands ([SECxxxxx](#)) rather than real constants. Failure criteria is available via [FC](#) and other [FCxxx](#) commands.

##### [SHELL281](#) -- Finite Strain Shell

An 8-node element with six degrees of freedom at each node. The element is suitable for analyzing thin to moderately-thick shell structures and is appropriate for linear, large rotation, and/or large strain nonlinear applications. The layer information is input using the section commands (SECxxxxx) rather than real constants. Failure criteria is available via **FC** and other **FCxxx** commands.

### **SOLSH190 -- 3-D Layered Structural Solid Shell**

An 8-node 3-D solid shell element with three degrees of freedom per node (UX, UY, UZ). The element can be used for simulating shell structures with a wide range of thickness (from thin to moderately thick). The element has full nonlinear capabilities including large strain. The layer information is input using the section commands rather than real constants. The element can be stacked to model through-the-thickness discontinuities. Failure criteria is available using the FC commands.

### **SOLID185 Layered Solid -- 3-D Layered Structural Solid Element**

A 3-D 8-Node Layered Solid used for 3-D modeling of solid structures. It is defined by eight nodes having three degrees of freedom at each node: translations in the nodal x, y, and z directions. The element has plasticity, hyperelasticity, stress stiffening, creep, large deflection, and large strain capabilities. It also has mixed formulation capability for simulating deformations of nearly incompressible elastoplastic materials, and fully incompressible hyperelastic materials. The element allows for prism and tetrahedral degenerations when used in irregular regions. Various element technologies such as B-bar, uniformly reduced integration, and enhanced strains are supported. Failure criteria is available via **FC** and other **FCxxx** commands. **SOLID278** is a companion thermal element.

### **SOLID186 Layered Solid -- 3-D Layered Structural Solid Element**

A higher-order version of the **SOLID185** element. **SOLID279** is a companion thermal element.

#### **13.1.1.1. Other Element Types with Composite Capabilities**

In addition to the layered elements mentioned above, other composite element capabilities are available, but are not considered further here:

- **SOLID65** -- This 3-D reinforced concrete solid element models an isotropic medium with optional reinforcing in three different user-defined orientations.
- **BEAM188** and **BEAM189** -- These 3-D finite strain beam elements can have their sections built up with multiple materials.

#### **13.1.2. Defining the Layered Configuration**

The most important characteristic of a composite material is its layered configuration. Each layer may be made of a different orthotropic material and may have its principal directions oriented differently. For laminated composites, the fiber directions determine layer orientation.

Specify individual layer properties to define the layered configuration.

The following topics related to defining the layered configuration are available:

13.1.2.1. Specifying Individual Layer Properties

13.1.2.2. Sandwich and Multiple-Layered Structures

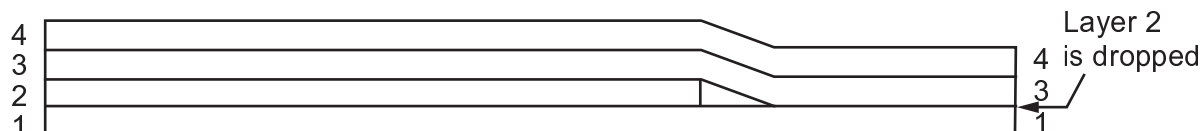
13.1.2.3. Node Offset

### 13.1.2.1. Specifying Individual Layer Properties

With this method, the layer configuration is defined layer-by-layer from bottom to top. The bottom layer is designated as layer 1, and additional layers are stacked from bottom to top in the positive Z (normal) direction of the element coordinate system. You need to define only half of the layers if stacking symmetry exists.

At times, a physical layer will extend over only part of the model. In order to model continuous layers, these dropped layers may be modeled with zero thickness. [Figure 13.1: Layered Model Showing Dropped Layer](#) (p. 411) shows a model with four layers, the second of which is dropped over part of the model.

**Figure 13.1: Layered Model Showing Dropped Layer**



For each layer, the following properties are specified in the element real constant table (**R**, **RMORE**, **RMODIF**) accessed with REAL attributes:

- Material properties (via a material reference number MAT)
- Layer orientation angle commands (THETA)
- Layer thickness (TK)

You can also define layered sections via the Section Tool (**Prep>Sections>Shell - Add/edit**). For each layer, the following are specified in the section definition via the section commands (**SECTYPE**, **SECDATA**) or the Section Tool accessed with the **SECNUM** attributes.

- Material properties (via a material reference number MAT)
- Layer orientation angle commands (THETA)
- Layer thickness (TK)
- Number of integration points per layer (NUMPT)

#### Layer Property Descriptions

Following is more information about each of the layer properties:

- *Material Properties* -- As with any other element, the **MP** command defines the linear material properties, and the **TB** command is used to define the nonlinear material data tables. The only difference is that the material attribute number for each layer of an element is specified in the element's real constant table. For the layered elements, the **MAT** command attribute is used only for the BETD, ALPD, DMPPR, and REFT arguments of the **MP** command. The [linear material properties](#) for each layer may be either isotropic or orthotropic. Typical fiber-reinforced composites contain orthotropic materials and these properties are most often supplied in the major Poisson's ratio form. Material property directions are parallel to the layer coordinate system, defined by the element coordinate system and the layer orientation angle.
- *Layer Orientation Angle* -- Defines the orientation of the layer coordinate system with respect to the element coordinate system. It is the angle (in degrees) between X-axes of the two systems. By default, the layer coordinate system is parallel to the element coordinate system. All elements have a default coordinate

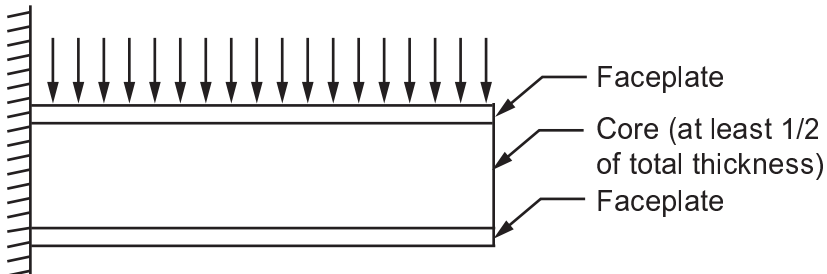
system which you can change using the ESYS element attribute (**ESYS**). You may also write your own subroutines to define the element and layer coordinate systems (USERAN and USANLY); for more information, see the *Guide to User-Programmable Features*.

- *Layer Thickness* -- If the layer thickness is constant, you only need to specify TK(I), the thickness at node I. Otherwise, the thicknesses at the four corner nodes must be input. Dropped layers may be represented with zero thickness.
- *Number of integration points per layer* -- Allows you to determine in how much detail the program should compute the results. For very thin layers, when used with many other layers, one point would be appropriate. But for laminates with few layers, more would be needed. The default is three points. This feature applies only to sections defined via the SECxxxxxx section commands.

### 13.1.2.2. Sandwich and Multiple-Layered Structures

Sandwich structures have two thin faceplates and a thick, but relatively weak, core. [Figure 13.2: Sandwich Construction \(p. 412\)](#) illustrates sandwich construction.

**Figure 13.2: Sandwich Construction**



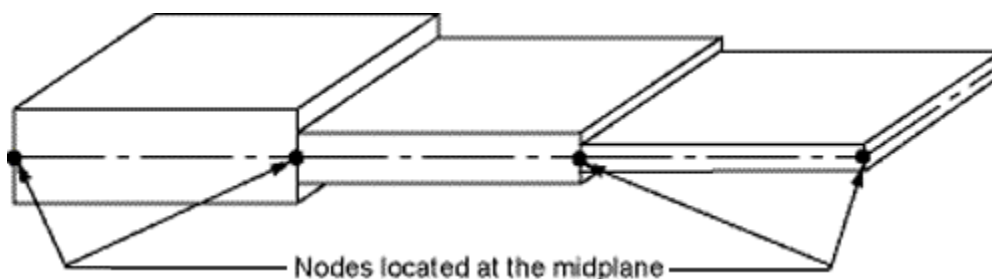
You can model sandwich structures with [SHELL181](#) or [SHELL281](#); however, both elements model the transverse-shear deflection using an energy-equivalence method that makes the need for a sandwich option unnecessary.

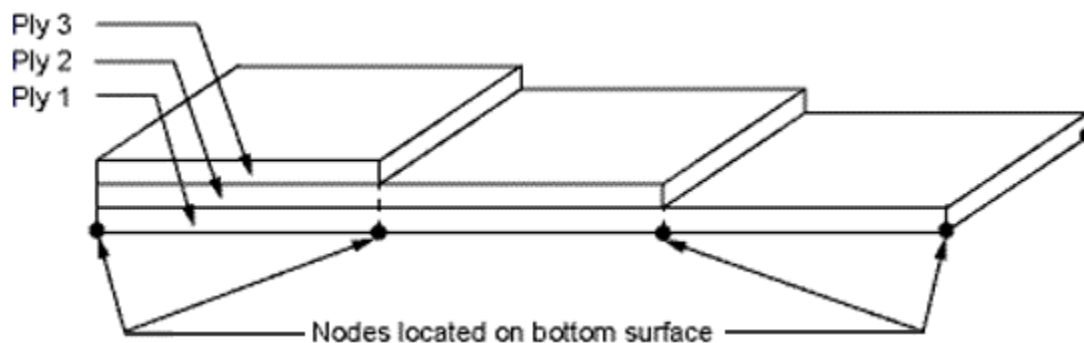
### 13.1.2.3. Node Offset

For [SHELL181](#) and [SHELL281](#) using sections defined via the section commands (SECxxxxxx), nodes can be offset during the definition of the section via the **SECOFFSET** command.

The figures below illustrate how you can conveniently model ply drop-off in shell elements that are adjacent to each other. In [Figure 13.3: Layered Shell With Nodes at Midplane \(p. 412\)](#), the nodes are located at the middle surfaces and these surfaces are aligned. In [Figure 13.4: Layered Shell With Nodes at Bottom Surface \(p. 413\)](#), the nodes are located at the bottom surfaces and these surfaces are aligned.

**Figure 13.3: Layered Shell With Nodes at Midplane**



**Figure 13.4: Layered Shell With Nodes at Bottom Surface**

### 13.1.3. Specifying Failure Criteria for Composites

Failure criteria are used to learn if a layer has failed due to the applied loads. You can choose from three predefined failure criteria or specify up to six failure criteria of your own (user-written criteria). The predefined criteria are:

- **Maximum Strain Failure Criterion**-- Allows nine failure strains.
- **Maximum Stress Failure Criterion** -- Allows nine failure stresses.
- **Tsai-Wu Failure Criterion** -- Allows nine failure stresses and three additional coupling coefficients.

The failure strains, stresses, and coupling coefficients may be temperature-dependent. For details about the data required for each criterion, see the [Element Reference](#).

Failure criteria are orthotropic, so you *must* input the failure stress or failure strain values for *all* directions. (The exception is that compressive values default to tensile values.)

If you do not want the failure stress or strain to be checked in a particular direction, specify a large number in that direction.

The following topics for specifying failure criteria for composites are available:

[13.1.3.1. Using the FC Family of Commands](#)

[13.1.3.2. User-Written Failure Criteria](#)

For more information, see [Failure Criteria](#) in the [Mechanical APDL Theory Reference](#).

#### 13.1.3.1. Using the FC Family of Commands

To specify a failure criterion, use the family of **FC** commands. These **FC** commands can be used for any 2-D or 3-D structural solid element or any 3-D structural shell element: **FC**, **FCDELE**, and **FCLIST**.

A typical sequence of commands to specify a failure criterion using these commands follows:

```
FC,1,TEMP,, 100, 200 ! Temperatures
FC,1,S,XTEN, 1500, 1200 ! Maximum stress components
FC,1,S,YTEN, 400, 500
FC,1,S,ZTEN,10000, 8000
FC,1,S,XY , 200, 200
FC,1,S,YZ ,10000, 8000
FC,1,S,XZ ,10000, 8000
FCLIST, ,100 ! List status of Failure Criteria at 100.0 degrees
FCLIST, ,150 ! List status of Failure Criteria at 150.0 degrees
FCLIST, ,200 ! List status of Failure Criteria at 200.0 degrees
PRNSOL,S,FAIL ! Use Failure Criteria
```

### 13.1.3.2. User-Written Failure Criteria

You can specify user-written failure criteria via user subroutine `USERFC`. This subroutine should be linked with the program beforehand. For more information, see [User-Programmable Features and Nonstandard Uses](#) in the *Advanced Analysis Guide*.

### 13.1.4. Composite Modeling and Postprocessing Tips

Following are a few helpful hints and tips for modeling and postprocessing composite elements:

- 13.1.4.1. [Dealing with Coupling Effects](#)
- 13.1.4.2. [Obtaining Accurate Interlaminar Shear Stresses](#)
- 13.1.4.3. [Verifying Your Input Data](#)
- 13.1.4.4. [Specifying Results File Data](#)
- 13.1.4.5. [Selecting Elements with a Specific Layer Number](#)
- 13.1.4.6. [Specifying a Layer for Results Processing](#)
- 13.1.4.7. [Transforming Results to Another Coordinate System](#)

#### 13.1.4.1. Dealing with Coupling Effects

Composites exhibit several types of coupling effects, such as coupling between bending and twisting, coupling between extension and bending, etc. This is due to stacking of layers of differing material properties. As a result, if the layer stacking sequence is not symmetric, you may not be able to use model symmetry even if the geometry and loading are symmetric, because the displacements and stresses may not be symmetric.

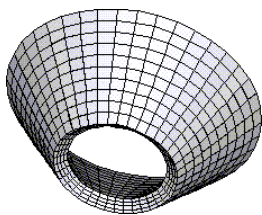
#### 13.1.4.2. Obtaining Accurate Interlaminar Shear Stresses

Interlaminar shear stresses are usually important at the free edges of a model. For relatively accurate interlaminar shear stresses at these locations, the element size at the boundaries of the model should be approximately equal to the total laminate thickness. For shells, increasing the number of layers per actual material layer does not necessarily improve the accuracy of interlaminar shear stresses. Interlaminar transverse-shear stresses in shell elements are based on the assumption that no shear is carried at the top and bottom surfaces of the element. These interlaminar shear stresses are only computed in the interior and are not valid along the shell element boundaries. Use of shell-to-solid submodeling is recommended to accurately compute all of the free edge interlaminar stresses.

#### 13.1.4.3. Verifying Your Input Data

Because a large amount of input data is necessary for composites, it is a good idea to verify the data before proceeding with the solution. The following commands are available for this purpose:

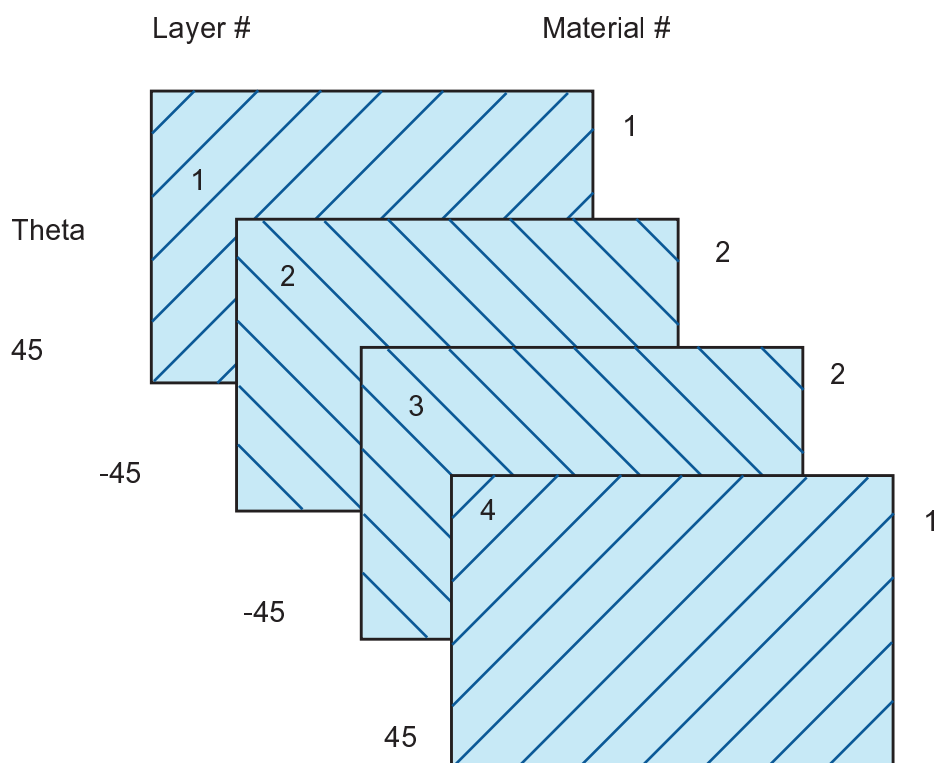
- **ELIST** -- Lists the nodes and attributes of all selected elements.
- **EPLOT** -- Displays all selected elements. Using the `/ESHAPE,1` command before **EPLOT** causes shell elements to be displayed as solids with the layer thicknesses obtained from real constants or section definition (see [Figure 13.5: Example of an Element Display](#) (p. 415)).
- `/PSYMB,LAYR,n` followed by **EPLOT** -- Displays layer number *n* for all selected layered elements. Use this method to display and verify each individual layer across the entire model. To use `/PSYMB,LAYR` with smeared reinforcing elements (`REINF265`), first set the vector-mode graphics option (`/DEVICE,VECTOR,1`).
- `/PSYMB,ESYS,1` followed by **EPLOT** -- Displays the element coordinate system triad for those elements whose default coordinate system has been changed.

**Figure 13.5: Example of an Element Display**

- **LAYPLOT** -- Displays the layer stacking sequence from real constants in the form of a sheared deck of cards. (See Figure 13.6: Sample **LAYPLOT** Display for [45/-45/ - 45/45] Sequence (p. 415).) The layers are crosshatched and color coded for clarity. The hatch lines indicate the layer angle (real constant THETA) and the color indicates layer material number (MAT). You can specify a range of layer numbers for the display.
- **SECPLOT** -- Displays the section stacking sequence from sections in the form of a sheared deck of cards. (See Figure 13.6: Sample **LAYPLOT** Display for [45/-45/ - 45/45] Sequence (p. 415).) The sections are crosshatched and color coded for clarity. The hatch lines indicate the layer angle (THETA) and the color indicates layer material number (MAT) defined by the **SECDATA** command. You can specify a range of layer numbers for the display.

#### 13.1.4.4. Specifying Results File Data

By default, only data for the bottom of the first (bottom) layer and top of the last (top) layer are written to the results file. If you are interested in data for all layers, set KEYOPT(8) = 1. Be aware, though, that this may result in a large results file.

**Figure 13.6: Sample LAYPLOT Display for [45/-45/ - 45/45] Sequence**



### 13.1.4.5. Selecting Elements with a Specific Layer Number

Use the **ESEL,S,LAYER** command to select elements that have a certain layer number. If an element has a zero thickness for the requested layer, the element is not selected.

For energy output, the results are applicable only to the entire element. You cannot get output results for individual layers.

### 13.1.4.6. Specifying a Layer for Results Processing

Use the **LAYER** command (in the POST1 postprocessor) or the **LAYERP26** (in the POST26 postprocessor) to specify the layer number for which results should be processed.

The **SHELL** command specifies a TOP, MID, or BOT location *within* the layer.

The default in POST1 is to store results for the bottom of the bottom layer, and the top of the top layer, and the layer with the maximum failure criterion value. In POST26, the default is layer 1.

If KEYOPT(8) = 1 (data stored for all layers), the **LAYER** and **LAYERP26** commands store the TOP and BOT results for the specified layer number. MID values are then calculated by average TOP and BOT values.

If KEYOPT(8) = 2 is set for **SHELL181** or **SHELL281** during SOLUTION, then **LAYER** and **LAYERP26** commands store the TOP, BOTTOM, and MID results for the specified layer number. In this case, MID values are directly retrieved from the results file. For transverse-shear stresses with KEYOPT(8) = 0, therefore, POST1 can only show a linear variation, whereas the element solution printout or KEYOPT(8) = 2 can show a parabolic variation.

### 13.1.4.7. Transforming Results to Another Coordinate System

By default, the POST1 postprocessor displays all results in the global Cartesian coordinate system. Issue the **RSYS** command to transform the results to a different coordinate system. In particular, **RSYS,SOLU** allows you to display results in the layer coordinate system if **LAYER** is issued with a nonzero layer number.

## 13.2. The FiberSIM-ANSYS Interface

FiberSIM (a product of [Vistagy, Inc.](#)) is a fiber draping tool used within popular CAD systems. FiberSIM provides accurate fiber paths for structural analysis, allows you to optimize potential laminate architectures, and generates flat patterns for product definition.

The FiberSIM-ANSYS interface allows you to use the information contained in a FiberSIM `.xml` file in your ANSYS model. (ANSYS does not use FiberSIM `.fml` files.) Generated by FiberSIM's draping calculations, the `.xml` file data contains the order of layers (including dropped layer information) and the layer orientation. In ANSYS, you supplement that information by adding material and thickness information to each layer via ANSYS section (**SECxxx**) commands.

### Elements Supported

The FiberSIM-ANSYS interface applies to the following elements only: **SHELL181**, **SOLID185** Layered Solid, **SOLID186** Layered Solid, **SOLSH190**, and **SHELL281**. When using solid elements, only one solid element thru the thickness should be used. The interface does not support the VT Accelerator or ANSYS DesignXplorer products.



## Available Topics

The following FiberSIM-ANSYS interface topics are available:

[13.2.1. Understanding the FiberSIM XML File](#)

[13.2.2. Using FiberSIM Data in ANSYS](#)

[13.2.3. FiberSIM-to-ANSYS Translation Details](#)

### 13.2.1. Understanding the FiberSIM XML File

The FiberSIM .xml file contains detailed orientation data for each layer in the laminate. Each FiberSIM ply is represented as a triangular facet mesh with node data (X, Y, Z) followed by facet connectivity data (node1, node2, node3, warp, weft). The data is necessary in ANSYS, where the fiber orientation is required over the entire layer surface. You map the FiberSIM .xml data to existing ANSYS elements using the FiberSIM orientation data. The warp and weft are the angles in radians from side node 1 - node 2 of the facet of the two fiber orientations. Hence, warp is used to define the layer orientation. Weft is not used.

For sections incorporating FiberSIM data, the .xml file containing the FiberSIM data must exist in its original format. ANSYS cannot use the data if you have altered the file in any way (for example, by opening the file in an XML editor, making changes, and then saving it again).

ANSYS accepts long ply names (up to 72 characters) if they are present in the FiberSIM file.

In this example, the sample.xml FiberSIM file demonstrates an extremely simple model of a square with five layers, including one partially dropped layer:

```
<?xml version="1.0"?>
<Laminate_Layup PART_NAME="LAMINATE1">
<Orientation XYZ="0.0 0.0 0.0"/>
<Ply NAME="STEEL">
<Material NAME="PPG-PL-3K-D1" THICKNESS="0.266700" CURED_THICKNESS="0.190500"/>
<Mesh_Point_Table COUNT="4" RECORD="XYZ">
0.000000 0.000000 0.000000<R/>
30.000000 0.000000 0.000000<R/>
0.000000 30.000000 0.000000<R/>
30.000000 30.000000 0.000000<R/>
</Mesh_Point_Table>
<Mesh_Polygon_Table COUNT="2" RECORD="NODES,WARP,WEFT">
1 2 3,0.200000,1.745329<R/>
4 2 3,0.200000,1.745329<R/>
</Mesh_Polygon_Table>
</Ply>
<Ply NAME="ALUM">
<Material NAME="PPG-PL-3K-D1" THICKNESS="0.266700" CURED_THICKNESS="0.170500"/>
<Mesh_Point_Table COUNT="4" RECORD="XYZ">
0.000000 0.000000 0.000000<R/>
30.000000 0.000000 0.000000<R/>
0.000000 30.000000 0.000000<R/>
30.000000 30.000000 0.000000<R/>
</Mesh_Point_Table>
<Mesh_Polygon_Table COUNT="2" RECORD="NODES,WARP,WEFT">
1 2 3,0.400000,1.919862<R/>
4 2 3,0.400000,1.919862<R/>
</Mesh_Polygon_Table>
</Ply>
<Ply NAME="ALUM5">
<Material NAME="PPG-PL-3K-D1" THICKNESS="0.266700" CURED_THICKNESS="0.150500"/>
<Mesh_Point_Table COUNT="4" RECORD="XYZ">
0.000000 0.000000 0.000000<R/>
30.000000 0.000000 0.000000<R/>
0.000000 30.000000 0.000000<R/>
30.000000 30.000000 0.000000<R/>
</Mesh_Point_Table>
<Mesh_Polygon_Table COUNT="2" RECORD="NODES,WARP,WEFT">
```

```

1 2 3,0.600000,2.094395<R/>
4 2 3,0.600000,2.094395<R/>
</Mesh_Polygon_Table>
</Ply>
<Ply NAME="ALUM">
<Material NAME="PPG-PL-3K-D1" THICKNESS="0.266700" CURED_THICKNESS="0.130500" />
<Mesh_Point_Table COUNT="4" RECORD="XYZ">
0.000000 0.000000 0.000000<R/>
30.000000 0.000000 0.000000<R/>
0.000000 30.000000 0.000000<R/>
30.000000 30.000000 0.000000<R/>
</Mesh_Point_Table>
<Mesh_Polygon_Table COUNT="1" RECORD="NODES,WARP,WEFT">
3 4 1,0.800000,2.268928<R/>
</Mesh_Polygon_Table>
</Ply>
<Ply NAME="STEEL">
<Material NAME="PPG-PL-3K-D1" THICKNESS="0.266700" CURED_THICKNESS="0.110500" />
<Mesh_Point_Table COUNT="4" RECORD="XYZ">
0.000000 0.000000 0.000000<R/>
30.000000 0.000000 0.000000<R/>
0.000000 30.000000 0.000000<R/>
30.000000 30.000000 0.000000<R/>
</Mesh_Point_Table>
<Mesh_Polygon_Table COUNT="2" RECORD="NODES,WARP,WEFT">
1 2 3,1.000000,2.443461<R/>
4,2,3,1.000000,2.443461<R/>
</Mesh_Polygon_Table>
</Ply>
</Laminate_Layup>

```

You may want to examine a FiberSIM .xml file before use to verify that the geometry data in the .xml file and in the ANSYS model use the same system of units and the same coordinate system. An effective method for doing so is to convert each layer to an ANSYS PREP7 input file and then plot it, as follows:

1. Select the layer of interest and delete the remainder of the file.
2. Remove the warp and weft angles as well as the <R/> symbols.
3. Using a spreadsheet program or text editor, convert the nodal and element blocks into ANSYS **N** and **E** commands. (Include columns for the command names and commas.)
4. Add **/PREP7** and **ET,1,200,4** commands at the beginning of the file.
5. Run the input file and plot the results.

