ANSYS Basic Analysis Procedures Guide

ANSYS Release 8.1

ANSYS Basic Analysis Procedures Guide

ANSYS Release 8.1

Trademark Information

Table of Contents

List of Figures

List of Tables

Chapter 1: Getting Started with ANSYS

1.1. Performing a Typical ANSYS Analysis

The ANSYS program has many finite element analysis capabilities, ranging from a simple, linear, static analysis to a complex, nonlinear, transient dynamic analysis. The analysis guide manuals in the ANSYS documentation set describe specific procedures for performing analyses for different engineering disciplines. The next few sections of this chapter cover general steps that are common to most analyses.

A typical ANSYS analysis has three distinct steps:

1.   Build the model.

2.   Apply loads and obtain the solution.

3.   Review the results.

1.2. Building a Model

Building a finite element model requires more of your time than any other part of the analysis. First, you specify a jobname and analysis title. Then, you use the PREP7 preprocessor to define the element types, element real constants, material properties, and the model geometry.

1.2.1. Specifying a Jobname and Analysis Title

This task is not required for an analysis, but is *recommended*.

1.2.1.1. Defining the Jobname

The *jobname* is a name that identifies the ANSYS job. When you define a jobname for an analysis, the jobname becomes the first part of the name of all files the analysis creates. (The extension or suffix for these files' names is a file identifier such as **.DB**.) By using a jobname for each analysis, you ensure that no files are overwritten.

If you do not specify a jobname, all files receive the name **FILE** or **file**, depending on the operating system. You can change the default jobname as follows:

· By using the initial jobname entry option when you enter the ANSYS program, either via the launcher or on the ANSYS execution command.

· From within the ANSYS program, you can use either of the following:
  Command(s): /FILNAME
  GUI: Utility Menu> File> Change Jobname

The **/FILNAME** command is valid only at the Begin level. It lets you change the jobname even if you specified an initial jobname at ANSYS entry. The jobname applies only to files you open after using **/FILNAME** and not to files that were already open. If you want to start new files (such as the log file, **Jobname.LOG**, and error file **Jobname.ERR**) when you issue **/FILNAME**, set the Key argument on **/FILNAME** to 1. Otherwise, those files that were already open will still have the *initial* jobname.

1.2.1.2. Defining an Analysis Title

The **/TITLE** command (**Utility Menu> File> Change Title**), defines a title for the analysis. ANSYS includes the title on all graphics displays and on the solution output. You can issue the **/STITLE** command to add subtitles; these will appear in the output, but not in graphics displays.

1.2.1.3. Defining Units

The ANSYS program does not assume a system of units for your analysis. Except in magnetic field analyses, you can use any system of units so long as you make sure that you use that system for all the data you enter. (Units must be consistent for all input data.)

For micro-electromechanical systems (MEMS), where dimensions are on the order of microns, see the conversion factors in Section 1.3: System of Units in the *ANSYS Coupled-Field Analysis Guide*.

Using the **/UNITS** command, you can set a marker in the ANSYS database indicating the system of units that you are using. This command *does not* convert data from one system of units to another; it simply serves as a record for subsequent reviews of the analysis.

1.2.2. Defining Element Types

The ANSYS element library contains more than 150 different element types. Each element type has a unique number and a prefix that identifies the element category: BEAM4, PLANE77, SOLID96, etc. The following element categories are available:

BEAM	MESH
CIRCUit	Multi-Point Constraint
COMBINation	PIPE
CONTACt	PLANE
FLUID	PRETS (Pretension)
HF (High Frequency)	SHELL
HYPERelastic	SOLID
INFINite	SOURCe
INTERface	SURFace
LINK	TARGEt
MASS	TRANSducer
MATRIX	USER
	VISCOelastic (or viscoplastic)

The element type determines, among other things:

- The degree-of-freedom set (which in turn implies the discipline - structural, thermal, magnetic, electric, quadrilateral, brick, etc.)
- Whether the element lies in 2-D or 3-D space.

BEAM4, for example, has six structural degrees of freedom (UX, UY, UZ, ROTX, ROTY, ROTZ), is a line element, and can be modeled in 3-D space. PLANE77 has a thermal degree of freedom (TEMP), is an 8-node quadrilateral element, and can be modeled only in 2-D space.