### FiberSIM's XML File Format

Beginning with FiberSIM 5.1, the FiberSIM XML file contains a material-attribute line of code for each ply, similar to the following:

```
<Material NAME="PPG-PL-3K-D1" THICKNESS="0.266700" CURED_THICKNESS="0.190500" />
```

ANSYS uses only the CURED\_THICKNESS attribute and ignores the others. ANSYS considers the cured thickness to be the layer thickness when *TKFLG* = 0 (**SECCONTROL**).

### Limitations of the FiberSIM XML File

The FiberSIM-ANSYS interface assumes that all layers are built up starting at one side and going through all layers to the other side. FiberSIM has an option where layers are built up from both sides, ending in the middle; however, the interface does not support this option.

The FiberSIM-ANSYS interface supports only "plies," and not "cores."

## 13.2.2. Using FiberSIM Data in ANSYS

The general process for using FiberSIM .xml file data in your ANSYS simulation follows:

1. Create nodes.
2. Define material properties.
3. Specify element types.
4. Issue section commands (**SECTYPE**, **SECDATA**, **SECCONTROL**, **SECOFFSET**, **SECREAD**, and **SLIST**) as needed to apply material properties and thickness to flat layers.
5. Define elements.
6. Specify nodal DOF constraints as needed.
7. Solve.

The FiberSIM-ANSYS translation combines the layup order and layer orientations from FiberSIM with the material and thickness information that you provide via ANSYS section (SECxxx) commands.

The following simple input file illustrates each part of the process:

```

/bat,list
/show,sample,grph
/prep7
/title, Sample FiberSIM input data handling using SHELL281 !
! Create nodes
!
n,21
n,22,10
n,23,10,7
n,24, ,7
ngen,2,10,11,14,1,,4
!
! Define material properties
!
mp, ex,1,30e6 ! steel
mp, ex,2,10e6 ! alum
mp, ex,3,11e6 ! alum5
mp,prxy,1,.3
mp,prxy,2,.3
mp,prxy,3,.3
mp,dens,1,.00003
mp,dens,2,.00001
mp,dens,3,.00001
!
! Setup for element 2 using section commands and
! FiberSIM to define layer info
et,2,281
keyopt,2,8,1
sectype,4,shell,fibersim, lamin1
secdata,0.2,1,,STEEL
secdata,1.0,2,,ALUM
secdata,0.2,3,,ALUM5
seccontrol, , , , .05 , , , ,100,0.155, 0 , 15 , 0 , 0
!
secoffset,bot
secread,sample.xml
/com ===== issue slist,all
slist,all
!
! Define elements
!
type,2 $real,6 $secn,4 $e,21,22,23,24,25,26,27,28 !

```

```

!
! Check information
!
/view,1,4,1,1
/eshape,1
eplot
/com laylist,1 =====
laylist,1
/com layplot,1 =====
layplot,1
finish
!
! Model is finished
! Apply the loads
!
/solution
outpr,,1
d,all,all
!
! Apply in-plane load
!
d,21,ux,.003
solve
d,all,all
!
! Apply lateral acceleration, acting on the density and added mass !
acel,,100
solve
finish
!
! Examine the results
!
/graphics,power
/post1
nlist,all
set,1,1
layer,5
presol,s,comp
prnsol,s,comp
layer,0
/eshape,1
plnsol,s,x
finish

```

### 13.2.3. FiberSIM-to-ANSYS Translation Details

Following is a general description of the process that the ANSYS program uses to convert FiberSIM data for use in an ANSYS simulation:

1. ANSYS computes the element centroid.
2. For each layer, ANSYS searches the FiberSIM `.xml` file to find the first triangular facet that includes the centroid to the tolerances specified via the **SECCONTROL** command.
3. If no facets are found, ANSYS assumes a dropped layer for that element.
4. If ANSYS finds more than one qualifying facet, it uses only the first one to define the angle of the layer.
5. If ANSYS finds no facets in any layer, it assumes that an error has occurred and terminates. If you input a non-zero edge tolerance value (**SECCONTROL**,,,,,,,*EDGTOL*), the error message includes the distance to the nearest FiberSIM triangle.

You can monitor the search process by setting the **SECCONTROL** command's *NEL* option. The option generates debug output for the specified number of elements.

When a layer has darts and the cut ends of the material are brought close together, the layer is not continuous across those parts. Therefore, ensure that a sufficient number of ANSYS elements are defined in those areas so that at least one element across the transition does not include that layer.

For large models, the search process can be time-consuming; therefore, for iterative analyses during the solution phase, ANSYS saves the layer information in the element saved-data file (.ESAV). ANSYS does not save the layer information during preprocessing and postprocessing, however, so operations such as **LAYPLOT** and **PRESOL,S,X** may require more time to execute than expected.

The **CDWRITE** command cannot archive FiberSIM data. Instead, retain the FiberSIM .xml file and reenter the section (SECxxxxxx) commands necessary to generate the complete model.



---

## Chapter 14: Fatigue

---

Fatigue is the phenomenon in which a repetitively loaded structure fractures at a load level less than its ultimate static strength. For instance, a steel bar might successfully resist a single static application of a 300 kN tensile load, but might fail after 1,000,000 repetitions of a 200 kN load.

The primary factors that contribute to fatigue failures include:

- Number of load cycles experienced
- Range of stress experienced in each load cycle
- Mean stress experienced in each load cycle
- Presence of local stress concentrations

A formal fatigue evaluation accounts for each of these factors as it calculates how "used up" a certain component will become during its anticipated life cycle.

For beam joint fatigue, also refer to the [FATJACK](#) solver.

The following fatigue-related topics are available:

- [14.1. How Fatigue Is Calculated](#)
- [14.2. Fatigue Terminology](#)
- [14.3. Evaluating Fatigue](#)

### 14.1. How Fatigue Is Calculated

The fatigue calculations rely on the ASME Boiler and Pressure Vessel Code, Section III (and Section VIII, Division 2) for guidelines on range counting, simplified elastic-plastic adaptations, and cumulative fatigue summation by Miner's rule.

For fatigue evaluations based on criteria other than those of the ASME Code, you can either write your own macro, or else interface your results with an appropriate third-party program (see the [ANSYS Parametric Design Language Guide](#) for more information on these two features).

The program features the following fatigue-calculation capabilities:

- You can postprocess existing stress results to determine the fatigue usage factors for any solid-element or shell-element model. (You can also manually input stresses for fatigue evaluation of line-element models.)
- You can store stresses at a preselected number of locations for a preselected number of events and loadings within the event.
- You can define stress concentration factors for each location and scale factors for each event.

## 14.2. Fatigue Terminology

A *location* is a node in your model for which fatigue stresses are to be stored. You would typically choose locations that represent points on the structure that would be susceptible to fatigue damage.

An *event* is a set of stress conditions that occur at different times during a unique stress cycle. See [Guidelines for Obtaining Accurate Usage Factors \(p. 430\)](#) later in this chapter for more information.

A *loading* is one of the stress conditions that is part of an event.

The *alternating stress intensity* is a measure of the difference in stress state between any two loadings. The program does *not* adjust the alternating stress intensity for mean-stress effects.

## 14.3. Evaluating Fatigue

You perform a fatigue evaluation in POST1, the general postprocessor, after you have completed a stress solution. The procedure normally consists of five general steps:

1. Enter POST1 (**/POST1**) and resume your database.
2. Establish the size (the number of locations, events, and loadings), define the fatigue material properties, identify stress locations, and define stress concentration factors.
3. Store stresses at locations of interest for various events and loadings; assign event repetitions and scale factors.
4. Activate the fatigue calculations.
5. Review the results.

The following fatigue-evaluation topics describe each step in detail and provide additional information:

[14.3.1. Enter POST1 and Resume Your Database](#)

[14.3.2. Establish the Size, Fatigue Material Properties, and Locations](#)

[14.3.3. Store Stresses and Assign Event Repetitions and Scale Factors](#)

[14.3.4. Activate the Fatigue Calculations](#)

[14.3.5. Review the Results](#)

[14.3.6. Other Approaches to Range Counting](#)

[14.3.7. Sample Input](#)

### 14.3.1. Enter POST1 and Resume Your Database

In order to perform a fatigue evaluation, you need to follow these steps:

1. Enter POST1.  
**Command(s): /POST1**  
**GUI: Main Menu > General Postproc**
2. Read your database file (JOBNAME.DB) into active memory. (If your intended fatigue evaluation is a continuation of an ongoing session, then JOBNAME.DB will already be in memory.) A results file (JOBNAME.RST) with nodal stress results should also be available; you will read results data into memory later.  
**Command(s): RESUME**  
**GUI: Utility Menu > File > Resume from**



## 14.3.2. Establish the Size, Fatigue Material Properties, and Locations

Define the following data:

- Maximum number of locations, events, and loadings
  - Fatigue material properties
  - Stress locations and stress concentration factors (SCFs)
1. Define the maximum number of stress locations, events, and loadings.

By default, your fatigue evaluation can consider up to five nodal locations, ten events, and three loadings within an event. You can use the following option to establish larger dimensions (that is, allow more locations, events, or loadings), if necessary.

**Command(s):** **FTSIZE**

**GUI: Main Menu > General Postproc > Fatigue > Size Settings**

2. Define material fatigue properties.

In order to calculate usage factors, and to include the effect of simplified elastic-plastic computations, you must define material fatigue properties. The material properties of interest in a fatigue evaluation are:

- The S-N curve, a curve of alternating stress intensity ( $(S_{\max} - S_{\min})/2$ ) versus allowable number of cycles. The ASME S-N curves already account for maximum mean stress effects. You should adjust your S-N curve to account for mean-stress effects, if necessary. If you do not input an S-N curve, alternating stress intensities will be listed in decreasing order for all possible combinations of stress conditions, but no usage factors will be calculated.

**Command(s):** **FP**

**GUI: Main Menu > General Postproc > Fatigue > Property Table > S-N Table**

- The  $S_m$ -T curve, a curve of design stress-intensity value versus temperature. This curve is needed if you want the program to detect whether or not the nominal stress range has gone plastic.

**Command(s):** **FP**

**GUI: Main Menu > General Postproc > Fatigue > Property Table > Sm\_T Table**

- Elastic-plastic material parameters M and N (strain hardening exponents). These parameters are required only if you desire simplified elastic-plastic code calculations. These parameters' values can be obtained from the ASME Code.

**Command(s):** **FP**

**GUI: Main Menu > General Postproc > Fatigue > Property Table > Elas-plas Par**

The following example illustrates the use of the **FP** command to input material fatigue properties:

```
! Define the S-N table:
FP,1,10,30,100,300,1000,10000      ! Allowable Cycles, N
FP,7,100000,1000000                ! "
FP,21,650,390,240,161,109,59      ! Alternating Stress-
FP,27,37,26                        ! Intensity Range, S, ksi
! Define the Sm-T table:
FP,41,100,200,300,400,500,600      ! Temperature, °F
FP,47,650,700,750,800             ! "
FP,51,20,20,20,18.7,17.4,16.4     ! "Design Stress-Intensity
FP,57,16.1,15.9,15.5,15.1         ! Value", Sm (=2/3*Sy or
! 1/3 *Su), ksi
```

```
! Define the elastic-plastic material parameters:
FP,61,1.7,.3           ! M and N
```

### 3. Define stress locations and stress concentration factors.

The following option allows you to explicitly define a nodal location of interest to your fatigue evaluation, define stress concentration factors (SCFs) for that location, and assign a short (20 character) title to that location.

**Command(s): FL**

**GUI: Main Menu > General Postproc > Fatigue > Stress Locations**

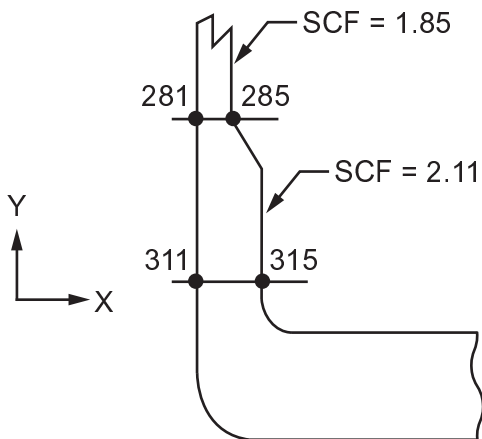
#### Note

Not all fatigue analyses will require the **FL** command. Locations are automatically defined for nodes when **FS**, **FSNODE**, or **FSSECT** are issued (see below). If your model contains sufficient grid detail, your stresses could be accurate enough that you would not need to apply calculated SCFs. (Supplemental SCFs for surface, size, or corrosion effects might still be required, however.) Where only one location is being examined, you could omit a title. If explicit definition of locations, SCFs, or titles are not required, you could forgo the **FL** command entirely.

Here is an example of some **FL** commands for a cylinder with a global Y axis, having two wall thicknesses of interest, where SCFs are to be applied (to the axial linearized stresses) at the outside wall.

```
FL,1,281,,,,Line 1 at inside
FL,2,285,,1.85,,Line 1 at outside
FL,3,311,,,,Line 2 at inside
FL,4,315,,2.11,,Line 2 at outside
```

**Figure 14.1: Cylinder Wall with Stress Concentration Factors (SCFs)**



## 14.3.3. Store Stresses and Assign Event Repetitions and Scale Factors

### 14.3.3.1. Storing Stresses

In order to perform a fatigue evaluation, the program must know the stresses at different events and loadings for each location, as well as the number of repetitions of each event. You can store stresses for each combination of location, event, and loading, using the following options:

- Manually stored stresses

- Nodal stresses from Jobname .RST
- Stresses at a cross-section

---

### Caution

*The program never assumes that a "zero" stress condition exists. If zero stress is one of the conditions to be considered, it must be explicitly input for each event in which it may occur.*

---

The following command sequences schematically illustrate how to store stresses. In some situations, you might prefer to use **LCASE** instead of **SET**.

Manually stored stresses:	FS
Nodal stresses retrieved from Job- name .RST:	SET, FSNODE
Stresses at a cross-section:	PATH, PPATH, SET, FSSECT

(Cross-section calculations also require data from Jobname .RST.)

You can use more than one method of storing stresses in an event. Each of these methods is explained in detail below.

#### 14.3.3.1.1. Manually Stored Stresses

You can use this option to store stresses and the temperature, and the time "manually" (without direct access to the results file Jobname .RST). In such cases, you are not using the fatigue module in POST1 as a postprocessor, but simply as a fatigue calculator. Line elements, such as beams, *must* be handled in this way since the fatigue module is not able to access data from a results file other than for solid elements or shell elements.

**Command(s): FS**

**GUI: Main Menu > General Postproc > Fatigue > Store Stresses > Specified Val**

Command input for this option is demonstrated by the following example:

```
FS,201,1,2,1,-2.0,21.6,15.2,4.5,0.0,0.0
FS,201,1,2,7,450.3
```

In this example, only the total stresses (items 1-6) and the temperature (item 7) are input. If the linearized stresses were also to be input, they would follow the temperatures as items 8 through 13. Time would be input at item 14.

---

#### Note

Time is not used in the fatigue calculation and is only included for reference purposes.

---

#### Note

In the special case of a beam element having only axial stress, you would input only one stress component (SX), leaving the rest of the stress fields blank.

---

### 14.3.3.1.2. Nodal Stresses from Jobname .RST

When you use this option, you cause a nodal stress vector containing six stress components to be stored directly from the results database. Stress components stored with this option can be modified with a subsequent **FS** command.

#### Note

You must issue a **SET** command, and possibly a **SHELL** command, before executing **FSNODE**. **SET** will read results for a particular load substep from the results file (Jobname .RST) into the database. **SHELL** allows you to select results from the top, middle, or bottom surfaces for shell elements (default is the top surface).

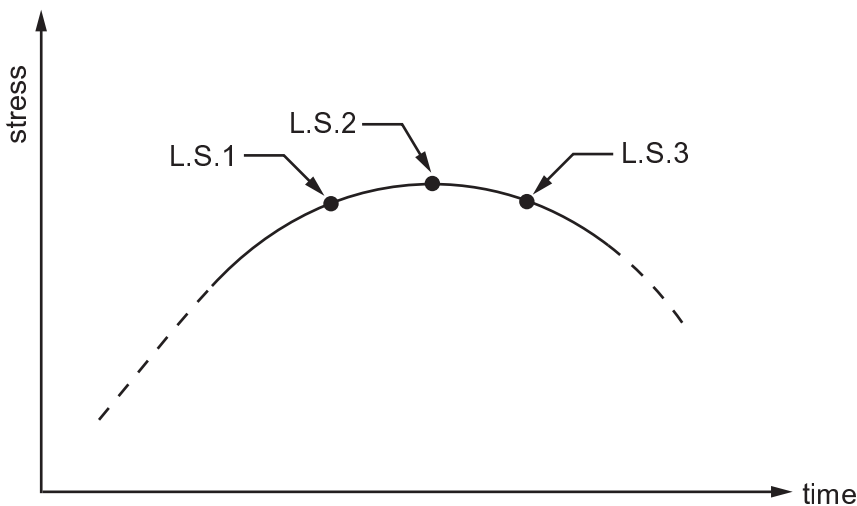
#### Command(s): **FSNODE**

GUI: Main Menu > General Postproc > Fatigue > Store Stresses > From rst File

Input by means of **FSNODE** is demonstrated by the following example for an event at one nodal location:

```
SET,1           ! Define data set for load step 1
FSNODE,123,1,1 ! Stress vector at node 123 assigned to event 1,
               ! loading 1.
SET,2           ! Define data set for load step 2
FSNODE,123,1,2 ! ...event 1, loading 2
SET,3           ! ...load step 3
FSNODE,123,1,3 ! ...event 1, loading 3
```

Figure 14.2: Three Loadings in One Event



### 14.3.3.1.3. Stresses at a Cross-Section

This option calculates and stores total linearized stresses at the ends of a section path (as defined by a preceding **PATH** and **PPATH** command). Because you will normally want the linearization to take place over a thickness representing the shortest distance between the two surfaces, use only the two surface nodes to describe the path in the **PPATH** command. This option retrieves stress information from the results database; therefore **FSSECT** must also be preceded by a **SET** command. Stress components stored with an **FSSECT** command can be modified with a subsequent **FS** command.

#### Command(s): **FSSECT**

GUI: Main Menu > General Postproc > Fatigue > Store Stresses > At Cross Sect

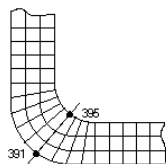
Input by means of **FSSECT** is demonstrated in the following example. If node locations are not assigned with an **FL** command, the **FSSECT** commands in this example will automatically assign location numbers to the two path nodes, 391 and 395. (See [Figure 14.3: Surface Nodes are Identified by PPATH Prior to Executing FSSECT](#) (p. 429).)

```

PATH,Name,2          ! Define the path using the two surface nodes
PPATH,1,391
PPATH,2,395
SET,1
FSSECT,,1,1        ! Store stresses at nodes 391 and 395

```

**Figure 14.3: Surface Nodes are Identified by PPATH Prior to Executing FSSECT**



### 14.3.3.2. Listing, Plotting, or Deleting Stored Stresses

Use the following options to list, plot, or delete stored stresses.

- List the stored stresses per location, per event, per loading, or per stress condition:  
**Command(s): FSLIST**  
**GUI: Main Menu > General Postproc > Fatigue > Store Stresses > List Stresses**
- Display a stress item as a function of loading number for a particular location and event:  
**Command(s): FSPLIT**  
**GUI: Main Menu > General Postproc > Fatigue > Store Stresses > Plot Stresses**
- Delete a stress condition stored for a particular location, event, and loading:  
**Command(s): FSDELE**  
**GUI: Main Menu > General Postproc > Fatigue > Store Stresses > Dele Stresses**
- Delete all stresses at a particular location:  
**Command(s): FL**  
**GUI: Main Menu > General Postproc > Fatigue > Stress Locations**
- Delete all stresses for all loadings in a particular event:  
**Command(s): FE**  
**GUI: Main Menu > General Postproc > Fatigue > Erase Event Data**

### 14.3.3.3. Assigning Event Repetitions and Scale Factors

This option assigns the number of occurrences to the event numbers (for all loadings at all locations of the event). It can also be used to apply scale factors to all of the stresses that make up its loadings.

**Command(s): FE**  
**GUI: Main Menu > General Postproc > Fatigue > Assign Events**

An example of this command usage is given below:

```

FE,1,-1          ! Erase all parameters and fatigue stresses formerly
                  ! used for event 1.
FE,2,100,1.2     ! Assign 100 occurrences to events 2, 3 and 4,
*REPEAT,3,1      ! and scale by 1.2.
FE,5,500         ! Assign 500 occurrences to event 5.

```

### 14.3.3.4. Guidelines for Obtaining Accurate Usage Factors

Structures are usually subjected to a variety of maximum and minimum stresses, which occur in unknown (or even random) order. Therefore, you must take care to achieve an accurate count of the number of repetitions of all possible stress ranges, in order to obtain a valid fatigue usage factor.

The program automatically calculates all possible stress ranges and keeps track of their number of occurrences, using a technique commonly known as the "rain flow" range-counting method. At a selected nodal location, a search is made throughout all of the events for the pair of loadings (stress vectors) that produces the most severe stress-intensity range. The number of repetitions possible for this range is recorded, and the remaining number of repetitions for the events containing these loadings is decreased accordingly. At least one of the source events will be "used up" at this point; remaining occurrences of stress conditions belonging to that event will subsequently be ignored. This process continues until all ranges and numbers of occurrences have been considered.

#### Caution

*It can be surprisingly easy to misuse the range-counting feature of the fatigue module. You must take pains to assemble events carefully if you want your fatigue evaluation to yield accurate usage factors.*

Consider the following guidelines when assembling events:

- Understand the internal logic of the range-counting algorithm. See [POST1 - Fatigue Module](#) in the *Mechanical APDL Theory Reference* for more details on how the program performs range counting.
- Because it can be difficult to predict the exact load step at which a maximum (or minimum) 3-D stress state occurs, good practice often requires that you include several clustered loadings in each event, in order to successfully capture the extreme stress state. (See [Figure 14.2: Three Loadings in One Event](#) (p. 428).)
- You will obtain consistently conservative results if you include only one extreme stress condition (either a local maximum or a local minimum) in any given event. If you group more than one extreme condition in a single event, you will sometimes generate *unconservative* results, as illustrated by the following example:

Consider a load history made up of two slightly different cycles:

Load Cycle 1: 500 repetitions of  $S_x = +50.0$  to  $-50.1$  ksi

Load Cycle 2: 1000 repetitions of  $S_x = +50.1$  to  $-50.0$  ksi

These load cycles will obviously sum to 1500 repetitions having an alternating stress intensity of about 50 ksi. However, carelessly grouping these loadings into only two events will result in an inaccurate range count. Let's see how this would happen:

Event 1,	loading 1: $S_x = 50.0$ loading 2: $S_x = -50.1$	500 repetitions
Event 2,	loading 1: $S_x = 50.1$ loading 2: $S_x = -50.0$	1000 repetitions

The possible alternating stress intensities are:

- |    |                      |           |
|----|----------------------|-----------|
| a. | From E1,L1 to E1,L2: | 50.05 ksi |
| b. | From E1,L1 to E2,L1: | 0.05 ksi  |
| c. | From E1,L1 to E2,L2: | 50.00 ksi |
| d. | From E1,L2 to E2,L1: | 50.10 ksi |
| e. | From E1,L2 to E2,L2: | 0.05 ksi  |
| f. | From E2,L1 to E2,L2: | 50.05 ksi |

Sorting these combinations by decreasing alternating stress intensity gives:

- |    |                      |           |
|----|----------------------|-----------|
| d. | From E1,L2 to E2,L1: | 50.10 ksi |
| a. | From E1,L1 to E1,L2: | 50.05 ksi |
| f. | From E2,L1 to E2,L2: | 50.05 ksi |
| c. | From E1,L1 to E2,L2: | 50.00 ksi |
| b. | From E1,L1 to E2,L1: | 0.05 ksi  |
| e. | From E1,L2 to E2,L2: | 0.05 ksi  |

The range counting then proceeds as follows:

- |    |                              |  |
|----|------------------------------|--|
| d. | 500 cycles of E1,L2 to E2,L1 | - this uses up 500 cycles of E1 and E2 |
| a. | 0 cycles of E1,L1 to E1,L2   | - E1 is all used up                    |
| f. | 500 cycles of E2,L1 to E2,L2 | - this uses up 500 more cycles of E2   |
| c. | 0 cycles of E1,L1 to E2,L2   | - both events are all used up          |
| b. | 0 cycles of E1,L1 to E2,L1   | - both events are all used up          |
| e. | 0 cycles of E1,L2 to E2,L2   | - both events are all used up          |

Thus, only 1000 repetitions of about 50 ksi range would be counted, instead of the known 1500 cycles. This error results solely from improper assembly of events.

If the loadings had each been described as separate events (such that  $E1,L1 \geq E1$ ;  $E1,L2 \geq E2$ ;  $E2,L1 \geq E3$ ; and  $E2,L2 \geq E4$ ), then the following range counts would be obtained:

- |    |                        |  |
|----|------------------------|--|
| d. | 500 cycles of E2 to E3 | - this uses up 500 cycles of E2 and E3         |
| a. | 0 cycles of E1 to E2   | - E2 is all used up                            |
| f. | 500 cycles of E3 to E4 | - uses up 500 more cycles of E3, and 500 of E4 |
| c. | 500 cycles of E1 to E4 | - uses up 500 more cycles of E4                |
| b. | 0 cycles of E1 to E3   | - E3 is all used up                            |
| e. | 0 cycles of E2 to E4   | - E2 and E4 are both all used up               |

Cumulative fatigue damage in this case would properly be calculated for 1500 repetitions of about 50 ksi range.

- Conversely, using separate events for each maximum and each minimum stress condition could sometimes become *too* conservative. In such cases, carefully choose those loadings that should be counted together, and group them into the same events. The following example illustrates how some events can appropriately contain multiple extreme stress conditions:

Consider a load history made up of these two load cycles:

- Load Cycle 1: 500 repetitions of  $S_x = +100.1$  to  $+100.0$  ksi
- Load Cycle 2: 1000 repetitions of  $S_x = +50.1$  to  $+50.0$  ksi

It is readily apparent that the worst possible combination of these cycles would yield 500 repetitions having alternating stress intensity of about 25 ksi range.

Proceeding as in the above example, grouping these loadings into two events would produce an accurate count of 500 repetitions of about 25 ksi range. Treating each loading as a separate event would yield an over-conservative count of 1000 repetitions of about 25 ksi range.

### 14.3.4. Activate the Fatigue Calculations

Now that you have locations, stresses, events, and material parameters all specified, you can execute the fatigue calculations at a specified location. The location is specified by either the location number or the node itself.

**Command(s):** **FTCALC**

**GUI:** Main Menu > General Postproc > Fatigue > Calculate Fatig

### 14.3.5. Review the Results

Fatigue calculation results are printed in the output. If you have routed your output [**/OUTPUT**] to a file (for example `Jobname. OUT`), then you can review the results by listing that file.

**Command(s):** **\*LIST**

**GUI:** Utility Menu > List > Files > Other > Jobname.OUT

If you have input an S-N curve, output is in the form of a table of alternating stress intensities (listed in decreasing order) with corresponding pairs of event/loadings, as well as cycles used, cycles allowed, temperature, and partial usage factor. Following that, the cumulative usage factor is shown for that particular location. This information is repeated for all locations.

As has been just mentioned, **FTCALC** output shows the contributing pairs of events and loadings for any given alternating stress-intensity range. This information can help you isolate the transients (events/loadings) causing the most fatigue damage.

A convenient way to modify your events would be to write all stored fatigue data on `Jobname. FATG`. (This option could be executed either before or after **FTCALC**.) Data are written to `Jobname. FATG` in terms of equivalent fatigue module commands. You can modify your events by editing `Jobname. FATG`; then use the **/INPUT** command to reread the modified fatigue commands.

**Command(s):** **FTWRITE**

**GUI:** Main Menu > General Postproc > Fatigue > Write Fatig Data

### 14.3.6. Other Approaches to Range Counting

Earlier, we discussed the "rain flow" range-counting method. This technique is useful whenever the exact time-history of various loadings is not known. However, if in your fatigue analysis the time-history is known, you can avoid the undue conservatism of this procedure simply by *running a separate fatigue analysis* [**FTCALC**] for each sequential event and then adding the usage factors manually.



### 14.3.7. Sample Input

A sample input listing for a fatigue evaluation is shown below:

```

! Enter POST1 and Resume the Database:
/POST1
RESUME,...
! Number of Locations, Events, and Loadings
FTSIZE,...
! Material Fatigue Properties:
FP,1,...      ! N values
FP,21,...     ! S values
FP,41,...     ! T values
FP,51,...     ! Sm values
FP,61,...     ! Elastic-plastic material parameters
! Locations, Stress Concentration Factors, and Location Titles
FL,...
! Store Stresses (3 Different Methods)
!   Store Stresses Manually:
  FS,...
!   Retrieve Stresses from the Results File:
  SET,...
  FSNODE,...
!   Store Stresses at a Cross-Section:
  PPATH,...
  SET,...
  FSSECT,...
! Event Repetitions and Scale Factors
FE,...
! Activate the Fatigue Calculations
FTCALC,...
! Review the Results (List the output file)
FINISH

```

See the [Command Reference](#) for a discussion of the **FTSIZE**, **FP**, **FL**, **FS**, **FSNODE**, **PPATH**, **FSSECT**, **FE**, and **FTCALC** commands.



---

## Chapter 15: Beam Analysis and Cross Sections

---

Beam elements are used to create a mathematical one-dimensional idealization of a 3-D structure. They offer computationally efficient solutions when compared to solid and shell elements.

The discussions here apply only to [BEAM188](#) and [BEAM189](#). Compared to other beam elements, these provide more robust nonlinear analysis capabilities, and significant improvements in cross-section data definition, analysis, and visualization.

For beam based code checks, also refer to the [BEAMST](#) solver.

The following topics are available for beam analysis and cross sections:

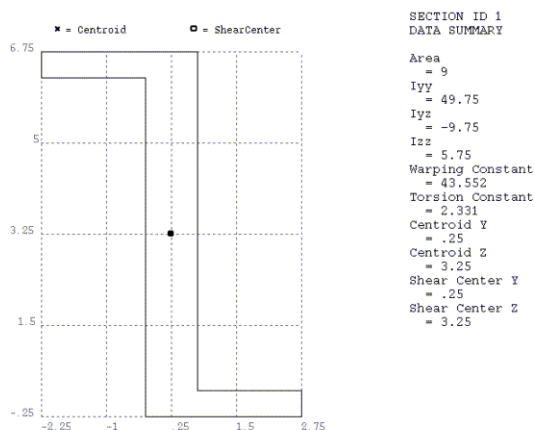
- [15.1. Overview of Cross Sections](#)
- [15.2. How to Create Cross Sections](#)
- [15.3. Creating Cross Sections](#)
- [15.4. Using Nonlinear General Beam Sections](#)
- [15.5. Using Preintegrated Composite Beam Sections](#)
- [15.6. Managing Cross Section and User Mesh Libraries](#)
- [15.7. Example Lateral Torsional Buckling Analysis](#)
- [15.8. Example Problem with Cantilever Beams](#)
- [15.9. Where to Find Other Examples](#)

### 15.1. Overview of Cross Sections

A cross section defines the geometry of the beam in a plane perpendicular to the beam axial direction. A library of eleven commonly-used beam cross section shapes is available, and user-defined cross section shapes are allowed. When a cross section is defined, the program builds a numeric model using a nine node cell for determining the properties ( $I_{yy}$ ,  $I_{zz}$ , etc.) of the section and for the solution to the Poisson's equation for torsional behavior.

[Figure 15.1: Plot of a Z Cross Section \(p. 435\)](#) shows the centroid and shear center of the cross section and the calculated section properties:

**Figure 15.1: Plot of a Z Cross Section**



Cross sections and user section meshes may be saved and stored in cross section library files. You may assign beam cross sections as attributes of a line using the **LATT** command. These section definitions will be incorporated into the generated beam elements when the line is meshed with **BEAM188** or **BEAM189**.

## 15.2. How to Create Cross Sections

The general procedure for creating cross sections consists of the following steps:

1. Define the section and associate a section ID number with the section subtype.
2. Define the geometry data for the section.

The following commands are available for creating, viewing, and listing cross sections, and for managing cross section libraries:

**Table 15.1: Cross Section Commands**

Command	Purpose
<b>PRESOL</b>	Prints section results
<b>SECTYPE</b>	Associates section <i>Subtype</i> with <i>SECID</i>
<b>SECDATA</b>	Defines section geometry data
<b>SECOFFSET</b>	Defines section offset for beam cross sections
<b>SECCONTROL</b>	Overrides program calculated properties.
<b>SECNUM</b>	Identifies the <i>SECID</i> to be assigned to an element
<b>SECPLOT</b>	Plots geometry of a beam section to scale
<b>SECWRITE</b>	Creates an ASCII file containing user mesh or a custom cross section
<b>/SECLIB</b>	Sets default section library path for <b>SECREAD</b>
<b>SECREAD</b>	Reads a user defined section library, mesh, or custom cross section
<b>SLIST</b>	Summarizes section properties
<b>SDELETE</b>	Deletes a cross section

For complete documentation of the cross section commands, see the [Command Reference](#).

### 15.2.1. Defining a Section and Associating a Section ID Number

Use the **SECTYPE** command to define a section and associate it with a section ID number. For example, the following command assigns a section identification number (2) to a predefined cross section shape (circular solid):

**Command(s):** **SECTYPE**, 2, BEAM, CSOLID

**GUI:** Main Menu > Preprocessor > Sections > Beam > Common Sectns

To define your own cross sections, use the subtype MESH. To define a cross section with integrated properties such as Iyy and Izz, use the subtype ASEC.

## 15.2.2. Defining Cross Section Geometry and Setting the Section Attribute Pointer

Use the **SECDATA** command to define the geometry of a cross section. Continuing with the example **SECTYPE** command shown above, note that the CSOLID subtype has two dimensions: the radius and the number of cells along its circumference. Thus, the **SECDATA** command shown below specifies 5 as the radius of the circular solid beam, and 8 as the number of cells along its circumference. The **SECNUM** command sets the element section attribute pointer to 2.

**Command(s):** **SECDATA**, 5, 8 and **SECNUM**, 2

**GUI:** Main Menu> Preprocessor> Sections> Beam> Common Sectns

Main Menu> Preprocessor> Meshing> Mesh> Attributes> Default Attribs

### 15.2.2.1. Determining the Number of Cells to Define

In general, you do not need to set the number of cells when building a cross section. The program calculate values for the cross section such as the area of the section and the moments of inertia about the coordinate axes using default integration rules and will produce results that are numerically exact.

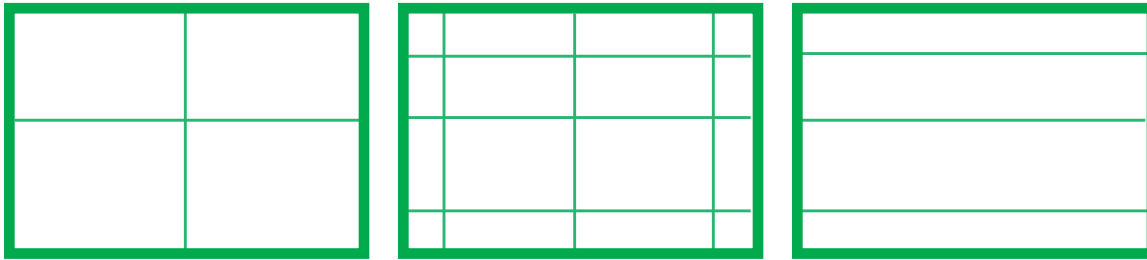
Since the torsion constant is derived from the mesh, the accuracy of the torsion constant is directly proportional to the mesh size of the cross section. The default mesh yields acceptable engineering accuracy.

There are two ways to specify the mesh size for common sections:

- invoking **SECTYPE**,,,,,*REFINEKEY* to set the level of mesh refinement for thin-walled sections (CTUBE, CHAN, I, Z, L, T, HATS, and HREC)
- specifying the number of divisions using **SECDATA** for solid sections (RECT, QUAD, and CSOLID)

The thin wall sections have a minimum of two integration points through thickness, so results produced using thin wall sections should be acceptable for materially nonlinear analysis.

However, when doing a plasticity analysis, the cell defaults may need to be changed for the solid sections. Here are examples of program-generated solid section cell meshes and the type of analysis you may wish to use them with.

**Figure 15.2: Types of Solid Section Cell Mesh**

Default 2 x 2 Mesh

4 x 4 Graded Mesh  
Captures plasticity  
through thickness and  
the onset of plasticity  
at the boundary1 x 4 Mesh  
for 2-D behavior

### 15.2.3. Meshing a Line Model with BEAM188 or BEAM189 Elements

Before you mesh a line with [BEAM188](#) or [BEAM189](#) elements, some of its attributes must be defined. These attributes include:

- The material set attribute pointer to be associated with the generated beam elements.
- The beam element type to be used in meshing the line.
- The orientation of the cross section with respect to the beam element axis. For detailed information about orientation nodes and beams, see [Generating a Beam Mesh With Orientation Nodes](#) in the *Modeling and Meshing Guide*.
- The cross section ID to be assigned to the generated beam elements.

Issue the **LATT** command to associate these attributes with the selected, unmeshed line:

**Command(s):** **LATT**, *MAT*, , *TYPE*, , *KB*, *KE*, *SECNUM*

**GUI:** Main Menu > Preprocessor > Meshing > Mesh Attributes > Picked Lines See the **LATT** command description for the meaning of *MAT* and *TYPE*.

The following arguments are described here in terms of their applicability to beams.

#### **KB**

Corresponds to any keypoint number in the model. All beam elements generated will have their beam section oriented such that the beam z-axis will lie in the plane defined by two line end keypoints and this keypoint number.

#### **KE**

Corresponds to any keypoint number in the model (optional). This keypoint determines the beam orientation at the end of the line as described above. If *KE* is given, *KB* determines the beam orientation at the beginning of the line. This is useful for creating twisted beams.

#### **SECNUM**

Corresponds to the beam section defined by the **SECTYPE** command with the section ID number as given by the *SECNUM*.

## 15.3. Creating Cross Sections

There are two main types of beam cross sections:

- common sections
- custom sections

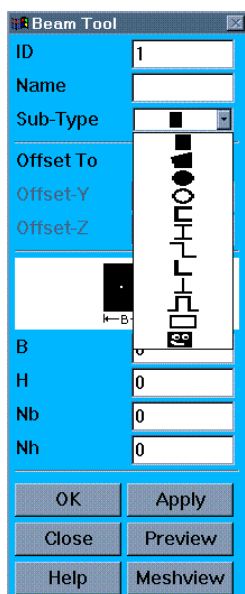
Common sections are described by a standard geometry and a single material. Custom sections are defined by an arbitrary geometry and may consist of several materials.

In addition, you can use defined sections to create tapered beams (for [BEAM188](#) and [BEAM189](#) only). See [Defining a Tapered Beam](#) (p. 441) for more information.

### 15.3.1. Using the Beam Tool to Create Common Cross Sections

The **SECTYPE**, **SECDATA**, and **SECOFFSET** commands (**Main Menu**> **Preprocessor**> **Sections**> **Beam**> **Common Sectns**) are all associated with the BeamTool in the GUI. The appearance of the BeamTool varies depending on the cross section subtype you select:

**Figure 15.3: BeamTool with Subtypes Drop Down List Displayed**



The top part of the BeamTool relates a section ID number to a subtype (and, optionally, a section name) [**SECTYPE**]. The middle of the BeamTool defines the section offset information, if needed [**SECOFFSET**]. The bottom contains the fields for section geometry information [**SECDATA**]. The dimensions defined by the **SECDATA** command are determined by the subtype selected. For documentation about a particular variant of the BeamTool, select the subtype that you want information about, and then click the **Help** button on the BeamTool. The subtype dimensions are also documented in the **SECDATA** command description.

### 15.3.2. Creating Custom Cross Sections with a User-defined Mesh

If you need to define a cross section that is not common, you must create a user mesh file. To create a user mesh file, create a 2-D solid model and save it using the **SECWRITE** command (**Main Menu**> **Preprocessor**> **Sections**> **Beam**> **Custom Sectns**> **Write From Areas**). This procedure is outlined in greater detail below:

1. Create all areas (**Main Menu**> **Preprocessor**> **Modeling**> **Create**> **Areas**).

2. Overlap the areas (**Main Menu> Preprocessor> Modeling> Operate> Booleans> Overlap> Areas**) or glue them (**Main Menu> Preprocessor> Modeling> Operate> Booleans> Glue> Areas**) where appropriate.
3. Save the model.
4. Set the number of line divisions for all lines (**Main Menu> Preprocessor> Meshing> Size Cntrl> ManualSize> Lines> Picked Lines** or use the MeshTool).
5. Select **Main Menu> Preprocessor> Sections> Beam> Custom Sectns> Write From Areas**. A picker appears. Pick the area(s) of the cells.
6. The program creates cells on the areas. Bad shape messages may appear during the mesh, but you can ignore them. You may, however, see an "Unable to mesh area ...." message; if you do, clear the elements from all areas (**Main Menu> Preprocessor> Meshing> Clear> Areas**) and repeat steps 4 and 5.
7. Write the .SECT file out to a unique name in the Write Section Library File dialog and click **OK**.
8. Read in the user mesh file (**Main Menu> Preprocessor> Sections> Beam> Custom Sectns> Read Sect Mesh**) to calculate the section properties. Material properties must be defined to calculate the cross section shear correction factors, material-weighted centroids and the shear centers.

### 15.3.2.1. Line Element Size

Even if you have already set the line element size (**LESIZE**), you will see the following message:

**Line element sizes may need to be specified for desired cross-section mesh. Please refer to the LESIZE command.**

If you have already set the line element size, click the **Close** button to continue. If you have not already set it, issue **LESIZE** with the appropriate information.

When creating cells on the area using the GUI, it is not necessary to define a plane element type. On the other hand, you must define a plane element type and mesh the area if you issue the **SECWRITE** command explicitly. **MESH200** with **KEYOPT(1) = 7** and **PLANE183** are the only valid plane element types.

### 15.3.3. Creating Custom Cross Sections with Mesh Refinement and Multiple Materials

When performing a plasticity analysis, you may need to refine the cross section mesh. A cross section consisting of more than one material may be defined to represent layers, reinforcements or sensors. When defining a multiple-material cross section, you will need to specify the material that each cross section cell is made of. You can take a previously-created cross section and modify it.

To create a custom cross section with a refined mesh and/or multiple materials, perform the following tasks:

1. Either read a common section (**Main Menu> Preprocessor> Sections> Beam> Custom Sectns> Edit/Built-up> Edit Common**) from the database or read a custom section from a .SECT file. (**Main Menu> Preprocessor> Sections> Beam> Custom Sectns> Edit/Built-up> Edit Custom**)
2. The Graphics window displays a **MESH200** plot.



3. Refine the section mesh. (**Main Menu**> **Preprocessor**> **Sections**> **Beam**> **Custom Sectns**> **Edit/Built-up**> **Refine Mesh**)
4. Modify the cell materials. (**Main Menu**> **Preprocessor**> **Sections**> **Beam**> **Custom Sectns**> **Edit/Built-up**> **Modify Material**) If you want to create a multiple-material section, define the materials. This is necessary to calculate the shear correction factors and material-weighted centroids.
5. Save the section to a .SECT file using **SECWRITE** (**Main Menu**> **Preprocessor**> **Sections**> **Beam**> **Custom Sectns**> **Edit/Built-up**> **Save**).
6. To calculate the section properties and use a custom section in the analysis, read in the user mesh file (**Main Menu**> **Preprocessor**> **Sections**> **Beam**> **Custom Sectns**> **Read Sect Mesh**).

---

### Note

If you redefine a material *after* creating the section, you must reissue the **SECTYPE** and **SECREAD** commands to recalculate the cross section.

When a cross section has multiple materials, and **/ESHape** is used to produce contour plots of stresses (and other quantities), the element averages the stresses across material boundaries. To limit this behavior, use small cross section cells around the material boundaries.

When using the **SECWRITE** command explicitly, a section with multiple materials can be created by assigning material attributes to the areas. During meshing, the elements inherit the material attribute from the area and this attribute is written to the .SECT file.

A material number (**MAT**) specified for a custom section overrides any existing material setting.

---

## 15.3.4. Defining Composite Cross Sections

A composite cross section is a cross section consisting of at least two materials and meshed with a user-defined mesh. When creating a composite section, define the materials before running the **SECREAD** command. This is necessary to calculate the shear correction factors, material-weighted centroids and the shear centers.

---

### Note

If you redefine a material after creating the section, you must reissue the **SECREAD** command to recalculate the cross section.

---

You can save composite cross sections as custom cross sections. For information on custom cross sections, see [Using the Beam Tool to Create Common Cross Sections](#) (p. 439)

## 15.3.5. Defining a Tapered Beam

For **BEAM188** and **BEAM189**, you can define tapered beams using the **TAPER** option on the **SECTYPE** command (**Main Menu**> **Preprocessor**> **Sections**> **Beam**> **Taper Sections**). The tapered section varies linearly between two specified locations, (x1, y1, z1) and (x2, y2, z2). Thus, two **SECData** commands are required to define the taper as shown below. At each of these end points, a valid beam section ID must be input (station-1 and station-2 in the example below).

**SECTYPE**,*SECID*,**TAPER**

**SECDATA**,station-1,x1,y1,z1

**SECDATA**,station-2,x2,y2,z2

The following is a typical command input stream used to create a tapered cross section:

```
sectype,1,beam,rect      ! define cross section at first end point
secdata,.0001,0.5
sectype,2,beam,rect      ! define cross section at far end
secdata,3,0.5
sectype,3,taper          ! new Section ID for tapered beam analysis
secdata,1,0.0,0.0        ! section 1 at location (0,0,0)
secdata,2,0.0,20.0       ! section 2 at location (0,20,0)
```

Continuing with this example, you can then use 3 as the taper section ID when assigning mesh attributes with the **SECNUM** or **LATT** command. The resulting beam cross section is (0.0001 \* 0.5) at end 1, and linearly tapers to (3 \* 0.5) at end 2.

The following assumptions apply to tapered beams defined with this method:

- The end sections must be defined prior to defining the taper.
- Sections defined at the end points must be topologically identical.
- A section cannot taper to a point (or zero area) at either end.
- The arbitrary beam section type (**SECTYPE**,,BEAM,ASEC) assumes a linear variation of properties along the length; large differences in properties between the ends are not recommended.

The program performs a number of checks (although not completely comprehensive) to ensure topological equality. The following items are compared:

- Number of section nodes
- Number of section elements
- Section type

If both end stations refer to custom cross sections with multiple materials, you must ensure that material IDs for the cells are the same for both ends.

At a Gauss point of integration, the **BEAM188** and **BEAM189** elements will find the closest point on the line defined by station-1 and station-2. Using this information, a linear interpolation is performed for the cross section mesh. The Gauss point must therefore be located within the end points. A Gauss point within a specified tolerance (**SELTOL**,*TOLER*) is considered to be coincident with an end station. The default tolerance value is 1.0E-8.

The tapered section treatment is significantly more expensive than a constant cross section (since re-computation is necessary). If this is a concern, use **KEYOPT**(12) of the beam element to specify the tapered section treatment.

- **KEYOPT**(12) = 0 - Linear tapered section analysis (more accurate, but expensive)
- **KEYOPT**(12) = 1 - Average cross section analysis (an approximation of the order of the mesh size, but faster)

## 15.4. Using Nonlinear General Beam Sections

The *nonlinear general beam section* (**SECTYPE**,,GENB) is an abstract cross section type that allows you to define axial, flexural, torsional, and transverse shear behavior as a function of axial strain, bending curvature, twist, and transverse shear strains.

The generalized section form of input does not require cross section geometry data or material data independently. For evaluating mass matrices, the program assumes a unit area of cross section. This form of data is useful for including an experimentally measured nonlinear response of a beam-like structural component, or for including complex behavior such as cross section distortion (not possible when using normal beam sections).

Nonlinear general beam sections also allow a nonlinear relationship of transverse shear forces to the corresponding transverse shear strains. Often, the input of generalized beam sections may be a result of a prior detailed slice analysis (for example, a segment of pipe analyzed using generalized plane strain elements).