You must be in PREP7, the general preprocessor, to define element types. To do so, you use the ET family of commands (**ET**, **ETCHG**, etc.) or their GUI path equivalents; see the *ANSYS Commands Reference* for details. You define the element type by name and give the element a type reference number. For example, the commands shown below define two element types, BEAM4 and SHELL63, and assign them type reference numbers 1 and 2 respectively.

```
ET,1,BEAM4
ET,2,SHELL63
```

This table of type reference number versus element name is called the *element type table*. While defining the actual elements, you point to the appropriate type reference number using the **TYPE** command (**Main Menu> Preprocessor> Modeling> Create> Elements> Elem Attributes**).

Many element types have additional options, known as KEYOPTs, and are referred to as KEYOPT(1), KEYOPT(2), etc. For example, KEYOPT(9) for BEAM4 allows you to choose results to be calculated at intermediate locations on each element, and KEYOPT(3) for SHELL63 allows you to suppress extra displacement shapes. You can specify KEYOPTs using the **ET** command or the **KEYOPT** command (**Main Menu> Preprocessor> Element Type> Add/Edit/Delete**).

1.2.3. Defining Element Real Constants

Element real constants are properties that depend on the element type, such as cross-sectional properties of a beam element. For example, real constants for BEAM3, the 2-D beam element, are area (AREA), moment of inertia (IZZ), height (HEIGHT), shear deflection constant (SHEARZ), initial strain (ISTRN), and added mass per unit length (ADDMAS). Not all element types require real constants, and different elements of the same type may have different real constant values.

You can specify real constants using the **R** family of commands (**R**, **RMODIF**, etc.) or their equivalent menu paths; see the *ANSYS Commands Reference* for further information. As with element types, each set of real constants has a reference number, and the table of reference number versus real constant set is called the *real constant table*. While defining the elements, you point to the appropriate real constant reference number using the **REAL** command (**Main Menu> Preprocessor> Modeling> Create> Elements> Elem Attributes**).

While defining real constants, keep these rules and guidelines in mind:

- When using one of the **R** commands, you must enter real constants in the order shown in Table 4.*n*.1 for each element type in the *ANSYS Elements Reference*.

- For models using multiple element types, use a separate real constant set (that is, a different REAL reference number) for each element type. The ANSYS program issues a warning message if multiple element types reference the same real constant set. However, a single element type may reference several real constant sets.

- To verify your real constant input, use the **RLIST** and **ELIST** commands, with $RKEY$ = 1 (shown below). **RLIST** lists real constant values for all sets. The command **ELIST**,,,,1 produces an easier-to-read list that shows, for each element, the real constant labels and their values.

  Command(s):
  ELIST

  GUI:

  Utility Menu> List> Elements> Attributes + RealConst
  Utility Menu> List> Elements> Attributes Only
  Utility Menu> List> Elements> Nodes + Attributes
  Utility Menu> List> Elements> Nodes + Attr + RealConst

  Command(s):
  RLIST

  GUI:

  Utility Menu> List> Properties> All Real Constants

Utility Menu> List> Properties> Specified Real Const

- For line and area elements that require geometry data (cross-sectional area, thickness, diameter, etc.) to be specified as real constants, you can verify the input graphically by using the following commands in the order shown:

    Command(s): /ESHAPE and **EPLOT**
    GUI: Utility Menu> PlotCtrls> Style> Size and Shape
    Utility Menu> Plot> Elements

ANSYS displays the elements as solid elements, using a rectangular cross-section for link and shell elements and a circular cross-section for pipe elements. The cross-section proportions are determined from the real constant values.

1.2.3.1. Creating Cross Sections

If you are building a model using BEAM44, BEAM188, or BEAM189, you can use the section commands (**SECTYPE**, **SECDATA**, etc.) or their GUI path equivalents to define and use cross sections in your models. See Chapter 16, "Beam Analysis and Cross Sections" in the *ANSYS Structural Analysis Guide* for information on how to use the BeamTool to create cross sections.

1.2.4. Defining Material Properties

Most element types require material properties. Depending on the application, material properties can be linear (see Section 1.2.4.1: Linear Material Properties) or nonlinear (see Section 1.2.4.2: Nonlinear Material Properties).