The behavior of beam elements is governed by the generalized-stress/generalized-strain relationship of the form:

$$\begin{bmatrix} N \\ M_1 \\ M_2 \\ \tau \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} AE(\varepsilon, T) & & & & & \\ & I_1 E(\kappa_1, T) & & & & \\ & & I_2 E(\kappa_2, T) & & & \\ & & & JG(\chi, T) & & \\ & & & & A_1 G(\gamma_1, T) & \\ & & & & & A_2 G(\gamma_2, T) \end{bmatrix} \begin{bmatrix} \varepsilon \\ \kappa_1 \\ \kappa_2 \\ \chi \\ \gamma_1 \\ \gamma_2 \end{bmatrix}$$

where:

$N$  = Axial force

$M_1$  = Bending moment in plane XZ

$M_2$  = Bending moment in plane XY

$\tau$  = Torque

$S_1$  = Transverse shear force in plane XZ

$S_2$  = Transverse shear force in plane XY

$\varepsilon$  = Axial strain

$\kappa_1$  = Curvature in plane XZ

$\kappa_2$  = Curvature in plane XY

$\chi$  = Twist of the cross section

$\gamma_1$  = Transverse shear strain in plane XZ

$\gamma_2$  = Transverse shear strain in plane XY

$A_E(\varepsilon, T)$  = Axial stiffness as a function of axial strain and temperature

$I_1^E(\kappa_1, T)$  = Flexural rigidity as a function of curvature and temperature in plane XZ

$I_2^E(\kappa_2, T)$  = Flexural rigidity as a function of curvature and temperature in plane XY

$J_G(\chi, T)$  = Torsional rigidity, as a function of torsion and temperature

$A_1^G(\gamma_1, T)$  = Transverse shear stiffness as a function of shear strain and temperature in plane XZ

$A_2^G(\gamma_2, T)$  = Transverse shear stiffness as a function of shear strain and temperature in plane XY  
 $T$  is the current temperature

Thermal expansion coefficients and mass density for the section as a function of temperature complete the definition of a generalized cross section.

### 15.4.1. Defining a Nonlinear General Beam Section

Each of the following commands specifies a particular component quantity necessary for defining a nonlinear general beam section:

**Table 15.2: Commands for Specifying Nonlinear General Beam Section Data**

Command	Quantity Defined and Data Specified
<b>BSAX</b> [1]	Axial strain and force $\epsilon, N, T$
<b>BSM1</b> [1]	Bending curvature and moment in plane XZ $\kappa_1, M_1, T$
<b>BSM2</b> [1]	Bending curvature and moment in plane XY $\kappa_2, M_2, T$
<b>BSTQ</b> [1]	Cross section twist and torque $\chi, \tau, T$
<b>BSS1</b> [1]	Transverse shear strain and force in plane XZ $\gamma_1, S_1, T$
<b>BSS2</b> [1]	Transverse shear strain and force in plane XY $\gamma_2, S_2, T$
<b>BSMD</b> [2]	Mass density of the beam section (assuming a unit area) $DENS$
<b>BSTE</b> [2]	Thermal expansion coefficient $ALPHA$

1. Repeatable for six independent temperatures, and up to 20 strain values.
2. Repeatable for six different temperatures.

**Temperature dependencies ( $T$ )** You can define each of the generalized section data components as temperature-dependent. It is possible to specify up to six temperatures ( $T$ ) by reissuing any command as necessary. If you issue a command for a temperature specified earlier, the most recent data supersedes the previous value. Values outside of the temperature range cannot be used.

#### 15.4.1.1. Strain Dependencies

Each component of a nonlinear beam section definition (axial, bending, torque, and transverse shear) can be a nonlinear function of the corresponding strain. The terms *generalized stress* and *generalized strain* describe the data defined via the **BSAX**, **BSM1**, **BSM2**, **BSTQ**, **BSS1**, and **BSS2** commands.

The generalized stress to generalized strain relationship can be nonlinear. The nonlinear response can be either purely elastic--that is, no permanent deformation, and fully recoverable deformation even though the behavior is nonlinear--or elasto-plastic. The option of a purely elastic or elasto-plastic response (**SECTYPE**) applies to all components of a beam section definition.

The following input illustrates a typical (temperature-independent) nonlinear axial behavior:

```
sectype,1,genb,plastic ! Elasto-plastic response
bsax ,0.0008,200      ! Axial strain 0.0008, Force of 200
bsax ,0.001 ,240      ! Axial strain 0.001 , Force of 240
bsax ,0.0014,300      ! Axial Strain 0.0014, Force of 300
```

The range of strain values must cover the anticipated maximum deformation of the structure. For all nonlinear beam section specifications, the stiffness disappears if the structural response is beyond the maximum strain value of the section data. You can define a maximum of 20 generalized stress-strain points at each temperature value, and the successive slopes must be smaller than the first slope. Isotropic hardening is assumed for the material response.

For a plastic nonlinear beam section subtype (**SECTYPE**,,GENB,PLASTIC), you must define *all* section components with two or more stress-strain points. However, you can define a *linear* behavior for any section component by specifying a larger maximum strain value. Following is an example where axial behavior is nonlinear, but bending response is linear:

```
sectype,1,genb,plastic ! Elasto-plastic response
bsax,0.0008,200
bsax,0.001 ,240
bsax,0.0014,300
! bending
bsml,0.1,10000
bsml,1,100000
```

For an elastic nonlinear beam section subtype (**SECTYPE**,,GENB,ELASTIC), a single stress-strain point is adequate for defining linear behavior.

## 15.4.2. Considerations for Using Nonlinear General Beam Sections

You can define nonlinear general beam sections only when using element **BEAM188** or **BEAM189**. When using [nonlinear beam section](#) data, the following conditions apply:

- The section data defined by each command listed in [Table 15.2: Commands for Specifying Nonlinear General Beam Section Data \(p. 444\)](#) is associated with the section most recently defined via the **SECTYPE** command.
- Beam stresses are not available for output; however, the stress resultants are available as **ETABLE** quantities.
- Section offsetting (**SECOFFSET**) is not available.
- General beam sections do not support [linear perturbation](#).
- Only the temperature of the beam axis is relevant.

Beam section rotary inertia is calculated internally based on axial and bending stiffnesses provided. If beam section rotary inertia of a more general form is available, use **MASS21** elements (rather than the **BSMD** command) to define beam section mass and rotary inertia at the beam nodes.

## 15.5. Using Preintegrated Composite Beam Sections

The behavior of beam elements is governed by the beam-section stiffness matrix and the section mass matrix. A [generalized nonlinear beam section](#) (**SECTYPE**,,GENB) allows only diagonal terms in the cross-section stiffness matrix defining the relationship between generalized stress and generalized strains; it is therefore suitable only for beams with homogeneous and isotropic materials where the coupling between different generalized strains is small. For composite beams made of dissimilar constituent

materials, or homogeneous beams with orthotropic material and material orientation not parallel to the beam axis, the coupling between different generalized strains can be significant and generally leads to full cross-section stiffness matrix.

The *preintegrated composite beam section* (**SECTYPE**,**COMB**,**MATRIX**) is an abstract cross section type that allows you to define a fully populated but symmetrical cross-section stiffness and mass matrix directly. You can use preintegrated composite beam sections when using **BEAM188** or **BEAM189** elements, provided that linear elastic material behavior is acceptable.

The full cross-section stiffness relates the generalized-stress to generalized-strain in the following form:

$$\begin{bmatrix} N \\ M_1 \\ M_2 \\ \tau \\ S_1 \\ S_2 \\ B_m \end{bmatrix} = \begin{bmatrix} S_{11}(T) & S_{12}(T) & S_{13}(T) & S_{14}(T) & S_{15}(T) & S_{16}(T) & S_{17}(T) \\ & S_{22}(T) & S_{23}(T) & S_{24}(T) & S_{25}(T) & S_{26}(T) & S_{27}(T) \\ & & S_{33}(T) & S_{34}(T) & S_{35}(T) & S_{36}(T) & S_{37}(T) \\ & & & S_{44}(T) & S_{45}(T) & S_{46}(T) & S_{47}(T) \\ & & & & S_{55}(T) & S_{56}(T) & S_{57}(T) \\ & & & & & S_{66}(T) & S_{67}(T) \\ & & & & & & S_{77}(T) \end{bmatrix} \begin{bmatrix} \epsilon \\ \kappa_1 \\ \kappa_2 \\ \chi \\ \gamma_1 \\ \gamma_2 \\ B_k \end{bmatrix}$$

where

- N = Axial force
- M<sub>1</sub> = Bending moment in plane XZ
- M<sub>2</sub> = Bending moment in plane XY
- τ = Torque
- S<sub>1</sub> = Transverse shear force in plane XZ
- S<sub>2</sub> = Transverse shear force in plane XY
- B<sub>m</sub> = Warping bi-moment
- ε = Axial strain
- κ<sub>1</sub> = Curvature in plane XZ
- κ<sub>2</sub> = Curvature in plane XY
- χ = Twist of the cross section
- γ<sub>1</sub> = Transverse shear strain in plane XZ
- γ<sub>2</sub> = Transverse shear strain in plane XY
- B<sub>k</sub> = Warping bi-curvature
- S<sub>ij</sub>(T) (where i = 1,7 and j = i,7) = Stiffness constants in the upper triangle of the cross-section stiffness matrix as a function of temperature
- T = the current temperature

With a unit beam length, the section mass matrix relates the resultant forces and torques to accelerations and angular accelerations as follows (applicable to the local element coordinate system):

$$\begin{bmatrix} N_x \\ N_y \\ N_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} C_{11}(T) & C_{12}(T) & C_{13}(T) & C_{14}(T) & C_{15}(T) & C_{16}(T) \\ & C_{22}(T) & C_{23}(T) & C_{24}(T) & C_{25}(T) & C_{26}(T) \\ & & C_{33}(T) & C_{34}(T) & C_{35}(T) & C_{36}(T) \\ & & & C_{44}(T) & C_{45}(T) & C_{46}(T) \\ & & & & C_{55}(T) & C_{56}(T) \\ & & & & & C_{66}(T) \end{bmatrix} \begin{bmatrix} A_x \\ A_y \\ A_z \\ R_x \\ R_y \\ R_z \end{bmatrix}$$

where

$N_x$  = Axial resultant force

$N_y$  = Resultant force in beam Y coordinate direction

$N_z$  = Resultant force in beam Z coordinate direction

$M_x$  = Torque about the beam axes

$M_y$  = Torque about the beam Y coordinate direction

$M_z$  = Torque about the beam Z coordinate direction

$A_x$  = Axial acceleration

$A_y$  = Acceleration in beam Y coordinate direction

$A_z$  = Acceleration in beam Z coordinate direction

$R_x$  = Angular acceleration about beam axes

$R_y$  = Angular acceleration about beam Y coordinate direction

$R_z$  = Angular acceleration about beam Z coordinate direction  $C_{ij}(T)$  (where  $i=1,7$  and  $j = i,7$ ) = mass or moment of inertia terms in the upper triangle of the cross-section mass matrix as a function of temperature

$T$  = Current temperature

### 15.5.1. Defining a Composite Beam Section

Each of the following commands specifies a particular component quantity necessary for defining a preintegrated composite beam section:

**Table 15.3: Commands for Specifying Preintegrated Composite Beam Section Data**

Command	Quantity Defined and Data Specified
<b>CBMX</b> [1]	Cross-section stiffness matrix. Upper triangle of 6 x 6 or 7 x 7 matrix. Row and column 7 are used only when KEYOPT(1) = 1.
<b>CBMD</b> [1]	Cross-section mass matrix. Upper triangle of 6 x 6 matrix.
<b>CBTMP</b> [1]	Temperature specification for a subsequent matrix.  $T$
<b>CBTE</b> [1]	Thermal expansion coefficient  $ALPHA$

1. Repeatable for six independent temperatures.

**Temperature dependencies (T)** You can define each of the preintegrated composite beam section data components as temperature-dependent. It is possible to specify up to six temperatures (T) by reissuing any command as necessary. If you issue a command for a temperature specified earlier, the most recent data supersedes the previous value.

#### 15.5.1.1. Matrix Input

Each component (axial, bending, torque, transverse shear, and warping) of a preintegrated beam section definition is input directly as a 7 x 7 matrix. The terms of the matrix are defined via the **CBMX** command.

The following input example uses the **CBMX** command to define a (temperature-independent) parameterized cross-section stiffness matrix for a circular solid cross section with unit radius:

```
Exx = 30e6
Nuxy = 0.3
Gxy = 0.5*Exx / (1+Nuxy)

Radius = 1
Pi = 3.14
Xarea = Pi * Radius**2
Izz = Pi/4 * Radius**4
Iyy = Pi/4 * Radius**4
J = Izz + Iyy

sectype,1,comb,matrix
cbmx,1, Exx*Xarea
cbmx,2, Exx*Iyy
cbmx,3, Exx*Izz
cbmx,4, Gxy*J
cbmx,5, Gxy*Xarea
cbmx,6, Gxy*Xarea
cbmx,7, 0
```

## 15.5.2. Considerations for Using Composite Beam Sections

You can define [preintegrated composite beam sections](#) when using element [BEAM188](#) or [BEAM189](#) elements. The following conditions apply:

- The section data defined by each command listed in [Table 15.3: Commands for Specifying Preintegrated Composite Beam Section Data \(p. 447\)](#) is associated with the section most recently defined (**SECTYPE**).
- Beam stresses are not available for output; however, the stress resultants are available as **ETABLE** quantities.
- Section offsetting (**SECOFFSET**) is not available.
- The preintegrated stiffness components must yield a positive definite section stiffness at analysis time.
- Only the temperature of the beam axis is relevant.

## 15.5.3. Example: Composite Beam Section Input

The following input example compares the results of a preintegrated composite beam section to an offset channel section:

```
/prep7

/com, Approximate material properties of steel in U.S. customary units
Exx = 30e6
nuxy = 0.3
gxy = 0.5*Exx / (1+nuxy)

/com, Parameterize length and number of elements to more easily study
/com, the effects of discretization and length-to-thickness ratio
leng = 25
nelem = 1

mp,ex,1,Exx
mp,prxy,1,nuxy
mp,gxy,1,gxy

et,1,188,1,,3      ! Include warping and use the cubic shape function

offy = -8
```



```

offz = -10

/com,
/com, Define a beam channel section. Unequal lengths and thicknesses are
/com, used to demonstrate an unsymmetric section with noncoincident
/com, centroid and shear center.
/com,
/com, An arbitrary nodal offset is used to demonstrate coupling between
/com, axial and bending stiffness, as well as coupling between shear
/com, and torsional stiffness.
/com,
sectype,1,beam,chan
secdata,1,1.5,2, 0.1, 0.15, 0.2
secoffset,user,offy,offz

/com, For the channel section, generate elements along the x-axis
secnum,1
n,1
*do,i,1,nelem
loc = i*leng/nelem
n,i+1,loc
e,i,i+1
*enddo

/com,
/com, Retrieve the essential calculated and geometric properties from
/com, the channel section
/com,
*get,xarea,secp,1,prop,area
*get,iyy,secp,1,prop,iyy
*get,izz,secp,1,prop,izz
*get,iyz,secp,1,prop,iyz
*get,J,secp,1,prop,tors
*get,iw,secp,1,prop,warp
*get,scyy,secp,1,prop,scyy
*get,sczz,secp,1,prop,sczz
*get,cgy,secp,1,prop,cgy
*get,cgz,secp,1,prop,cgz
*get,offy,secp,1,prop,offy
*get,offz,secp,1,prop,offz
*get,shcy,secp,1,prop,shcy
*get,shcz,secp,1,prop,shcz

yc = offy - cgy      ! Distance between nodal offset and centroid
zc = offz - cgz
ys = offy - shcy    ! Distance between nodal offset and shear center
zs = offz - shcz
yssq = ys*ys
zssq = zs*zs

/com,
/com, Adjust the preintegrated torsional stiffness based on nodal offset
k44 = Gxy*J
k44 = k44 + ys*ys*Gxy*xarea*sczz
k44 = k44 + zs*zs*Gxy*xarea*scyy

/com,
/com, Use the material and channel section properties to define the 7x7
/com, upper triangle of the preintegrated section stiffness matrix
/com,
sectype,2,comb,matrix
cbmx,1, Exx*xarea, -zc*Exx*xarea, yc*Exx*xarea
cbmx,2, Exx*iyy+zc*zc*Exx*xarea, -Exx*Iyz-yc*zc*Exx*xarea
cbmx,3, Exx*izz+yc*yc*Exx*xarea
cbmx,4, k44, -ys*Gxy*xarea*sczz, zs*Gxy*xarea*scyy
cbmx,5, Gxy*xarea*sczz
cbmx,6, Gxy*xarea*scyy
cbmx,7, Exx*iw

slist

/com,

```

```
/com, For the preintegrated section, generate elements parallel to
/com, the x-axis and offset from the beam with channel section
secn,2
n,1001,,10
*do,i,1,nelem
loc = i*leng/nelem
n,i+1001,loc,10
e,i+1000,i+1001
*enddo

/com,
/com, Load the beams as cantilevers. Fix one end and apply a
/com, transverse load in the z-direction at the free end

d,1,all
f,nelem+1,fz,-100

d,1001,all
f,nelem+1001,fz,-100

fini

/sol
solve
finish

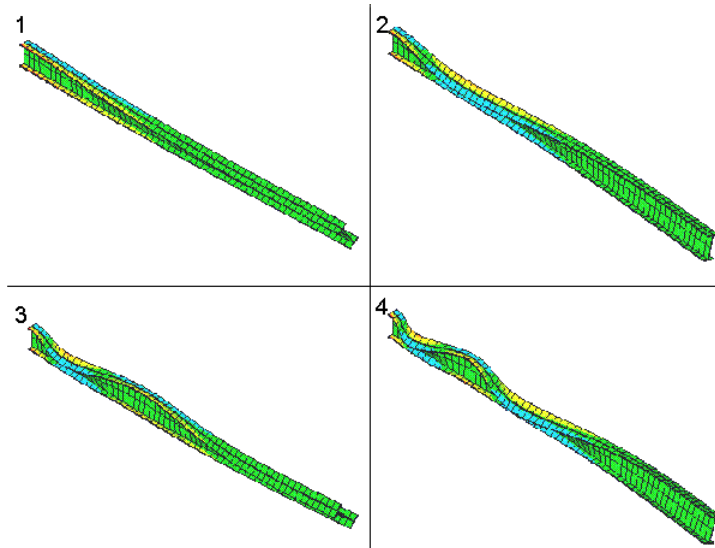
/post1
/com,
/com, COMPARE RESULTS: All displacements, including warping, compare
/com, well between beam and preintegrated sections
/SHOW,JPEG
prnsol,u
prnsol,rot
prnsol,warp
finish
```

## 15.6. Managing Cross Section and User Mesh Libraries

Cross section data for common sections such as CHAN and RECT can be stored in cross section libraries. To create standard cross sections for later use, create one or more cross sections, edit the Jobname.LOG file, and copy the appropriate **SECTYPE**, **SECDATA**, and **SECOFFSET** commands into a separate file with a SECT extension. These predefined cross sections can later be read into a model using the **/SECLIB** command (**Main Menu**> **Preprocessor**> **Sections**> **Section Library**> **Import Library**).

## 15.7. Example Lateral Torsional Buckling Analysis

You can use **BEAM188** and **BEAM189** elements to model not only straightforward beam bending and shear response but also to model beam response that involves lateral-torsional buckling. To create this type of model, you will need to create an adequately fine mesh of beam elements. You typically need to model a single beam member using a series of short beam elements, as shown in [Figure 15.4: Lateral-Torsional Buckling of a Cantilever I-Beam](#) (p. 451).

**Figure 15.4: Lateral-Torsional Buckling of a Cantilever I-Beam**

Lateral-Torsional Buckling of a Cantilever I-Beam, Modeled With 60 **BEAM188** Elements  
(Displayed Using **/ESHAPE**)

[Buckling Analysis](#) (p. 179) in the *Structural Analysis Guide* documents buckling analysis in detail. This sample problem shows what happens when a cantilever beam is subjected to a concentrated end load, which causes lateral-torsional buckling.

### 15.7.1. Problem Description

A straight, slender cantilever beam has one fixed end and one free end. A load is applied to the free end. The model is analyzed using eigenvalue buckling calculations, followed by a nonlinear load versus deflection study. The objective is to determine the critical value of the end load (indicated by P in [Figure 15.5: Diagram of a Beam With Deformation Indicated](#) (p. 452)) at which the beam undergoes a bifurcation indicated by a large displacement in the lateral direction.

### 15.7.2. Problem Specifications

The following material properties are used for this problem:

Young's modulus =  $1.0 \times 10^4$  psi

Poisson's ratio = 0.0

The following geometric properties are used for this problem:

L = 100 in

H = 5 in

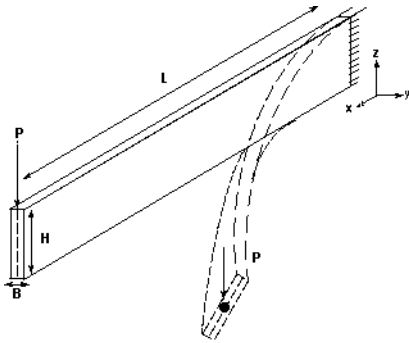
B = .2 in

Loading for this problem is:

P = 1 lb.

### 15.7.3. Problem Sketch

Figure 15.5: Diagram of a Beam With Deformation Indicated



### 15.7.4. Eigenvalue Buckling and Nonlinear Collapse

Eigenvalue buckling calculation is a linearized calculation, and is generally valid only for elastic structures. The yielding of materials occurs usually at loads lesser than that predicted by eigenvalue buckling analysis. This type of analysis tends to need less computation time than a full nonlinear buckling analysis.

You can also perform a nonlinear load versus deflection study, which employs an arc length solution strategy to identify critical loads. While the approach is more general, a collapse analysis may be computationally intensive.

The nonlinear collapse analysis must be performed on a structure with imperfections built in to the model, since a perfect model will not show signs of buckling. You can add imperfections by using eigenvectors that result from an eigenvalue buckling analysis. The eigenvector determined is the closest estimate of the actual mode of buckling. The imperfections added should be small when compared to a typical thickness of the beam being analyzed. The imperfections remove the sharp discontinuity in the load-deflection response. It is customary to use one to ten percent of the beam/shell thickness as the maximum imperfection introduced. The **UPGEOM** command adds displacements from a previous analysis and updates the geometry to the deformed configuration.

### 15.7.5. Set the Analysis Title and Define Model Geometry

1. Choose menu path **Utility Menu > File > Change Title**.
2. Enter the text "Lateral Torsional Buckling Analysis" and click **OK**.
3. Start the model creation preprocessor and define the keypoints for the beam. Choose menu path **Main Menu > Preprocessor > Modeling > Create > Keypoints > In Active CS**, and enter these keypoint numbers and the coordinates in the dialog as indicated:

Keypoint Number	X Location	Y Location	Z Location	Click This Button to Accept Values
1	0	0	0	<b>Apply</b>
2	100.0	0	0	<b>Apply</b>
3	50	5	0	<b>OK</b>

4. Create a straight line through keypoints 1 and 2. Choose menu path **Main Menu> Preprocessor> Modeling> Create> Lines> Lines> Straight Line**. The Create Straight Line picker appears. Select keypoints 1 and 2 in the Graphics window and click **OK** in the Create Straight Line picker.
5. Save the model. Choose menu path **Utility Menu> File> Save As**. Enter `buckle.db` in the "Save Database to" field and click **OK**.

### 15.7.6. Define Element Type and Cross Section Information

1. Choose menu path **Main Menu> Preferences** and select the "Structural" check box. Click **OK** to continue.
2. Choose menu path **Main Menu> Preprocessor> Element Type> Add/Edit/Delete**. The Element Types dialog appears.
3. Click Add ... The Library of Element Types dialog appears.
4. In the scroll box on the left, select "Structural Beam."
5. In the scroll box on the right, click "3-D finite strain, 3 node 189" to select **BEAM189**.
6. Click **OK**, and then click **Close** in the Element Types dialog.
7. Define a rectangular cross section for the beam. Choose menu path **Main Menu> Preprocessor> Sections> Beam> Common Sectns**. The BeamTool is displayed. The section ID is set to 1, and the subtype to RECT (signified by a rectangle on the subtype button) by default. Since you will be creating a rectangular cross section, there is no need to change the subtype.
8. In the lower half of the BeamTool, you will see a diagram of the cross section shape with dimension variables labeled. Enter the width of the cross section, 0.2, in the field labeled B. Enter the height of the cross section, 5.0, in the field labeled H. Click **Apply** to set the cross section dimensions.
9. Use the BeamTool to display information about the cross section. Click **Preview** on the BeamTool. A diagram and data summary of the cross section appear in the Graphics window. You can also preview the mesh of the cross section by selecting the **Meshview** button. Click **Close** in the BeamTool to continue.

### 15.7.7. Define the Material Properties and Orientation Node

1. Choose menu path **Main Menu> Preprocessor> Material Props> Material Models**. The Define Material Model Behavior dialog appears.
2. In the Material Models Available window on the right, double-click the following: **Structural, Linear, Elastic, Isotropic**. A dialog appears.
3. Enter 1E4 for EX (Young's modulus).
4. Enter 0.0 for PRXY (Poisson's ratio), and click **OK**. Material Model Number 1 appears in the Material Models Defined window on the left.
5. Choose menu path **Material> Exit** to close the Define Material Model Behavior dialog.
6. Replot the line by choosing menu path **Utility Menu> Plot> Lines**.
7. Select the line and define the orientation node of the line as an attribute. Choose menu path **Main Menu> Preprocessor> Meshing> Mesh Attributes> Picked Lines**. The Line Attributes picker appears. Select the line in the Graphics window and click **Apply** in the Line Attributes picker.

8. The Line Attributes dialog appears. The program includes the material attribute pointer to the material set 1, the element type attribute pointer to the local element type 1, and the section attribute pointer to the section ID 1 by default. Click on the radio button beside the Pick Orientation Keypoint(s) label to change it to Yes and click **OK**.
9. The Line Attributes picker reappears. Type 3 in the picker, press the **Enter** key, and click **OK**.
10. Save the model. Choose menu path **Utility Menu> File> Save As**. If the `buckle.db` file is not already selected, select it. Select **OK**, and when prompted whether you want to overwrite the existing file, click **OK**.

### 15.7.8. Mesh the Line and Verify Beam Orientation

1. Define the mesh size and number of divisions. Choose menu path **Main Menu> Preprocessor> Meshing> Size Cntrl> ManualSize> Lines> All Lines**. Enter 10 in the "No. of Element Divisions" field and click **OK**.
2. Mesh the line. Choose menu path **Main Menu> Preprocessor> Meshing> MeshTool**. Click MESH on the MeshTool and the Mesh Lines picker appears. Pick the line in the Graphics window, and then click **OK** in the Mesh Lines picker. Click **Close** in the MeshTool to close it.
3. Rotate the meshed line. Choose menu path **Utility Menu> PlotCtrls> Pan, Zoom, Rotate**. The **Pan, Zoom, Rotate** tool appears. Select **ISO** and click **Close**. The beam is rotated in the Graphics window.
4. Verify the beam orientation. Choose menu path **Utility Menu> PlotCtrls> Style> Size and Shape**. Select the radio button next to the **/ESHAPE** label to turn **/ESHAPE** on and click **OK**.

### 15.7.9. Define the Boundary Conditions

1. Define a boundary condition to the fixed end. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On Keypoints**. The Apply U,ROT on KPs picker appears.
2. Define keypoint 1 as the fixed end. In the picker, type 1, press the **Enter** key, then click **OK**. The Apply U,ROT on KPs dialog appears.
3. Click "All DOF" to select it, and click **OK**. The boundary condition information appears in the Graphics window at keypoint 1.
4. Apply a force to the free end. Choose menu path **Main Menu> Solution> Define Loads> Apply> Structural> Force/Moment> On Keypoints**. The Apply F/M on KPs picker appears.
5. Identify keypoint 2 as the free end. Type 2 in the picker, press the **Enter** key, and then click **OK**. The Apply F/M on KPs dialog appears.
6. In the drop down list for Direction of force/mom, select FY.
7. Enter 1 for the Force/moment value in the Apply F/M on KPs dialog, and click **OK**. The force symbol appears in the Graphics window at keypoint 2.
8. Save the model. Choose menu path **Utility Menu> File> Save As**. If the `buckle.db` file is not already selected, select it. Click **OK** and when prompted whether you want to overwrite the existing file, click **OK** again.

## 15.7.10. Solve the Eigenvalue Buckling Analysis

1. Set analysis options. Choose menu path **Main Menu> Solution> Unabridged Menu> Analysis Type> Analysis Options**. The Static or Steady-State Analysis dialog appears.
2. Use the sparse solver for the solution. In the Static or Steady-State Analysis dialog, make sure that Sparse solver is selected in the drop down box beside the Equation solver label.
3. Include prestress effect, which will be stored for later use in the eigenvalue buckling calculation. In the drop down list labeled Stress stiffness or prestress, select "Prestress ON." Click **OK** to close the Static or Steady-State Analysis dialog.
4. Choose menu path **Main Menu> Solution> Solve> Current LS**. Review the summary information in the **/STATUS** command window, then select Close from its menu bar. Click **OK** in the Solve Current Load Step window to begin the solution.
5. When the Solution is Done! window appears, click **Close** to close it.
6. Choose menu path **Main Menu> Finish**.
7. Choose menu path **Main Menu> Solution> Analysis Type> New Analysis**.
8. Select the "Eigen Buckling" option, then click **OK**.
9. Choose menu path **Main Menu> Solution> Analysis Type> Analysis Options**. The Eigenvalue Buckling Options dialog appears. Select the "Block Lanczos" option. Enter 4 in the "No. of modes to extract" field, then click **OK**.
10. Set the element calculation key for the **MPXAND** command. Choose menu path **Main Menu> Solution> Load Step Opts> ExpansionPass> Single Expand> Expand Modes**.
11. In the Expand Modes dialog, enter 4 in the "No. of modes to expand" field, change the "No" to "Yes" beside the "Calculate elem results" label, and click **OK**.
12. Choose menu path **Main Menu> Solution> Solve> Current LS**. Review the summary information in the **/STATUS** command window, then select Close from its menu bar. Click **OK** in the Solve Current Load Step window to begin the solution.
13. When the Solution is Done! window appears, click **Close** to close it.
14. Choose menu path **Utility Menu> PlotCtrls> Style> Size and Shape**. Be sure the radio button beside the label Display of element shapes ... (**/ESHAPE**) is set to On and click **OK**.
15. Display the results summary. Choose menu path **Main Menu> General Postproc> Results Summary**. After you have reviewed the results, click **Close** to close the window.
16. Choose menu path **Main Menu> General Postproc> Read Results> First Set**
17. Plot the first mode shape of the beam. Choose menu path **Main Menu> General Postproc> Plot Results> Deformed Shape**. The Plot Deformed Shape dialog appears. Select "Def + undef edge" and click **OK**.
18. Choose menu path **Main Menu> Finish**.

### 15.7.11. Solve the Nonlinear Buckling Analysis

1. Introduce model imperfections calculated by the previous analysis. Choose menu path **Main Menu> Preprocessor> Modeling> Update Geom**. In the Update nodes using results file displacements dialog, enter 0.002 in the "Scaling factor" field, 1 in the "Load step" field, 1 in the "Substep" field, and `file.rst` in the "Selection" field. Click **OK**.
2. Choose menu path **Main Menu> Solution> Analysis Type> New Analysis**.
3. Select the "Static" option, then click **OK**.
4. Choose menu path **Main Menu> Solution> Load Step Opts> Output Ctrl> DB/Results File** and be sure the drop down lists display All Items and All entities respectively. Choose the Every substep for the File write frequency radio button and click **OK**.
5. Choose menu path **Main Menu> Solution> Analysis Type> Analysis Options**. Select the radio button beside Large deform effects, then click **OK**.
6. Set the arc-length method, and set parameters for the termination of the solution. Choose menu path **Main Menu> Solution> Load Step Opts> Nonlinear> Arc-Length Opts**. Select the Arc-length method on/off radio button and set it to On. Choose the pull down menu next to the Lab label and select Displacement lim. Enter 1.0 in the "Max desired U" field. Enter 2 in the "Node number for VAL" field. From the pull-down menu (next to the "Degree of freedom" label), select "UZ." Click **OK**.
7. Define the number of substeps to be run during this load step. Choose menu path **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time and Substps**. Enter 10000 in the "Number of substeps" field and click **OK**.
8. Solve the current model. Choose menu path **Main Menu> Solution> Solve> Current LS**. Review the summary information in the **/STATUS** command window, then select Close from its menu bar. Click **OK** in the Solve Current Load Step window to begin the solution. A Nonlinear Solution window with a Stop button appears. A convergence graph is built, and can take several minutes to complete.
9. You may receive a warning message. You should review the information in the message but you do not need to close it. Continue waiting for the solution to complete. When the Solution is Done! window appears, click **Close** to close it.
10. Choose menu path **Main Menu> Finish**.

### 15.7.12. Plot and Review the Results

1. Replot the beam. Choose menu path **Utility Menu> Plot> Elements**.
2. Define the load point deflection to be read from the results file. Choose menu path **Main Menu> TimeHist PostPro> Define Variables**. When the Defined Time-History Variables dialog appears, select **Add**.
3. When the Add Time-History Variable dialog appears, be sure the Nodal DOF result option is selected. Click **OK**.
4. The Define Nodal Data picker appears. In the Graphics window, pick node 2 (the end node on the right side of the beam) and click **OK**.



5. The Define Nodal Data dialog appears. Be sure the Ref number of variable and Node number are both set to 2. Enter TIPLATDI in the "User-specified label" field. Select Translation UZ from the menu and click **OK**.
6. Define the total reaction force to be read from the results file. Select **Add** from the Defined Time-History Variables dialog.
7. When the Add Time-History Variable dialog appears, choose the Reaction forces radio button and then click **OK**.
8. The Define Reaction Force picker appears. Pick the end node on the left side of the beam and click **OK**.
9. The Define Reaction Force Variable dialog appears. Be sure the "Ref number of variable" is set to 3 and "Node number" is set to 1. Select "Struct Force FY" from the menu and click **OK**. Click **Close** to close the Defined Time-History Variables dialog.
10. Choose menu path **Main Menu> TimeHist Postpro> Math Operations> Multiply**. In the Multiply Time-History Variables dialog, enter 4 in the "Reference number for result" field, -1.0 in the "1st Factor" field, and 3 in the "1st Variable" field. Click **OK**.
11. Display the X variable. Choose menu path **Main Menu> TimeHist Postpro> Settings> Graph**. Choose the single variable button, enter 2 in the "Single variable no." field, and click **OK**.
12. Plot the load versus deflection curve to confirm the critical load calculated by the eigenvalue method. Choose menu path **Main Menu> TimeHist PostPro> Graph Variables**. Enter 4 in the "1st variable to graph" field. Click **OK**.
13. List the variables versus time. Choose menu path **Main Menu> TimeHist PostPro> List Variables**. Enter 2 in the "1st variable to list" field and 4 in the "2nd variable" field and click **OK**.
14. Check the values in the PRVAR Command window to see how they compare against the values generated by the eigenvalue buckling analysis. Expected results are: Critical buckling load,  $P_{cr} = 0.01892$ . Close the PRVAR Command window.

### 15.7.13. Plot and Review the Section Results

1. Replot the beam. Choose menu path **Utility Menu> Plot> Elements**.
2. Define the compression stress at the base to be read from the results file. Choose menu path **Main Menu> TimeHist PostPro> Define Variables**. When the Defined Time-History Variables dialog appears, select **Add**.
3. When the Add Time-History Variable dialog appears, be sure the "...by seq no." option is selected. Click **OK**.
4. The Define Element Results picker appears. In the Graphics window, pick element 1 (the end element on the left side of the beam) and click **OK**.
5. The Define Element Results by Seq No. dialog appears. Be sure the Ref number of variable is set to 5 and Element number is set to 1. Enter BASESX in the "User-specified label" field. Select "LS" from the menu.
6. To retrieve the X component of Stress from the bottom middle of the section (this is node 3, or the 2nd node where stress is stored), type 4 for the Comp Sequence number and click **OK**.

7. Plot variable versus time. Choose menu path **Main Menu> TimeHist PostPro> Graph Variables**. Enter 5 in the "1st variable to graph" field. Click **OK**.
8. List the variable versus time. Choose menu path **Main Menu> TimeHist PostPro> List Variables**. Enter 5 in the "1st variable to list" field and click **OK**.
9. Choose menu path **Main Menu> Finish**.
10. In the Toolbar, click **Quit**.
11. Choose a save option and click **OK**.

## 15.8. Example Problem with Cantilever Beams

Here is the input file for the problem described in the previous section:

```

/filename,tutor-sag13s

/graphics,power
/gst,on
/show,buckle,grph
/prep7
k,1,0,0,0
k,2,100.0,0,0
k,3,50,5,0
lstr,1,2
et,1,beam189
sectype,1, beam, rect
secdata, 0.2,5.0
slist, 1, 1
mp,ex,1,1e4
mp,nuxy,1,0.0
lsel,s, , , 1, 1, 1
latt,1, ,1,0, 3, ,1
lesize, all, , ,10
secnum,1
lmesh,all
/view,,1,1,1
/eshape,1
eplot
dk,1, , , ,0,all
fk,2,fy,1.0
finish
/solu
pstres,on
eqslv,sparse ! eqslv,sparse is the default for static and full transient
solve
finish
/solu
antype,buckle
bucopt,lanb,4
mexpand,4,, ,yes
solve
finish
/post1
/eshape,1
/view, 1 ,1,1,1
/ang, 1
set,list
set,1,1
/show, jpeg
pldisp,2
/show,close
finish

/prep7
upgeom,0.002,1,1,tutor-sag13s,rst

```

```
/solu
antype,static
outres,all,all
nlgeom,on
arclen,on,25,0.0001
arctrm,u,1.0,2,uz
nsubst,10000
solve
finish
/show,jpeg
/post26
nsol,2,2,u,z,tiplatdi
rforce,3,1,f,y
prod,4,3,,,,,-1.0,1,1
xvar,2
plvar,4
prvar,2,4
esol,5,1,,ls,4
plvar,5
/show,close
prvar,5
finish
```

## 15.9. Where to Find Other Examples

Several ANSYS publications, particularly the *Mechanical APDL Verification Manual*, describe additional beam analyses.

The *Mechanical APDL Verification Manual* consists of test case analyses demonstrating the analysis capabilities of the ANSYS program. While these test cases demonstrate solutions to realistic analysis problems, the *Mechanical APDL Verification Manual* does not present them as step-by-step examples with lengthy data input instructions and printouts. However, most ANSYS users who have at least limited finite element experience should be able to fill in the missing details by reviewing each test case's finite element model and input data with accompanying comments.

The *Mechanical APDL Verification Manual* contains one beam test case:

[VM222 - Warped Cantilever Beam](#)



---

## Chapter 16: Shell Analysis and Cross Sections

---

Shell elements are used to create a mathematical 2-D idealization of a 3-D structure. They offer computationally efficient solutions for modelling shell structures when compared to solid elements.

The discussion in this chapter applies to 3-D finite-strain shell elements such as [SHELL181](#) and [SHELL281](#). Compared to other shells, these shell elements provide more robust nonlinear analysis capabilities, and significant improvements in cross-section data definition, analysis, and visualization.

The method for defining shell sections described here can also be used to define the cross-sectional properties of the layered thermal shell elements [SHELL131](#) and [SHELL132](#). However, information presented here concerning integration points (*NUMPT* on the [SECDATA](#) command) and section properties ([SECCONTROL](#) command) does not apply to [SHELL131](#) and [SHELL132](#).

The following topics are available for shell analysis and cross sections:

- [16.1. Understanding Cross Sections](#)
- [16.2. How to Create Cross Sections](#)
- [16.3. Using Preintegrated General Shell Sections](#)

### 16.1. Understanding Cross Sections

A cross section defines the geometry of the shell in a plane parallel to the shell x-y plane. Through the section family of commands, you can describe the z direction of the element by defining consecutive layers. Each layer may vary in thickness, material type, orientation (from element x-axis), and number of integration points.

Alternatively, you can define homogeneous shell section behavior directly via preintegrated shell sections, a method commonly used in analyses involving laminated composite structures. With preintegrated shell sections, you can directly specify the membrane, bending, and coupling properties. The preintegrated method also allows analysis of complex geometry (with repeated patterns such as corrugated sheets) using equivalent shell section properties. For more information, see [Using Preintegrated General Shell Sections](#) (p. 465).

### 16.2. How to Create Cross Sections

The general procedure for creating cross sections consists of the following steps:

1. Define the section and associate a section ID number with the section.
2. Define the geometry data for the section.

The following commands are available for creating, viewing, and listing cross sections, and for managing cross-section libraries:

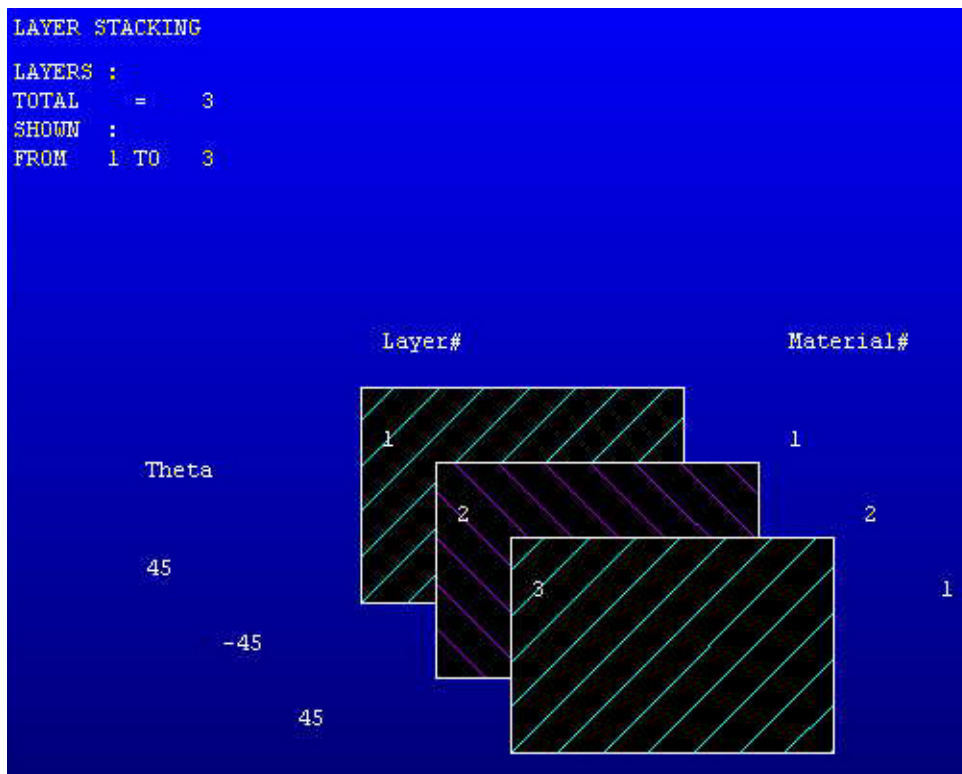
**Table 16.1: Cross-Section Commands**

Command	Purpose
<b>SECTYPE</b>	Associates section with <i>SECID</i> (section number)
<b>SECDATA</b>	Defines section geometry data
<b>SECCON-TROL</b>	Overrides program calculated properties.
<b>SECFUNC-TION</b>	Specifies shell section thickness as a tabular function.
<b>SECNUM</b>	Identifies the <i>SECID</i> (section number) to be assigned to an element
<b>SECOFFSET</b>	Defines section offset for shell cross sections
<b>SECPLLOT</b>	Plots geometry of a shell section to scale
<b>SLIST</b>	Summarizes section properties
<b>SDELETE</b>	Deletes a cross section

For complete documentation of the cross-section commands, see the [Command Reference](#).

Figure 16.1: Plot of a Shell Section (p. 462) shows the layer stacking of a shell section. The layer order, along with material and orientation of each layer, is represented here:

**Figure 16.1: Plot of a Shell Section**



### 16.2.1. Defining a Section and Associating a Section ID Number

Use the **SECTYPE** command to define a section and associate it with a section ID number. For example, the following command assigns a section identification number (2) to a shell section:

**Command(s):** **SECTYPE**, 2, SHELL

**GUI:** Main Menu> Preprocessor> Sections> Shell> Add/Edit

### 16.2.2. Defining Layer Data

Use the **SECDATA** command to define the layers of a shell section. Each consecutive **SECDATA** command defines the next layer's thickness, material, orientation, and number of integration points. (The number of integration points input on **SECDATA** is not used by thermal shell elements.) The layer orientation angle is the angle between the layer coordinate system and the x-axis of the element coordinate system.

You may designate the number of integration points (1, 3, 5, 7, or 9) located thru the thickness of each layer. When only 1, the point is always located midway between the top and bottom surfaces. If 3 or more points, 2 points are located on the top and bottom surfaces respectively and the remaining points are distributed equal distance between the 2 points. An exception occurs when designating 5 points, where the quarter point locations are moved 5 percent toward their nearest layer surface to agree with the locations selected with real constant input. The default for each layer is 3.

If a shell section has only one layer, and the number of section integration points is equal to one, then the shell does not have any bending stiffness. This may result in solver difficulties, and may affect convergence adversely.

**Command(s):** **SECDATA**, 0.5, 1, 45, 3 **SECDATA**, 0.5, 2, -45, 3 **SECDATA**, 0.5, 1, 45, 3

**GUI:** Main Menu> Preprocessor> Sections> Shell> Add/Edit

### 16.2.3. Overriding Program Calculated Section Properties

Use the **SECCONTROL** command to override the program calculated section properties. By default, the program calculates shear correction factors and mass for each element based on the input section geometry and material properties. Any values input on the **SECCONTROL** command will replace the defaults. (**SECCONTROL** does not apply to thermal shell elements.)

**Command(s):** **SECCONTROL**, 0.8, 0.0, 0.8, 1.0

**GUI:** Main Menu> Preprocessor> Sections> Shell> Add/Edit

### 16.2.4. Specifying a Shell Thickness Variation (Tapered Shells)

Use the **SECFUNCTION** command to associate a tabular thickness variation with the section. A table that describes thickness with respect to the global Cartesian coordinate system may be associated with a shell section. This thickness is interpreted as the total thickness of a shell. The total thickness of a layered shell, and all layer thickness values, will be scaled according to the tabular function.

**Command(s):** **SECFUNCTION**, %table\_name%

**GUI:** Main Menu> Preprocessor> Sections> Shell> Add/Edit

### 16.2.5. Setting the Section Attribute Pointer

Use the **SECNUM** command to associate an element with a particular section. Any element created after the **SECNUM** command will have this section identification number (2) as its section attribute.

**Command(s):** **SECNUM**, 2

**GUI:** Main Menu> Preprocessor> Meshing> Mesh Attributes> Define> Default Attrs

## 16.2.6. Associating an Area with a Section

Use the **AATT** command to associate an area with a shell section type. When the area is meshed, the new elements are associated with the section specified on the **AATT** command. The section specified by the **SECNUM** command is ignored.

**Command(s):** **AATT**, , , , , 2

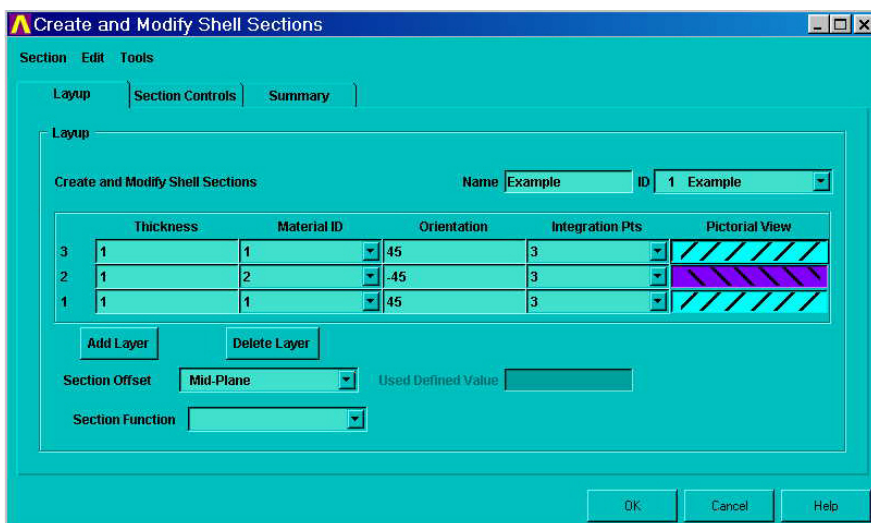
**GUI:** Main Menu > Preprocessor > Meshing > Mesh Attributes > All Areas

Main Menu > Preprocessor > Meshing > Mesh Attributes > Picked Areas

## 16.2.7. Using the Shell Tool to Create Sections

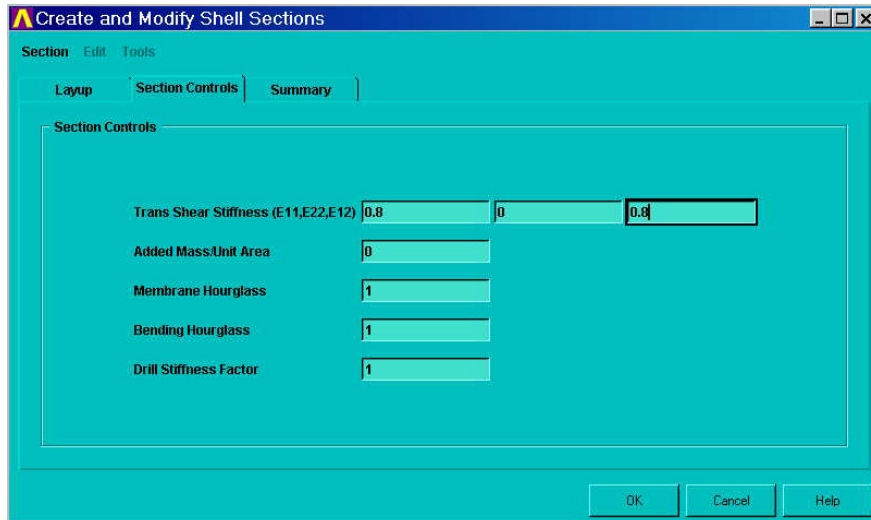
The **SECTYPE**, **SECDATA**, **SECOFFSET**, **SECFUNCTION**, and **SECCONTROL** commands (**Main Menu > Preprocessor > Sections > Shell > Add/Edit**) are all associated with the ShellTool in the GUI.

**Figure 16.2: Shell Tool With Layup Page Displayed**



The top part of the Layup page of the ShellTool relates a section ID number to a Shell section type (and, optionally, a section name) [**SECTYPE**]. The middle part of the page of the ShellTool defines each layer in the positive Z direction of the element [**SECDATA**]. Note that the order of the rows in the spreadsheet ascends up the page, resembling that stacking of the layers. (Integration point data for each layer is ignored by thermal shell elements.) The bottom contains the fields for section offset information, if needed [**SECOFFSET**]. You can define tapered shells by specifying a section function relating thickness to global coordinates [**SECFUNCTION**].



**Figure 16.3: Shell Tool With Section Controls Page Displayed**

The transverse shear stiffness, hourglass coefficients and drill stiffness of the sections are calculated from the section geometry and material properties. The default added mass per unit area of the section is always zero. On the Section Controls page, you can override the program calculated quantities [SECCONTROL]. (The Section Controls page does not apply to thermal shell elements.)

**Figure 16.4: Shell Tool With Summary Page Displayed**

On the Summary page, you can review the section properties.

## 16.2.8. Managing Cross-Section Libraries

Cross-section data for shell sections can be stored in cross-section libraries. To create standard cross sections for later use, create one or more cross sections, edit the `Jobname.LOG` file, and copy the appropriate **SECCONTROL**, **SECDATA**, **SECFUNCTION**, **SECOFFSET**, and **SECTYPE** commands into a separate file with a `SECT` extension.

## 16.3. Using Preintegrated General Shell Sections

You can use preintegrated general shell sections (**SECTYPE**,GENS) when using the `SHELL181` or `SHELL281` element, provided that linear elastic material behavior is acceptable. Compared to standard shell usage

with independent material and section definitions, preintegration requires fewer system resources because numerical integration through the thickness of the shell is not required.

The behavior of shell elements is governed by the generalized-stress/generalized-strain relationship of the form:

$$\begin{Bmatrix} \mathbf{N} \\ \mathbf{M} \end{Bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\varepsilon} \\ \boldsymbol{\kappa} \end{Bmatrix} - (T - T^I) \begin{Bmatrix} \mathbf{M}^T \\ \mathbf{B}^T \end{Bmatrix}$$

$$[\mathbf{S}] = [\mathbf{E}]\{\boldsymbol{\gamma}\}$$

where:

$\{\mathbf{N}\} = \{N_{11}, N_{22}, N_{12}\}$  are membrane stress resultants per unit length

$\{\mathbf{M}\} = \{M_{11}, M_{22}, M_{12}\}$  are bending stress resultants per unit length

$\{\boldsymbol{\varepsilon}\} = \{\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12}\}$  are membrane strains

$\{\boldsymbol{\kappa}\} = \{\kappa_{11}, \kappa_{22}, \kappa_{12}\}$  are curvatures

$\{\mathbf{S}\} = \{S_1, S_2\}$  are transverse shear forces per unit length

$\{\boldsymbol{\gamma}\} = \{\gamma_1, \gamma_2\}$  are transverse shear strains

$[\mathbf{A}]$ ,  $[\mathbf{B}]$ ,  $[\mathbf{D}]$ , and  $[\mathbf{E}]$  are the section-stiffness matrices for membrane, coupling, bending, and shear behavior, respectively

$\{\mathbf{M}^T\}$  are stress resultants caused by a unit rise in temperature on a fully constrained model

$\{\mathbf{B}^T\}$  are bending-related stress resultants caused by a unit rise in temperature on a fully constrained model

$T^I$  is the stress-free initial temperature

$T$  is the current temperature

(Subscripts 1 and 2 denote shell surface coordinates as employed by the shell element.)

The preintegrated form of input allows you to import homogeneous section-stiffness constants evaluated in other analyses or by third-party, special-purpose software tools. For example, defining preintegrated section stiffnesses  $[\mathbf{A}]$ ,  $[\mathbf{B}]$ , and  $[\mathbf{D}]$  is common in analyses of layered composites, corrugated shells, or other complex section construction.

### 16.3.1. Defining a Preintegrated Shell Section

Each of the following commands specifies a particular component quantity necessary for defining a preintegrated shell section:

**Table 16.2: Commands for Specifying Preintegrated Shell Section Data**

Command	Quantity Defined and Data Specified
<b>SSPA</b>	Membrane stiffness -- Symmetric part of submatrix $[\mathbf{A}]$ $A_{11}, A_{21}, A_{31}, A_{22}, A_{32}, A_{33}, T$
<b>SSPB [1]</b>	Coupling stiffness -- Symmetric part of submatrix $[\mathbf{B}]$ $B_{11}, B_{21}, B_{31}, B_{22}, B_{32}, B_{33}, T$
<b>SSPD [1]</b>	Bending stiffness -- Symmetric part of submatrix $[\mathbf{D}]$ $D_{11}, D_{21}, D_{31}, D_{22}, D_{32}, D_{33}, T$

Command	Quantity Defined and Data Specified
<b>SSPE</b> [1]	Transverse shear stiffness -- Symmetric part of submatrix $[E]$ $E_{11}, E_{21}, E_{22}, T$
<b>SSMT</b>	Membrane thermal effects -- $\{M^T\}$ $M^T_{11}, M^T_{22}, M^T_{12}, T$
<b>SSBT</b> [1]	Bending thermal effects -- $\{B^T\}$ $B^T_{11}, B^T_{22}, B^T_{12}, T$
<b>SSPM</b>	Mass density of the shell section (assuming a unit thickness) $DENS, T$

1. If you are using the **SHELL181** or **SHELL281** element's Membrane option (**KEYOPT**(1) = 1), it is not necessary to issue the **SSPB**, **SSPD**, **SSPE**, and **SSBT** commands. In this case, the only relevant quantities are membrane stiffness (**SSPA**), membrane thermal effects (**SSMT**), and density (**SSPM**).

**Temperature dependencies (T)** You can define each of the preintegrated shell section data as temperature-dependent. It is possible to specify up to six temperatures ( $T$ ) by reissuing any command as necessary. If you issue a command for a temperature specified earlier, the most recent data supersedes the previous value.

### 16.3.2. Considerations for Using Preintegrated Shell Sections

You can define preintegrated shell sections only when using the **SHELL181** or **SHELL281** element. When using **preintegrated shell section** data, the following conditions apply:

- The section data defined by each command listed in [Table 16.2: Commands for Specifying Preintegrated Shell Section Data \(p. 466\)](#) is associated with the section most recently defined via the **SECTYPE** command.
- You cannot use the shell element with heat-transfer shells.
- Shell stresses are not available for output; however, the stress resultants are available as **ETABLE** quantities.
- The thickness of the shell is assumed to remain constant even in a large-strain analysis.
- Section offsetting (**SECOFFSET**) is not available.
- The preintegrated stiffness components must yield a positive definite section stiffness at analysis time.

If using preintegrated shell sections in a **contact analysis**, ANSYS, Inc. recommends the pure Lagrange multiplier method.

For standard layered-shell sections, you can obtain submatrices  $[A]$ ,  $[B]$ ,  $[D]$ , and  $[E]$  via the **SLIST** command's **FULL** option.



---

## Chapter 17: Reinforcing

---

Reinforced materials are used extensively in civil construction, aircraft structures, automobiles, advanced sports equipment, and medical devices. Reinforcing commonly appears in fiber or cable forms, such as steel rebar in reinforced concrete, nylon strands in tires, and carbon fibers in various composite materials.

ANSYS allows you to model the reinforcing fibers with specialized reinforcing elements. The reinforcing elements interact with standard structural elements, referred to as the *base elements*, via the common nodes.

Reinforcing sections (**SECTYPE**,REINF) define the location and orientation of the reinforcing (**SECDATA**). The sections are referenced by REINF264 and REINF265 elements.

The following topics related to reinforcing are available:

- 17.1. Assumptions About Reinforcing
- 17.2. Modeling Options for Reinforcing
- 17.3. Defining Reinforcing Sections and Elements
- 17.4. Reinforcing Simulation and Postprocessing

### 17.1. Assumptions About Reinforcing

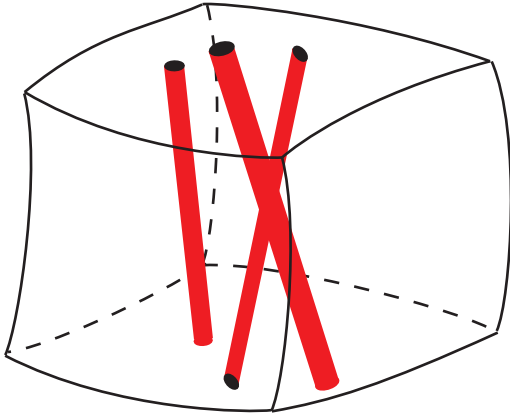
The cross-section area of reinforcing fibers is small compared to the length of the fibers. The bending, torsion, and transverse shear stiffness (all present in beam elements) are ignored in reinforcing elements. ANSYS considers only the axial stiffness.

ANSYS assumes a secure bond between the reinforcing fibers and the base element. The relative movement between these two components is not permitted; therefore, the motion of reinforcing fibers is determined solely by the motion of the base element. Based on this simplification, ANSYS adopts the same nodes and connectivity for a reinforcing element and its base element.

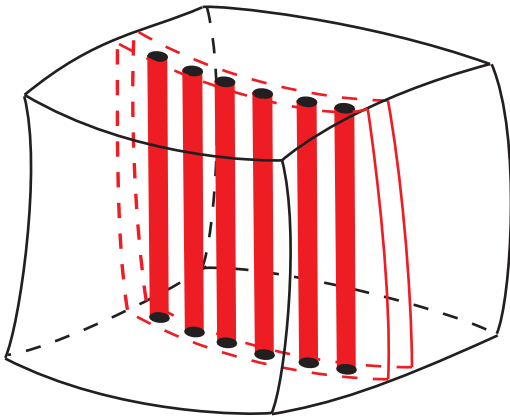
### 17.2. Modeling Options for Reinforcing

ANSYS provides *discrete* and *smear*d reinforcing modeling options.

Use the discrete modeling option (**SECTYPE**,SECID,REINF,DISC) for reinforcing fibers with nonuniform materials, cross-section areas, or arbitrary orientations. ANSYS models each fiber separately as a spar having only uniaxial stiffness, as shown.

**Figure 17.1: Discrete Reinforcing Modeling Option**

Use the smeared option (**SECTYPE**,**SECID**,**REINF**,**SMEAR**) for modeling reinforcing fibers that appear in layered form. ANSYS treats one layer of fibers with identical material, orientation, and cross-section area as a homogeneous membrane having unidirectional stiffness, as shown:

**Figure 17.2: Smeared Reinforcing Modeling Option**

For detailed information about the reinforcing options, see the documentation for the [REINF264](#) and [REINF265](#) elements, and [REINF264 - 3-D Discrete Reinforcing](#) and [REINF265 - 3-D Smeared Reinforcing](#) in the *Mechanical APDL Theory Reference*.

### 17.3. Defining Reinforcing Sections and Elements

Use the following general process to create reinforcing elements:

1. Create the base elements.
2. Build the reinforcing sections.
3. Select the base elements that you want to reinforce.
4. Generate the reinforcing elements from the base elements (**EREINF**).
5. Inspect and verify the newly created reinforcing elements.

Reinforcing elements ignore any subsequent modifications made to the base elements. ANSYS, Inc. recommends issuing the **EREINF** command only *after* you have finalized the base elements. If you delete

or modify base elements after generating the reinforcing elements, remove all affected reinforcing elements and *reissue* the **EREINF** command to avoid inconsistencies.

The **EREINF** command determines the reinforcing element type according to the base element type and the current reinforcing section. (See the example input files in [Example: Discrete Reinforcing \(p. 471\)](#) and [Example: Smeared Reinforcing \(p. 472\)](#).) The command does not create any new reinforcing elements if the current section is not a valid reinforcing section or if the section is incompatible with the base elements. To define a section ID to associate with the subsequently-defined reinforcing elements, issue the **SECNUM** command.

Reinforcing locations are given with respect to the base element. Unexpected reinforcing placement may result from incorrectly oriented base elements. You can achieve the desired base element orientation via the **ORIENT** or **VEORIENT** command. In some cases, you may need to create different reinforcing sections for different base elements with inconsistent orientations.

As shown in the example input in [Example: Discrete Reinforcing \(p. 471\)](#) and [Example: Smeared Reinforcing \(p. 472\)](#), you can manually adjust the translucency of the base elements to show the reinforcing elements. Additional automatic reinforcing display options are available via the ANSYS GUI (**Main Menu**> **Preprocessor**> **Sections**> **Reinforcing**> **Display Options**).

### 17.3.1. Example: Discrete Reinforcing

Following is typical sequence of commands for defining discrete reinforcing elements:

```
/batch,list
/show, sample,grph
/title, Sample input data for generating discrete reinforcing elements
/prep7
!
! Define material properties
!
mp, ex,1,10e6 ! Base Material
mp, ex,2,30e6 ! Reinforcing Material
mp,prxy,1,.3
mp,prxy,2,.3
mp,dens,1,.00001
mp,dens,2,.00002
!
! Define base geometry
!
k,1,0.,0.
k,2,10.,0.
l,1,2
!
! Define base elements
!
et,1,BEAM188
mat,1
sect,1,BEAM,I
secd,1.,1.,1.,0.2,0.2,0.3
esize, 2.0
lmesh, 1
!
! Define discrete reinforcing section
!
sect,2,REINF,DISC
secd,2,0.01,BEAM,-0.25,0.1
secd,2,0.01,BEAM, 0 .0,0.1
secd,2,0.01,BEAM, 0.25,0.1
secd,2,0.01,BEAM,-0.25,0.9
secd,2,0.01,BEAM, 0 .0,0.9
secd,2,0.01,BEAM, 0.25,0.9
!
! Create reinforcing element via EREINF
```

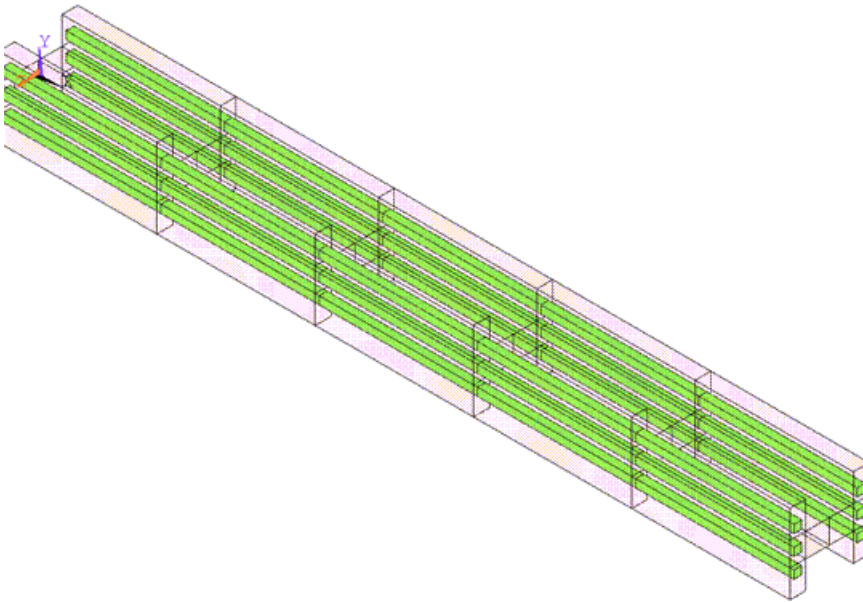
```

!
esel,all
secn, 2 ! use reinforcing section!
EREINF
!
! Inspect newly created reinforcing elements
!
esel,s,type,,1
! adjust the translucency level of the
! base element to reveal the embedded reinforcing
! elements
/trlcy,elem,0.9
esel,all
/view,1,1,1,1
! Turn on the expanded element shapes
/eshape,1
eplot
fini

```

The input listing generates the following output:

**Figure 17.3: Discrete Reinforcing Element Display (with Translucent Base Elements)**



### 17.3.2. Example: Smeared Reinforcing

Following is a typical sequence of commands for defining smeared reinforcing elements:

```

/batch,list
/show, sample,grph
/title, Sample input data for generating smeared reinforcing elements
/prep7
!
! Define material properties
!
mp, ex,1,10e6 ! Base Material
mp, ex,2,30e6 ! Reinforcing Material
mp,prxy,1,.3
mp,prxy,2,.3
mp,dens,1,.00001
mp,dens,2,.00002
!
! Define base geometry
!
CYLIND, 10.,12.,0.,20.,0,90.
!

```



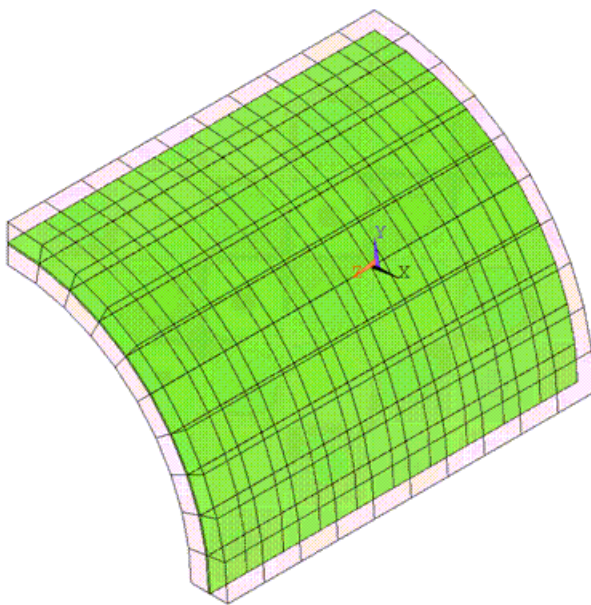
```

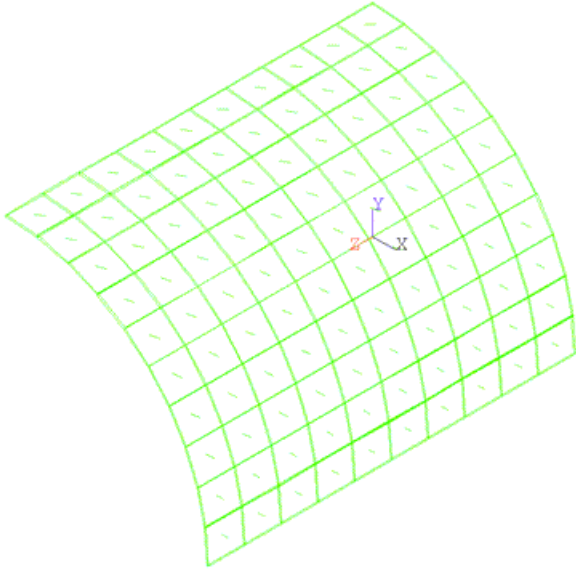
! Define base elements
!
et,1,SOLID185
mat,1
esize, 2.0
vmesh, 1
!
! Define smeared reinforcing section
!
sect,1,REINF,SMEAR
secd,2,0.1,1.0,,45.0,ELEF,3,0.5
!
! Create reinforcing element via EREINF
!
esel,all
EREINF
!
! Inspect newly created reinforcing elements
!
esel,s,type,,1
! adjust the translucency level of the
! base element to reveal the embedded reinforcing
! elements
/trlcy,elem,0.9
esel,all
/view,1,1,1,1
! Turn on the expaned element shapes
/eshape,1
eplot
!
! Verify the reinforcing fiber orientation
!
! Switch to the vector plot
/dev,vect,1
! Select only reinforcing elements (optional)
esel,s,type,,2
/psymb,layr,-1
eplot
fini

```

The input listing generates the following output:

**Figure 17.4: Smeared Reinforcing Element Display (with Translucent Base Elements)**



**Figure 17.5: Fiber Orientation Display on Smeared Reinforcing Elements**

## 17.4. Reinforcing Simulation and Postprocessing

The **REINF264** and **REINF265** reinforcing elements have plasticity, stress stiffening, creep, large deflection, initial stress, and large strain capabilities. You can use them in any type of simulation that their base elements support.

Reinforcing elements allow tension-only (cable) and compression-only (gap) options. You can specify those options via the **SECCONTROL** command. A nonlinear iterative solution procedure is necessary for simulating the tension-only or compression-only behaviors.

Reinforcing elements always list or display stress and strain results in local fiber directions. Therefore, the coordinate system specified via the **RSYS** command has no effect. Reinforcing results display is supported in both PowerGraphics and FullGraphics. In PowerGraphics mode, all reinforcing fibers (discrete) or layers (smeared) are displayed shown concurrently. In FullGraphics mode, however, only one reinforcing fiber or layer can be displayed at one time. To specify the desired layer or fiber to display in FullGraphics mode, issue the **LAYER** command.

You can perform a failure analysis on the reinforcing elements. For more information, see [Specifying Failure Criteria for Composites](#) (p. 413).

---

## Chapter 18: Modeling Hydrostatic Fluids

---

Hydrostatic fluid elements are used to model fluids that are fully enclosed by solids (containers). The available elements are:

- [HSFLD241](#) for 2-D models (plane stress, axisymmetric, or plane strain)
- [HSFLD242](#) for 3-D models

These elements are formulated for linear and nonlinear static and transient dynamic applications. For more information, see [HSFLD241 - 2-D Hydrostatic Fluid](#) and [HSFLD242 - 3-D Hydrostatic Fluid](#) in the *Mechanical APDL Theory Reference*.

The following topics related to using hydrostatic fluid elements are available:

- 18.1. [Hydrostatic Fluid Element Features](#)
- 18.2. [Defining Hydrostatic Fluid Elements](#)
- 18.3. [Material Definitions and Loading](#)
- 18.4. [Example Model Using Hydrostatic Fluid Elements](#)
- 18.5. [Results Output](#)

For an example analysis, see [Hydrostatic Fluid Analysis of an Inflating and Rolling Tire](#) in the *Technology Demonstration Guide*.

### 18.1. Hydrostatic Fluid Element Features

The following are characteristics of the hydrostatic fluid elements, [HSFLD241](#) and [HSFLD242](#):

- The fluid volume is fully enclosed within the solid so it has no free surface.
- All hydrostatic fluid elements defining a fluid volume share a pressure node with a hydrostatic pressure degree of freedom, so the fluid volume has uniform pressure, temperature and density.
- The pressure node can be located anywhere within the fluid volume. It is automatically moved to the geometric center of the fluid volume if there are no displacement degree-of-freedom constraints specified. However, if the fluid volume is bounded by one or more symmetry lines or planes, the pressure node must be on the symmetry line or plane or the intersecting corner or line of multiple symmetry lines or planes, and it must have symmetry boundary conditions.
- For 2-D models, the planar behavior of [HSFLD241](#) elements (plane stress or plain strain) is based on the underlying solid element. Use `KEYOPT(3)` to define axisymmetric behavior.
- The fluid may be modeled as incompressible (use `KEYOPT(6) = 1`), or it may be modeled as compressible by defining fluid material models (**TB,FLUID**). The effects of fluid viscosity are ignored.
- Fluid flow between two fluid volumes with separate pressure nodes may be modeled by connecting the pressure nodes with [FLUID116](#) coupled thermal-fluid pipe elements. In this case, you must set `KEYOPT(1) = 1` for the hydrostatic fluid elements to activate the HDSP and PRES (pressure) degrees of freedom at the pressure node, and you must set `KEYOPT(1) = 3` on the [FLUID116](#) elements. The

PRES (pressure) and HDSP (hydrostatic pressure) degrees of freedom are made to be the same at the pressure node.

- Inertial effects such as sloshing cannot be modeled due to a uniform pressure assumption. However, fluid mass can be added to the hydrostatic fluid element surface nodes that are shared with underlying solid or shell elements. Use KEYOPT(5) to distribute fluid mass to surface nodes.

## 18.2. Defining Hydrostatic Fluid Elements

Hydrostatic fluid elements must be associated with solid or shell elements. Use [HSFLD241](#) with 2-D solid or axisymmetric solid elements; use [HSFLD242](#) with 3-D solid or shell elements.

Follow this general procedure to automatically generate the hydrostatic fluid elements on the faces of selected solid or shell elements:

1. Build the model using solid or shell elements.
2. Select the solid or shell elements and the attached nodes that enclose the fluid volume.
3. Define a node to be used as the pressure node for the fluid volume.
4. Use the **ESURF** command with the pressure node ID specified as *XNODE* to generate hydrostatic fluid elements.
5. Inspect and verify the newly created hydrostatic fluid elements; the elements should be triangle-shaped (2-D or axisymmetric) or pyramid-shaped (3-D) with a common vertex at the pressure node.

As an alternative to the above method, you can use the **E** or **EN** command to manually define hydrostatic fluid elements that share nodes of underlying solid or shell elements. For the 2-D or axisymmetric case ([HSFLD241](#)), nodes I through K (I, J for lower order; I, J, K for higher order) should be shared with the underlying 2-D or axisymmetric solid element, and node L should be defined as the pressure node. For the 3-D case ([HSFLD242](#)), nodes I through P (I through L for lower order; I through P for higher order) should be shared with the underlying 3-D solid or shell element, and node Q should be defined as the pressure node.

In general, the nodes should be ordered in a counter-clockwise manner to get a positive volume. If the node ordering is not correct, it can be reversed by either using the **ESURF** command or changing the order on the **E** or **EN** commands.

In some situations, the element volume may need to be negative with nodes ordered in the clockwise direction. For example, when modeling an inflated tire inner tube, the pressure node will be located outside of the fluid volume. In this case, some of the hydrostatic fluid elements filling the doughnut-shaped fluid volume will need to have negative volume (see (1) in the figure below), while others will have positive volume (see (2) in the figure below) such that the total volume of all the hydrostatic fluid elements adds up to the volume of the enclosed fluid. For this reason, hydrostatic fluid elements with negative volume are allowed. However, to avoid errors, a warning message is issued at the beginning of the analysis for all hydrostatic fluid elements having negative volumes. An internal element component containing all hydrostatic fluid elements with negative volumes is created and listed in this warning message.

**Figure 18.1: Negative and Positive Volumes for Hydrostatic Fluid Elements****(1) Hydrostatic fluid elements having negative volume****(2) Hydrostatic fluid elements having positive volume**

Hydrostatic fluid elements do not require underlying solid or shell elements in following cases:

- The fluid volume is enclosed by a rigid solid. Instead of adding very stiff underlying solid or shell elements, displacement degree-of-freedom constraints may be imposed directly on the hydrostatic fluid element nodes. Another alternative is to use a rigid body instead of stiff solid or shell elements (see [Modeling Rigid Bodies](#) in the *Contact Technology Guide*).
- Some of the underlying solid or shell elements undergo large displacements while others do not. For example, when a piston moves inside a fluid-filled cylinder, the fluid volume in contact with the piston and the cylinder may be modeled by a hydrostatic fluid element with one node shared with the solid or shell element on the piston, and the other node shared with the solid or shell element on the cylinder. Such an arrangement allows the piston to displace without adversely affecting the total fluid volume or pressure calculations. (For an example of this type of model, see [Example Model Using Hydrostatic Fluid Elements](#) (p. 479).)

## 18.3. Material Definitions and Loading

### 18.3.1. Fluid Materials

Use the **TB,FLUID** command to define a material model for compressible hydrostatic fluid elements. All elements sharing a pressure node must share the same material. There are three ways to define material data for compressible fluids: liquid, gas, or pressure-volume data. See [Fluids](#) in the *Material Reference* for a complete description of these material models.

Use the **TB,FLUID** command with *TBOPT* = LIQUID to define material behavior for a liquid, and specify the following material constants using the **TB,DATA** command:

Constant	Symbol	Meaning
C1	K	Bulk modulus
C2	$\alpha$	Coefficient of thermal expansion
C3	$\rho_{of}$	Initial density

Use the **TB,FLUID** command with  $TBOPT = GAS$  to define material behavior for a gas, and specify the following material constant using the **TB,DATA** command:

Constant	Symbol	Meaning
C1	$\rho_{0f}$	Initial density

Since the compressible gas is modeled using the Ideal Gas Law, you must specify a reference pressure (use real constant PREF on the hydrostatic fluid element) and you must specify a reference temperature (use the **TREF** or **MP,REFT** command) with temperature offset (use the **TOFFST** command).

Use the **TB,FLUID** command with  $TBOPT = PVDATA$  to define compressible fluid behavior in terms of a pressure-volume curve. Use the **TBPT** command to enter the data points ( $X, Y$ ) for the curve:

Constant	Meaning
$X$	Pressure value
$Y$	Corresponding volume value

Use the **MP,DENS** command to define initial density for a fluid with pressure-volume data.

Use the **TBTEMP** command to define temperature dependent material constants for all of the above material options.

Use the **MP,DENS** and **MP,ALPX** commands to specify density and coefficient of thermal expansion, respectively, for an incompressible fluid that allows thermal expansion.

### 18.3.2. Loads and Boundary Conditions

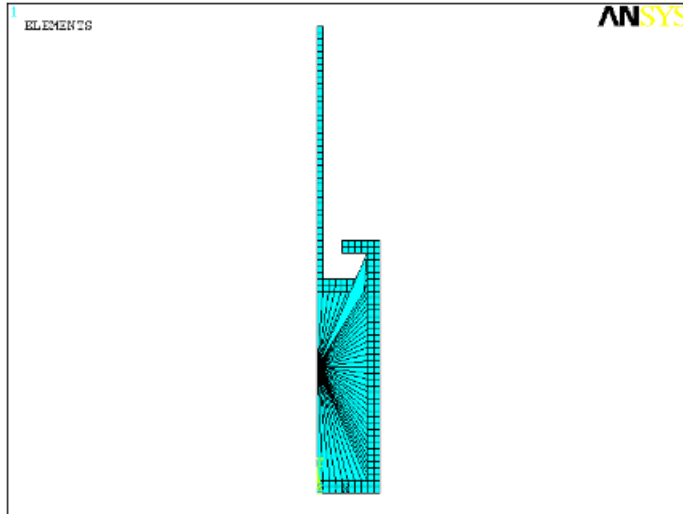
The following loads and boundary conditions can be applied to hydrostatic fluid elements:

Load	Command	Comments
Initial fluid pressure	<b>IC</b>	Apply initial fluid pressure at pressure node with HDSP degree of freedom.
Initial fluid temperature	<b>TREF</b> or <b>MP,REFT</b>	<b>TREF</b> applies to entire model. <b>MP,REFT</b> applies to a specific material.
Temperature offset (from absolute zero)	<b>TOFFST</b>	
Temperature loads	<b>BF</b>	By default, the temperature is set to <b>TUNIF</b> .
Hydrostatic pressure DOF constraint	<b>D</b>	Apply this constraint type to the pressure node. This constraint is equivalent to applying a surface load on the underlying solid surface, as far as the effect of fluid pressure on the underlying solid is concerned. The change in hydrostatic pressure value is assumed to occur as a result of the addition or removal of fluid mass to or from the containing vessel.
Fluid mass flow rate	<b>F</b>	Apply fluid mass flow rate at the pressure node. A positive value adds fluid mass to the containing vessel, and a negative value results in removal of fluid mass.

## 18.4. Example Model Using Hydrostatic Fluid Elements

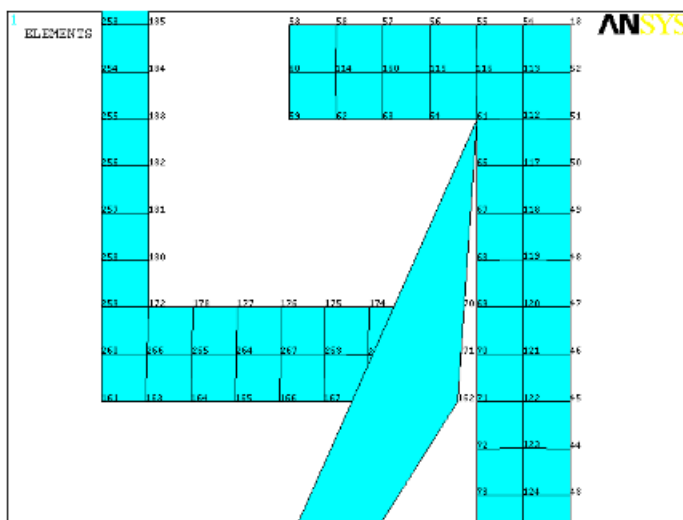
The following example shows an axisymmetric model of a cylinder-piston arrangement. The contained fluid in the cylinder is modeled with hydrostatic fluid elements, shown as triangular elements in the figure below.

**Figure 18.2: Meshed Model**



Due to the gap between the cylinder and piston, a hydrostatic fluid element with no underlying solid elements is added manually. The node numbering on this hydrostatic fluid element is reversed to achieve a negative fluid volume that balances the extra fluid volume created by hydrostatic fluid elements attached to the cylinder above the piston. The figure below shows a close-up view of the node numbering in the area where the hydrostatic fluid element with negative volume and no underlying solid element is defined.

**Figure 18.3: Close-up View of Meshed Model**



The following sequence of commands generates this model, including the hydrostatic fluid elements, material models, and boundary conditions.

```
/batch,list
/title, Sample input data for defining hydrostatic fluid elements
/prep7

!
! Define element types
!

et,1,182 !plane182
keyopt,1,3,1 !axisymmetric option
et,2,241 !hsfld241
keyopt,2,3,1
!
! Define material type
!

! Material properties for Steel
mp,ex,1,2.0e5 !Young's Modulus in N/mm^2
mp,nuxy,1,0.3 !Poisson's ratio
mp,dens,1,7.7e-9!Density in ton/mm^3

! Material properties for hydraulic fluid
tb,fluid,2,,,liquid
tbdata,1,2000.0 !Bulk Modulus in N/mm^2
tbdata,2,6.4e-4 !Coefficient of thermal expansion
tbdata,3,8.42e-5!Density in ton/mm^3

!
! Define nodes, keypoints, areas
!

! Nodes
n,1,0.0,50.0 !pressure node

! Keypoints
k, 1, 0.0, 0.0
k, 2,25.0, 0.0
k, 3,25.0,100.0
k, 4,10.0,100.0
k, 5,10.0, 95.0
k, 6,20.0, 95.0
k, 7,20.0, 5.0
k, 8, 0.0, 5.0
k, 9, 0.0, 80.0
k,10,19.0, 80.0
k,11,19.0, 85.0
k,12, 2.5, 85.0
k,13, 2.5,185.0
k,14, 0.0,185.0

! Areas
a,1,2,3,4,5,6,7,8
a,9,10,11,12,13,14

!
! Define mesh
!

! Planar elements
type,1
mat,1
esize,2.5
amesh,all

! Hydrostatic fluid elements--automatic generation
type,2
mat,2
lsel,s,line,,6,9
nsll,s,1
esurf,1
allsel,all
```



```

! Hydrostatic fluid elements--E command
e,61,162,,1 !element with negative volume--nodes 61 and 162 swapped

!
! Define degree-of-freedom constraints (boundary conditions)
!

! Symmetry boundary condition on pressure node
d,1,ux,0.0

! Symmetry boundary condition on planar mesh
lsel,s,line,,8,14,6
nsl1,s,1
d,all,ux,0.0
allsel,all

! Fixed boundary condition on planar mesh
lsel,s,line,,1
nsl1,s,1
d,all,all,0.0
allsel,all

finish

```

## 18.5. Results Output

Hydrostatic pressure can be output as the HDSP degree-of-freedom solution. The solution value is the change in hydrostatic pressure from the value at the start of the analysis. In the case of compressible gas, the solution value is the change in hydrostatic pressure with respect to the reference pressure. If any pressure value is specified using the **IC** command, then the value is added to the solution.

The following element output quantities are available for the [HSFLD241](#) and [HSFLD242](#) elements:

Output Quantity	Symbol	Meaning
DENSITY	$\rho$	Current fluid density
TVOL	$V$	Current total fluid volume
TMAS	$M$	Current total fluid mass
MFLO	$w$	Current fluid mass flow rate
TVOLO	$V_o$	Original total fluid volume

These NMISC output quantities provide results for the entire fluid volume, so all hydrostatic fluid elements sharing the pressure node have the same output value. See the [HSFLD241](#) and [HSFLD242](#) element descriptions for more information about output for these elements.



---

## Appendix A. Example Analyses with Multiple Imposed Rotations

The examples shown here demonstrate how the results change when the order of imposed rotations is changed in a large deflection analysis.

### A.1. Problem Description

A robotic arm, modeled with one BEAM188 element, is initially aligned along the global Z-axis with one end at the origin. On the node at the origin displacements are fixed, and non-zero rotations are applied in all three directions. Two ways of applying rotations are shown: sequential and simultaneous. Animations are provided to show the differences in results.

The rotations are applied in the following manner:

**Sequential rotations** - The beam is first rotated by 60° clockwise about the X-axis, followed by a 90° counterclockwise rotation about the local z-axis, and finally another 90° counterclockwise rotation about the local x-axis. The rotations about each axis are imposed in separate load steps.

**Simultaneous rotations** - The total values for the compound rotations are the same as the sequential case, but the rotations are imposed in a single load step. The beam is rotated by 60° clockwise about the X-axis, by  $(90^\circ)(\sin(60^\circ)+\cos(60^\circ))$  counterclockwise about the Y-axis and by  $(90^\circ)(\sin(60^\circ)-\cos(60^\circ))$  clockwise about the Z-axis.

For the case of sequentially applied rotations, two sample inputs are given: one with imposed rotations and the other with imposed rotational velocities. This demonstrates that the use of imposed rotations and imposed rotational velocities both achieve the same results in a static analysis.

### A.2. Sample Inputs for Imposed Rotations

The following input listing shows the model setup, including node and element definitions, material definitions, and section definitions. This PREP7 input was used for all the examples in this discussion.

#### Beam Model Setup

```
/BATCH,LIST
/PREP7
!Parameters
PI=3.1415926536

!Element type definitions
ET,1,188 !Beam element

!Node definitions
N,1,0.,0.,0.
N,2,0.,0.,0.6

!Material properties
MP,EX,1,2.01E11 !Young's modulus for Steel
MP,NUXY,1,0.3 !Poisson's ratio

!Section properties
SECTYPE,1,BEAM,RECT
SECDATA,0.1,0.2 !Beam cross-section
```

```

!Element definitions
TYPE,1
MAT,1
REAL,1
SECNUM,1
EN,1,1,2

FINISH

```

## A.2.1. Sequentially Applied Rotations

The following input illustrates compound rotations imposed over multiple load steps with each component applied sequentially on top of the previous deformed configuration.

### Example 1: Sequentially Applied Rotations

```

/prep7
!parameters
pi=3.1415926536

!element type definitions
et,1,188 !beam element

!node definitions
n,1,0.,0.,0.
n,2,0.,0.,0.6

!material properties
mp,ex,1,2.01e11 !young's modulus for steel
mp,nuxy,1,0.3 !poisson's ratio

!section properties
sectype,1,beam,rect
secdata,0.1,0.2 !beam cross-section

!element definitions
type,1
mat,1
real,1
secnum,1
en,1,1,2

finish

/solu !solution
nlgeom,on !geometrically nonlinear analysis

!boundary conditions for load step 1
d,1,ux,0.0 !displacement at node 1 fixed in x-direction
d,1,uy,0.0 !displacement at node 1 fixed in x-direction
d,1,uz,0.0 !displacement at node 1 fixed in x-direction
d,1,rotx,-pi/6.0 !60° clockwise rotation at node 1 about x-direction
d,1,roty,0.0 !rotation at node 1 fixed about y-direction
d,1,rotz,0.0 !rotation at node 1 fixed about z-direction
time,1.0 !time at the end of load step 1
nsub,10,100,10 !initial, maximum and minimum number of substeps
outres,all,all !save results
solve !solve load step 1

!boundary conditions for load step 2
!60° clockwise rotation at node 1 about x-axis (from load step 1)
d,1,rotx,-pi/6.0
!y-component of 90° counterclockwise rotation at node 1 about local z-axis
d,1,roty,(pi/2.0)*sin(pi/6.0)
!z-component of 90° counterclockwise rotation at node 1 about local z-axis
d,1,rotz,(pi/2.0)*cos(pi/6.0)
time,2.0 ! time at the end of load step 2
nsub,10,100,10
outres,all,all

```

```

solve !solve load step 2

!boundary conditions for load step 3
!60° clockwise rotation at node 1 about x-axis (from load step 1)
d,1,rotx,-pi/6.0
!y-component of 90° counterclockwise rotation at node 1 about local z-axis (from load step 2)
!and y-component of 90° counterclockwise rotation at node 1 about local x-axis
d,1,roty,(pi/2.0)*sin(pi/6.0)+(pi/2.0)*cos(pi/6.0)
!z-component of 90° counterclockwise rotation at node 1 about local z-axis (from load step 2)
!and z-component of 90° counterclockwise rotation at node 1 about local x-axis
d,1,rotz,(pi/2.0)*cos(pi/6.0)-(pi/2.0)*sin(pi/6.0)
time,3.0 !time at the end of load step 3
nsub,10,100,10
outres,all,all
solve !solve load step 3
finish

/post26 !time-history postprocessor
nsol,2,1,rot,x,rotx1 !store results for rotation about x-axis at node 1
nsol,3,1,rot,y,roty1 !store results for rotation about y-axis at node 1
nsol,4,1,rot,z,rotz1 !store results for rotation about z-axis at node 1
nsol,5,2,u,x,ux2 !store results for x-displacement at node 2
nsol,6,2,u,y,uy2 !store results for y-displacement at node 2
nsol,7,2,u,z,uz2 !store results for z-displacement at node 2
prvar,2,3,4 !print results
prvar,5,6,7
finish

```

The following input illustrates compound rotational velocities imposed over multiple load steps with each component applied sequentially. The difference between imposing rotations and rotational velocities should be noted.

### Example 2: Sequentially Applied Rotational Velocities

```

/prep7
!parameters
pi=3.1415926536

!element type definitions
et,1,188 !beam element

!node definitions
n,1,0.,0.,0.
n,2,0.,0.,0.6

!material properties
mp,ex,1,2.01e11 !young's modulus for steel
mp,nuxy,1,0.3 !poisson's ratio

!section properties
sectype,1,beam,rect
secdata,0.1,0.2 !beam cross-section

!element definitions
type,1
mat,1
real,1
secnum,1
en,1,1,2

finish

/solu !solution
nlgeom,on !geometrically nonlinear analysis
kbc,1 !step applied load/boundary condition

!boundary conditions for load step 1
d,1,ux,0.0 !displacement at node 1 fixed in x-direction

```

## Example Analyses with Multiple Imposed Rotations

---

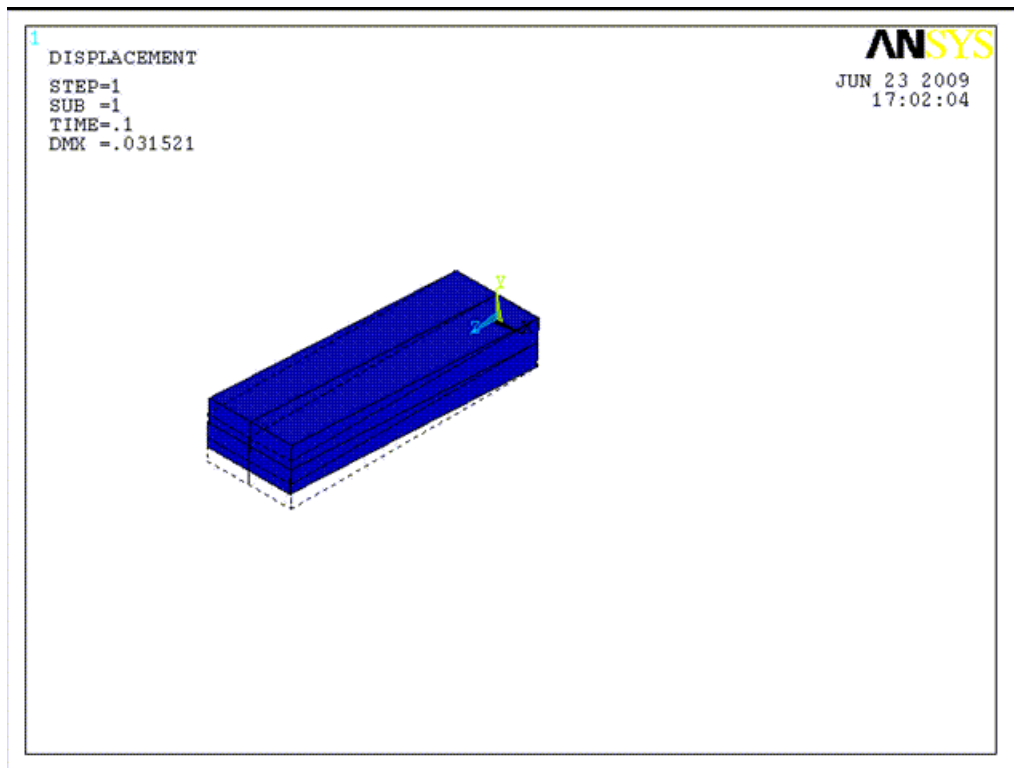
```
d,1,uy,0.0 !displacement at node 1 fixed in x-direction
d,1,uz,0.0 !displacement at node 1 fixed in x-direction
d,1,omgx,-pi/6.0 !60°/s clockwise rotational velocity at node 1 about x-direction
d,1,omgy,0.0 !zero rotational velocity at node 1 about y-direction
d,1,omgz,0.0 !zero rotational velocity at node 1 about z-direction
time,1.0 !time at the end of load step 1
nsub,10,100,10 !initial, maximum and minimum number of substeps
outres,all,all !save results
solve !solve load step 1

!boundary conditions for load step 2
!zero rotational velocity at node 1 about x-direction
d,1,omgx,0.0
!y-component of 90°/s counterclockwise rotational velocity at node 1 about local z-axis
d,1,omgy,(pi/2.0)*sin(pi/6.0)
!z-component of 90°/s counterclockwise rotational velocity at node 1 about local z-axis
d,1,omgz,(pi/2.0)*cos(pi/6.0)
time,2.0 ! time at the end of load step 2
nsub,10,100,10
outres,all,all
solve !solve load step 2

!boundary conditions for load step 3
!zero rotational velocity at node 1 about x-direction
d,1,omgx,0.0
!y-component of 90°/s counterclockwise rotational velocity at node 1 about local x-axis
d,1,omgy,(pi/2.0)*cos(pi/6.0)
!z-component of 90°/s counterclockwise rotational velocity at node 1 about local x-axis
d,1,omgz,-(pi/2.0)*sin(pi/6.0)
time,3.0 ! time at the end of load step 3
nsub,10,100,10
outres,all,all
solve !solve load step 3
finish

/post26 !time-history postprocessor
nsol,2,1,rot,x,rotx1 !store results for rotation about x-axis at node 1
nsol,3,1,rot,y,roty1 !store results for rotation about y-axis at node 1
nsol,4,1,rot,z,rotz1 !store results for rotation about z-axis at node 1
nsol,5,2,u,x,ux2 !store results for x-displacement at node 2
nsol,6,2,u,y,uy2 !store results for y-displacement at node 2
nsol,7,2,u,z,uz2 !store results for z-displacement at node 2
prvar,2,3,4 !print results
prvar,5,6,7
finish
```

The following animation shows how the beam deforms under the sequentially applied load. The result is the same for applied rotations and applied rotational velocities.

**Figure 1: Rotated Configuration Resulting from Sequentially Applied Rotations**

## A.2.2. Simultaneously Applied Rotations

The following input illustrates compound rotations imposed over a single load step with all components applied together.

### Example 3: Simultaneously Applied Rotations

```

/prep7
!parameters
pi=3.1415926536

!element type definitions
et,1,188 !beam element

!node definitions
n,1,0.,0.,0.
n,2,0.,0.,0.6

!material properties
mp,ex,1,2.01e11 !young's modulus for steel
mp,nuxy,1,0.3 !poisson's ratio

!section properties
sectype,1,beam,rect
secdata,0.1,0.2 !beam cross-section

!element definitions
type,1
mat,1
real,1
secnum,1
en,1,1,2

finish

```

## Example Analyses with Multiple Imposed Rotations

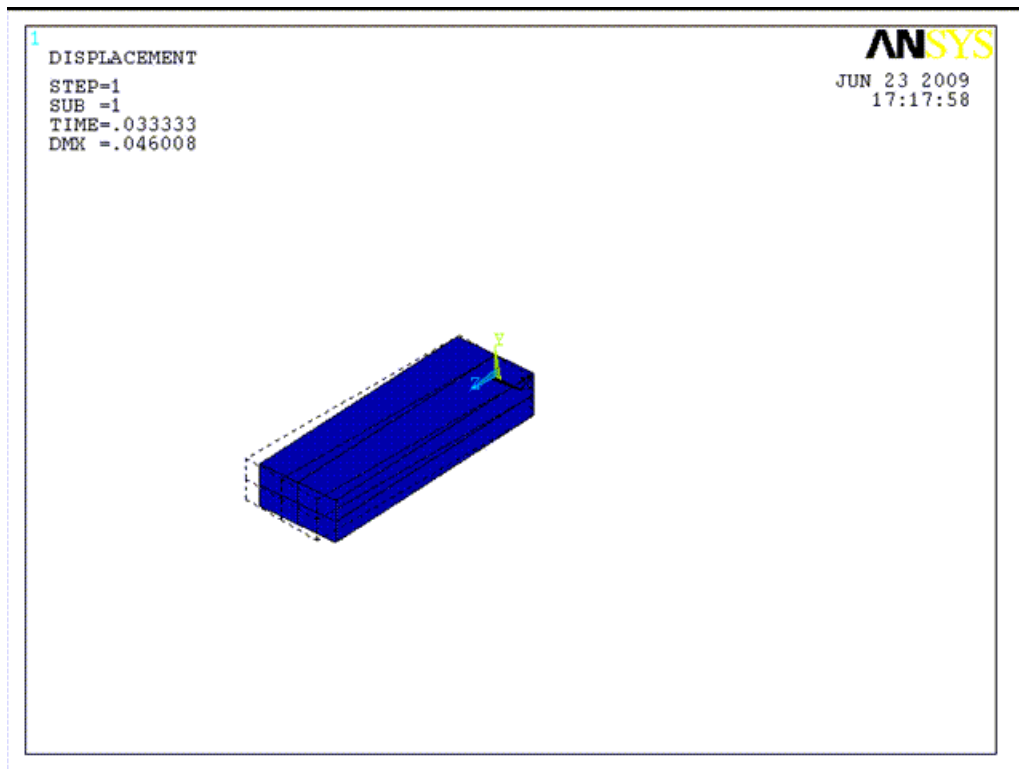
```
/solu !solution
nlgeom,on !geometrically nonlinear analysis

!boundary conditions for load step 1
d,1,ux,0.0 !displacement at node 1 fixed in x-direction
d,1,uy,0.0 !displacement at node 1 fixed in x-direction
d,1,uz,0.0 !displacement at node 1 fixed in x-direction
!60° clockwise rotation at node 1 about x-axis
d,1,rotx,-pi/6.0
!y-component of 90° counterclockwise rotation at node 1 about local z-axis
!and y-component of 90° counterclockwise rotation at node 1 about local x-axis
d,1,roty,(pi/2.0)*sin(pi/6.0)+(pi/2.0)*cos(pi/6.0)
!z-component of 90° counterclockwise rotation at node 1 about local z-axis
!and z-component of 90° counterclockwise rotation at node 1 about local x-axis
d,1,rotz,(pi/2.0)*cos(pi/6.0)-(pi/2.0)*sin(pi/6.0)
time,1.0 !time at the end of load step 1
nsub,30,100,30 !initial, maximum and minimum number of substeps
outres,all,all !save results
solve !solve load step 1
finish

/post26 !time-history postprocessor
nsol,2,1,rot,x,rotx1 !store results for rotation about x-axis at node 1
nsol,3,1,rot,y,roty1 !store results for rotation about y-axis at node 1
nsol,4,1,rot,z,rotz1 !store results for rotation about z-axis at node 1
nsol,5,2,u,x,ux2 !store results for x-displacement at node 2
nsol,6,2,u,y,uy2 !store results for y-displacement at node 2
nsol,7,2,u,z,uz2 !store results for z-displacement at node 2
prvar,2,3,4 !print results
prvar,5,6,7
finish
```

The following animation shows how the beam deforms under the simultaneously applied load.

**Figure 2: Rotated Configuration Resulting from Simultaneously Applied Rotations**





---

# Index

## A

- adequate mesh density, 266
- alpha (mass) damping, 2, 86, 159
- alternating stress intensity, 424
- amplitude
  - of harmonic loads, 82
- analysis options
  - harmonic analysis
    - mode-superposition method, 98
  - modal analysis, 80
  - nonlinear static structural analysis, 243
  - single-point spectrum analysis, 138
  - transient dynamic analysis
    - mode-superposition method, 122
- analysis type
  - buckling, 179
  - full transient, 113
  - harmonic, 77
  - modal, 39
  - nonlinear structural, 193
  - spectrum, 135
  - static structural, 9, 11
  - structural, 1
  - transient dynamic, 107
- animation
  - gasket results, 334
- anisotropic hyperelasticity, 210
- applying loads, 80, 251, 254
  - harmonic analysis, 82
  - increments for buckling analysis, 180
  - nonlinear analysis, 251
  - structural static analysis, 16-17
  - transient dynamic analysis, 117
  - using TABLE array parameters, 17
- arc-length method, 13, 181, 195
  - compared to nonlinear stabilization method, 264
  - results checking, 264
  - understanding, 263
- automatic time stepping, 132, 180, 270
  - creep, 213
  - large strain, 200

## B

- beam tool, 439
- beams
  - composite sections, 445
  - cross sections, 435
  - nonlinear general sections, 443
- beta (stiffness) damping, 2, 86, 138, 159

- birth and death, 270-271
  - element, 258
- bisection, 198
- Block Lanczos extraction method, 97
- Block Lanczos method, 41, 62
- nonlinear structural analysis
  - restarts, 254
- brake squeal (prestressed modal) analysis, 53
- buckling analysis, 179
  - eigenvalue
    - procedure, 182
  - introduction to, 179
  - methods of solution, 179
  - nonlinear
    - procedure, 180
  - obtain the eigenvalue buckling solution, 183
  - obtain the static solution, 182
- bulk modulus, 477

## C

- calculating
  - covariance in POST26, 167
- causes of nonlinear behavior, 194
  - changing status, 194
  - geometric nonlinearities, 194
  - material nonlinearities, 195
- cells
  - defining for a section, 437
- changing
  - status, 194
- combinations
  - material model examples, 216
- composite beam sections, 445
- composite cross section, 441
- composites
  - element types for, 409
  - FiberSIM-ANSYS interface, 416
  - guidelines, 414
  - how to model, 409
  - introduction to, 409
  - layer properties, 411
- concrete, 210, 257
- configurational force (see material force)
- constant damping ratio, 2, 86, 138, 159
- constraint equations
  - with QR damped method, 65
- contact, 194
  - node-to-node, 257
- containers, 475
- contour displays, 20, 143, 186, 252, 334
  - activating, 48
- control elements, 270

controlling cutbacks, 12  
controlling restarts, 12  
convergence  
  checking, 245  
  enhancements, 267  
  failure, 267  
convergence criteria, 245  
  default value, 245  
  disjointed structures, 245  
  displacement checking, 245  
  force checking, 245  
  vector norms, 245  
convergence tolerances, 12  
coupling, 414  
covariance  
  calculating, 167  
creating  
  sections, 436, 461  
  beam tool, 439  
creep, 201, 210  
  automatic time stepping, 213  
  explicit, 210, 249  
  explicit creep procedure, 213  
  implicit, 210, 249  
  implicit creep procedure, 211  
  ratio, 249  
creep criteria, 15, 249  
creep strain rate, 210  
cross sections (see sections) (see Sections)  
  composite, 441  
custom convergence criteria, 245  
cutback changes via SOLCONTROL, 247  
cutback criteria, 12

## D

damped method  
  modal analysis, 41, 64  
damping, 2, 116, 254  
  harmonic analysis, 86  
  spectrum analysis  
    PSD, 159  
  table of options, 2  
  transient dynamic analysis, 114, 116, 122  
database output, 11, 87, 101, 113, 122, 126, 243  
DDAM (see dynamic design analysis method)  
debonding, 404  
deep drawing, 213  
deformed shape  
  display, 20, 143, 186, 252  
  displaying, 334  
delamination, 381  
  in a contact analysis, 404

  meshing, 401  
  results, 402  
deleting  
  sections, 436, 461  
deterministic analysis, 137  
disjointed structures, 245  
display deformed shape, 89  
displaying  
  deformed shapes, 48, 252  
dynamic analyses  
  harmonic, 77  
  spectrum, 135  
  transient, 107  
dynamic design analysis method (DDAM), 136, 168

## E

eigenmodes  
  reusing eigenmodes, 68  
  reusing extracted eigenmodes, 69  
eigenvalue analysis, 39  
eigenvalue buckling, 179  
eigenvalue buckling solution  
  buckling analysis, 183  
eigenvalue calculation  
  shift point, 183  
eigenvalue extraction methods, 183  
eigenvalues, 39  
  number to be extracted, 183  
eigenvectors, 39  
elastomers, 202  
element birth and death, 258  
element damping, 2  
element loads, 97  
element types  
  structural, 2  
elements  
  associating sec IDs with, 436, 461  
engineering strain, 200  
engineering stress, 200  
epoxy, 409  
equation solvers, 12, 80, 244  
equilibrium iterations, 195, 246  
  number of, 12, 246  
error estimation, 20  
example input  
  nonlinear structural  
    transient, 256  
example modal analyses, 50  
example nonlinear analysis  
  command method, 281  
  GUI method, 273  
example problems

- beam analysis, 459
- buckling analysis, 192
- harmonic analysis, 96
- nonlinear analysis, 284
- single-point response spectrum analysis, 158
- static analysis, 37
- transient dynamic analysis, 133
- example spectrum analysis
  - GUI method, 146
- excitation direction, 138
- expand
  - single solution, 126
- expanded modes
  - number to use, 138
- expansion pass, 100
  - activating, 101, 126
  - harmonic analysis
    - mode-superposition method, 100
  - transient dynamic analysis
    - mode-superposition method, 126
- expansion pass options
  - harmonic analysis, 101
  - transient dynamic analysis, 126
- experimental data for hyperelasticity, 206
- explicit creep, 210
- explicit creep procedure, 213
- extra element shapes, 271
- extracting modes, 39
- extrapolation of results, 15, 87, 101, 126, 251

## F

- failure criteria, 413
  - maximum strain, 413
  - maximum stress, 413
  - Tsai-Wu, 413
- failures
  - convergence, 267
- fatigue
  - alternating stress intensity, 424
  - design stress-intensity value ( $S_m$ ), 425
  - evaluation procedure, 424
  - event, 424
  - event repetitions, 429
  - fatigue data file (Jobname.FATG), 432
  - guidelines for assembling events, 430
  - how calculated, 423
  - introduction to, 423
  - loading, 424
  - location, 424
  - manually stored stresses, 427
  - material properties, 425
  - S-N curve, 425

- sample input, 433
- scale factors, 429
- Sm-T curve, 425
- strain hardening exponents, 425
- stress concentration factors, 425
- stress locations, 425
- stresses at a cross-section, 428
- stresses from results file, 428
- fatigue calculations
  - activating, 432
- fiberglass, 409
- FiberSIM-ANSYS interface, 416
- files
  - Jobname.FATG, 432
  - .SECT, 439
- fluid, 477
- forced vibrations, 77
- forcing frequency range, 86
- fracture, 339
- fracture mechanics
  - additional sources of information, 378
  - analysis procedure, 347
  - calculating fracture parameters, 350
  - cohesive zone method, 347
  - crack growth simulation, 346
  - crack-tip region modeling, 348
  - energy-release rate, 341
  - energy-release rate evaluation (VCCT), 356
  - fracture modes, 339
  - Gurson's model method, 347
  - interface element method, 346
  - introduction to, 339
  - J-Integral, 340, 351
  - material force, 343, 371
  - numerical evaluation of parameters, 350
  - overview, 339
  - parameters, 340
  - stress-intensity factor, 342, 363
  - T-stress, 343, 370
  - VCCT, 356
- free-body analysis (see inertia relief)
- frequency range for expansion, 101
- frequency vector, 166
- frequency-dependent damping ratio, 159
- full harmonic analysis
  - procedure, 79
- full transient dynamic analysis
  - procedure, 109

## G

- gap conditions, 122
  - and QR Damped method, 122

- gas, 477
  - gasket joints simulation
    - applications, 320
    - element topologies, 318
    - elements, 319
    - formulation, 318
    - introduction to, 317
    - materials, 320
    - meshing, 327
    - output listing, 332
    - plotting data, 327
    - procedure, 317
    - results, 333
    - sample problem, 335
    - solution, 331
    - thickness, 319
  - gasket results
    - animation, 334
    - POST1, 334
    - POST26, 335
    - reviewing, 334
    - tabular listing, 334
  - general shell sections, 465
  - geometric nonlinearities, 194, 200
  - geometry data for sections, 436, 461
  - geometry data of a section, 437, 463
  - glass, 215
- ## H
- harmonic analysis
    - full method
      - procedure, 79
    - introduction to, 77
    - methods of solution, 78
    - mode-superposition method
      - procedure, 97
      - sample input, 103
    - prestressed, 104
    - uses for, 77
  - harmonic forcing frequency, 82
  - harmonic solutions
    - number of, 86
  - hydrostatic fluid, 475
  - hyperelasticity, 201-202
    - anisotropic, 210
    - Arruda-Boyce model, 204
    - Bergstrom-Boyce hyperviscoelastic option, 208
    - Blatz-Ko foam model, 205
    - experimental data, 206
    - Gent model, 205
    - large strain, 202
    - Mooney-Rivlin model, 203
    - Mullins effect option, 209
    - neo-Hookean model, 204
    - Ogden compressible foam model, 205
    - Ogden model, 203
    - polynomial form model, 204
    - response function, 206
    - user-defined option, 208
    - Yeoh model, 205
- ## I
- ideal gas law, 477
  - implicit creep, 210
  - implicit creep procedure, 211
  - incompressible fluid, 475
  - inertia relief, 17
    - calculating, 17
    - output, 18
  - initial accelerations, 110
  - initial conditions
    - transient dynamic analysis
      - full method, 110
      - mode-superposition method, 122
  - initial displacement, 110
  - initial velocity, 110
  - integration point, 251
  - integration time step, 107, 113, 122
    - guidelines for, 130
  - interface
    - delamination , 397
    - elements, 398
    - material model, 2
    - materials, 399
    - meshing, 401
    - results, 402
- ## K
- kinematic analysis, 254, 270
- ## L
- large deflection and load direction, 198
  - large deformation effects, 11, 113, 180
  - large strain
    - analyses, 200
    - automatic time stepping, 200
    - hyperelasticity, 202
    - logarithmic strain, 200
    - true stress, 200
    - viscoplasticity, 213
  - large-deformation effects, 200, 254
  - layer orientation angle, 411
  - layer thickness, 411
  - layered elements, 409

libraries  
  sections, 450, 465

library paths  
  for cross sections, 436

line element results, 20, 48

line search option, 12, 247, 270

linear perturbation, 287

listing  
  section properties, 436, 461  
  section results, 436

listings  
  tabular, 49

load step options, 101, 126  
  eigenvalue buckling, 183  
  harmonic analysis  
    full method, 85  
    mode-superposition method, 98  
  modal analysis, 46  
  mode-superposition transient dynamic analysis, 122  
  Newton-Raphson, 247  
  spectrum analysis  
    PSD, 159  
    single-point, 138  
  time, 122  
  transient dynamic analysis, 117  
    full method, 117  
    mode-superposition method, 122

load steps, 195  
  applying for transient loading, 117

load types, 15  
  harmonic analysis, 82

load vector, 97, 122

load-versus-time curve, 110

loads  
  applying, 16, 84, 122, 180, 251, 254  
  applying for transient loading, 117  
  direction in a large-deflection analysis, 198  
  harmonic analysis, 82  
    mode-superposition method, 98  
  modal analysis, 44  
  nonlinear analysis, 251  
  rotations in a large-deflection analysis, 199, 483  
  structural static analysis, 16-17  
  transient dynamic analysis, 117  
    full method, 117  
    mode-superposition method, 122

Loads  
  applying, 80

logarithmic strain, 200

**M**

macros, 18

mass damping, 2

mass flow rate, 477

mass matrix formulation, 14, 80, 116

material force, 343

material model  
  combination examples, 216

material model interface, 2

material nonlinearities, 195, 201

material properties  
  fatigue, 425  
  nonlinear, 201

material-dependent damping, 2

maximum strain failure criterion, 413

maximum stress failure criterion, 413

memory saving  
  when generating stiffness matrix, 244

meshing  
  a line for a section, 438

metal forming, 213

middle step residual, 130

modal analysis  
  applying loads, 44  
  brake squeal (prestressed), 53  
  introduction to, 39  
  large-deflection prestressed, 50  
  listing natural frequencies, 48  
  mode-extraction methods, 41, 61  
  options, 40  
  prestressed, 49  
  process, 39  
  residual vector method, 66

modal damping, 2, 138

modal solution  
  obtaining, 121  
  single-point response spectrum analysis, 137

mode numbers, 15

mode superposition  
  harmonic analysis, 97  
  mode-extraction methods, 97

mode-extraction methods, 41, 61  
  Block Lanczos, 41, 62, 97  
  damped, 41, 64  
  PCG Lanczos, 41, 62  
  QR Damped, 41, 97-98, 121, 137  
  QR damped, 65, 122  
  Subspace, 64  
  Supernode, 41, 63  
  unsymmetric, 41, 64

mode-frequency analysis, 39

mode-superposition  
  transient dynamic analysis, 120

mode-superposition method, 79, 109

mode-superposition transient dynamic analysis  
  obtain the modal solution, 121  
  obtain the mode-superposition transient solution,  
  122  
modes  
  mass matrix formulation, 44  
  number to expand, 43  
  number to extract, 43  
  prestress effects calculation, 44  
  residual vectory calculation, 44  
monitor results in real time, 250  
MPRS analysis, 173  
multi-point response spectrum, 136

## N

Newton-Raphson, 195  
Newton-Raphson option, 14  
Newton-Raphson options, 247  
nodal forces and moments, 20  
nodal solution listing format, 101  
nonlinear analysis  
  tips and guidelines, 265  
nonlinear behavior  
  causes, 194  
nonlinear diagnostics, 267  
nonlinear general beam sections, 443  
nonlinear stabilization  
  activating, 259  
  artificial dashpot elements, 261  
  compared to arc-length method, 264  
  constant or reduced, 260  
  controlling, 259  
  energy dissipation ratio, 259  
  results checking, 262  
  tips and hints, 262  
  understanding, 258  
nonlinear structural analysis  
  automatic time stepping, 270  
  buckling, 180  
  conservative vs. nonconservative, 197  
  creep, 210  
  example transient input, 256  
  geometric nonlinearities, 200  
  hyperelasticity, 202  
  interpolation of results, 252  
  introduction to, 193  
  levels of operation, 195  
  load directions, 198  
  material nonlinearities, 201  
  nonlinear elements, 257  
  swelling, 216  
  termination, 254

  verifying results, 271  
  viscoelasticity, 215  
  viscoplasticity, 213  
nonlinear transient analysis  
  procedure, 254  
number of eigenvalues to be extracted, 183  
number of harmonic solutions, 86  
number of integration points per layer, 411  
number of solutions to expand, 101

## O

obtain the modal solution, 97  
offsets  
  for sections, 436, 461  
output, 271  
  database and results file, 11, 87, 101, 113, 122, 126,  
  243  
  printed, 15, 87, 101, 122, 126, 251  
output control, 87, 122  
output controls, 47, 126, 183

## P

participation factor calculations, 47, 159  
path dependency, 197  
PCG Lanczos method, 41, 62  
perturbation analysis, 287  
phase angle for expansion, 101  
phase angle of harmonic loads, 82  
plastic, 201  
plasticity, 270  
  rate-dependent, 213  
plastics, 215  
plotting  
  sections, 436, 461  
POST1, 19, 88, 103, 118, 143, 145, 165, 186, 251, 424  
  using, 89, 119  
POST26, 19, 88, 103, 118, 166, 251  
  using, 88, 118  
postprocessing  
  fatigue, 423  
  for different analysis (see reviewing results)  
power spectral density (PSD), 136  
predictor, 246  
predictor-corrector option, 12, 246  
preintegrated composite beam sections, 445  
preintegrated general shell sections, 465  
pressure loads  
  in a large-deflection analysis, 198  
prestress effects calculation, 14, 182  
prestressed analysis  
  harmonic, 104  
  mode superposition harmonic, 105

---

- transient, 129
- prestressed full transient dynamic analysis
  - procedure, 129
- prestressed harmonic analysis
  - procedure, 104
- prestressed modal analysis, 49
- prestressed modal analysis of a large deflection, 50
- prestressed mode-superposition harmonic analysis
  - procedure, 105
- prestressed mode-superposition transient dynamic analysis
  - procedure, 130
- printed output, 15, 87, 101, 122, 126, 251
- probabilistic analysis, 137
- PSD analysis, 159
- PSD type, 159
- PSD-versus-frequency table, 159

## Q

- QR Damped extraction method
  - harmonic analysis, 97
- QR damped method
  - modal analysis, 41, 65
  - transient analysis, 122
- QR Damped method
  - gap conditions, 122
  - harmonic analysis, 98
  - spectrum analysis, 137
  - transient analysis, 121

## R

- ramped loads, 86, 114, 122, 254
- random vibrations, 159
  - and power spectral density, 136
- range counting, 432
- rate-dependent plasticity , 213
- ratio
  - creep, 249
- Rayleigh damping constants, 2
- reaction forces and moments, 20
- reading
  - user-defined mesh, 436
- real constants
  - for layered elements, 411
- reference temperature, 14
- reinforcing
  - assumptions, 469
  - defining sections and elements, 470
  - introduction to, 469
  - modeling options, 469
  - simulation and postprocessing, 474
  - smeared, 472

- relative stress distributions
  - buckling analysis, 186
- residual vector method for modal analysis, 66
- response function for hyperelasticity, 206
- response PSD
  - calculating, 166
- response spectrum
  - multi-point, 135
  - single-point, 135
  - type, 138
- restart control, 12
- restarts, 257
- results
  - data from a PSD analysis, 165
  - extrapolation of, 15, 87, 101, 126, 251
- results file output, 11, 87, 101, 113, 122, 126, 243
- reviewing results
  - eigenvalue buckling, 186
  - expanded solution
    - mode-superposition method, 103
  - fatigue, 432
  - harmonic analysis
    - full method, 88
  - modal analysis, 47
  - nonlinear static structural, 251
  - nonlinear transient structural, 256
  - spectrum analysis
    - PSD, 164
    - single-point, 143
  - static structural analysis, 19
  - transient dynamic analysis
    - full method, 118
    - mode-superposition method, 126
- rolling, 213
- rubber, 202

## S

- sample buckling analysis
  - command method, 191
  - GUI method, 186
- sample harmonic analysis
  - command method, 96
  - GUI method, 90
- sample input
  - fatigue, 433
  - full transient dynamic analysis, 119
  - harmonic analysis
    - mode-superposition method, 103
  - mode-superposition transient analysis, 128
  - spectrum analysis
    - PSD, 167
  - transient dynamic analysis

- mode-superposition method, 128
- sample spectrum analysis
  - command method, 152
- sample static analysis
  - command method, 34
  - GUI method, 22
- sandwich structures, 412
- scaling the load vector, 98
- .SECT file, 439
- sections, 461
  - associating subtypes with IDs, 436, 461
  - creating, 436, 461
    - beam tool, 439
  - defining dimensions, 436, 461
  - geometry data of, 437, 463
  - introduction to, 435
  - libraries of, 450, 465
  - meshing a line for, 438
  - solid, 437
  - thin wall, 437
- shape memory alloy, 213
- shells, 461
  - preintegrated general sections, 465
- shift point for eigenvalue calculation, 183
- single solution to expand, 126
- single-point response spectrum, 136
  - process, 137
- single-point spectrum analysis
  - load step option for, 138
- small deflection, 200
- small strain analyses, 200
- S-N curve, 425
- solid sections, 437
- solution
  - obtain the mode-superposition transient solution, 122
- solution controls, 242
- solution controls dialog box
  - full transient analysis, 113
    - defining analysis type and options, 113
  - static analysis, 11
- solution listing format, 80
- solution method
  - harmonic response, 80
- solution methods, 78, 108
  - full, 108
    - transient dynamic analysis, 78
    - mode superposition, 79
    - mode-superposition , 109
- solution termination, 13
- solutions to be expanded, 126
- solver
  - memory saving option, 244
- solvers
  - equation, 12, 244
- sparse direct solver, 80
- spectral-value-versus-frequency curve, 138
- spectrum analysis, 2
  - DDAM, 168
  - introduction to, 135
  - multi-point, 173
  - PSD
    - procedure, 159
    - sample input, 167
  - random vibration (PSD)
    - obtain the spectrum solution, 159
  - running multipile
    - single-point, 145
  - single-point
    - process, 137
  - single-point response
    - obtain the modal solution, 137
    - obtain the spectrum solution, 138
- spectrum data, 159
- spectrum solution
  - random vibration (PSD) analysis, 159
  - single-point response spectrum analysis, 138
- spectrum type
  - single-point response, 138
- stability analysis, 179
- static solution
  - buckling analysis, 182
- static structural analysis
  - introduction to, 9
  - loading, 16
  - procedure, 9
- stepped loads, 86, 114, 122, 254
- stiffness matrix
  - memory saving option, 244
- strain
  - creep, 210
- strain rate
  - creep, 210
- strain rate effect, 12
- stress calculations, 101
- stress concentration factors, 425
- stress relaxation, 210
- stress stiffening, 200
- stress stiffening effects, 13, 129, 247
- structural analyses
  - buckling, 179
  - fatigue, 423
  - harmonic, 77
  - modal, 39



---

- nonlinear, 193
- spectrum, 135
- static, 9
- transient dynamic, 107
- structural analysis, 1
  - element types, 2
  - nonlinear, 193
- structural energy error estimation, 20
- Subspace method, 64
- substeps, 195, 198, 243
- Supernode method, 41, 63
- swelling, 201, 216
- swelling subroutine, 216

**T**

- TABLE array parameters, 17
- tabular listings, 20, 49, 89, 143, 252
- tapered beams, 441
- termination
  - solution, 13
- thin wall sections, 437
- time, 129
- time integration effects, 114, 129
- time step optimization
  - thermal, 249
- time steps, 195, 198, 200, 243
  - automatic, 11, 113, 132, 180, 244
  - number of, 11, 113, 270
  - size of, 11, 113, 270
- time-integration effects, 254
- transient dynamic analysis
  - full method
    - procedure, 109
  - introduction to, 107
  - load step options for, 117
  - methods of solution, 108
  - middle step residual, 130
  - mode-superposition method
    - procedure, 120
  - preparing for, 107
  - prestressed, 129
- transient integration parameters, 114, 122, 254
- true strain, 200
- true stress, 200
- Tsai-Wu failure criterion, 413

**U**

- unstable structures
  - arc-length method, 263
  - nonlinear stabilization, 258
  - techniques for solving, 258
- unsymmetric method, 41, 64

- user mesh files, 450
- userswstrain, 216

**V**

- VCCT
  - crack growth simulation, 381
- VCCT energy-release rate evaluation, 356
- vector displays, 20, 143
- vector norms, 245
- vector plots, 89
- vibration analysis, 39
- viscoelasticity, 201, 215
  - element types, 215
- viscoplasticity, 201, 213
  - Anand option, 213
  - element types, 213
  - exponential visco-hardening (EVH) option, 213
  - large strain, 213
  - Peirce option, 213
  - Perzyna option, 213

**W**

- wave propagation effects, 130
- writing
  - user mesh, 436