As with element types and real constants, each set of material properties has a material reference number. The table of material reference numbers versus material property sets is called the *material table*. Within one analysis, you may have multiple material property sets (to correspond with multiple materials used in the model). ANSYS identifies each set with a unique reference number.

While defining the elements, you point to the appropriate material reference number using the **MAT** command.

1.2.4.1. Linear Material Properties

Linear material properties can be constant or temperature-dependent, and isotropic or orthotropic. To define *constant* material properties (either isotropic or orthotropic), use one of the following:

    Command(s): MP
    GUI: Main Menu> Preprocessor> Material Props> Material Models

(See Section 1.2.4.4: Material Model Interface for details on the GUI.)

You also must specify the appropriate property label; for example EX, EY, EZ for Young's modulus, KXX, KYY, KZZ for thermal conductivity, and so forth. For isotropic material you need to define only the X-direction property; the other directions default to the X-direction value. For example:

```
MP,EX,1,2E11      ! Young's modulus for material ref. no. 1 is 2E11
MP,DENS,1,7800    ! Density for material ref. no. 1 is 7800
MP,KXX,1,43       ! Thermal conductivity for material ref. no 1 is 43
```

Besides the defaults for Y- and Z-direction properties (which default to the X-direction properties), other material property defaults are built in to reduce the amount of input. For example, Poisson's ratio (NUXY) defaults to 0.3, shear modulus (GXY) defaults to EX/2(1+NUXY)), and emissivity (EMIS) defaults to 1.0. See the *ANSYS Elements Reference* for details.

You can choose constant, isotropic, linear material properties from a material library available through the GUI. Young's modulus, density, coefficient of thermal expansion, Poisson's ratio, thermal conductivity and specific heat are available for 10 materials in four unit systems.

Caution: The property values in the material library are provided for your convenience. They are typical values for the materials you can use for preliminary analyses and noncritical applications. As always, *you* are responsible for all data input to the ANSYS program.

To define *temperature-dependent* material properties, you can use the **MP** command in combination with the **MPTEMP** or **MPTGEN** command. You also can use the **MPTEMP** and **MPDATA** commands. The **MP** command allows you to define a property-versus-temperature function in the form of a polynomial. The polynomial may be linear, quadratic, cubic, or quartic:

$$\text{Property} = C_0 + C_1 T + C_2 T^2 + C_3 T^3 + C_4 T^4$$

C_n are the coefficients and T is the temperature. You enter the coefficients using the *C0*, *C1*, *C2*, *C3*, and *C4* arguments on the **MP** command. If you specify just *C0*, the material property is constant; if you specify *C0* and *C1*, the material property varies linearly with temperature; and so on. When you specify a temperature-dependent property in this manner, the program internally evaluates the polynomial at discrete temperature points with linear interpolation between points (that is, piecewise linear representation) and a constant-valued extrapolation beyond the extreme points. You *must* use the **MPTEMP** or **MPTGEN** command *before* the **MP** command for second and higher-order properties to define appropriate temperature steps.

The second way to define temperature-dependent material properties is to use a combination of **MPTEMP** and **MPDATA** commands. **MPTEMP** (or **MPTGEN**) defines a series of temperatures, and **MPDATA** defines corresponding material property values. For example, the following commands define a temperature-dependent enthalpy for material 4:

```
MPTEMP,1,1600,1800,2000,2325,2326,2335        ! 6 temperatures (temps 1-6)
MPTEMP,7,2345,2355,2365,2374,2375,3000        ! 6 more temps (temps 7-12)
MPDATA,ENTH,4,1,53.81,61.23,68.83,81.51,81.55,82.31    ! Corresponding
MPDATA,ENTH,4,7,84.48,89.53,99.05,112.12,113.00,137.40 ! enthalpy values
```

If an unequal number of property data points and temperature data points are defined, the ANSYS program uses only those locations having both points defined for the property function table. To define a different set of temperatures for the next material property, you should first erase the *current* temperature table by issuing **MPTEMP** (without any arguments) and then define new temperatures (using additional **MPTEMP** or **MPTGEN** commands).

The **MPPLOT** command displays a graph of material property versus temperature. Figure 1.1: "Sample MPPLOT Display" shows a plot of the enthalpy-temperature curve defined in the example above. The **MPLIST** command lists material properties.

Figure 1.1 Sample MPPLOT Display
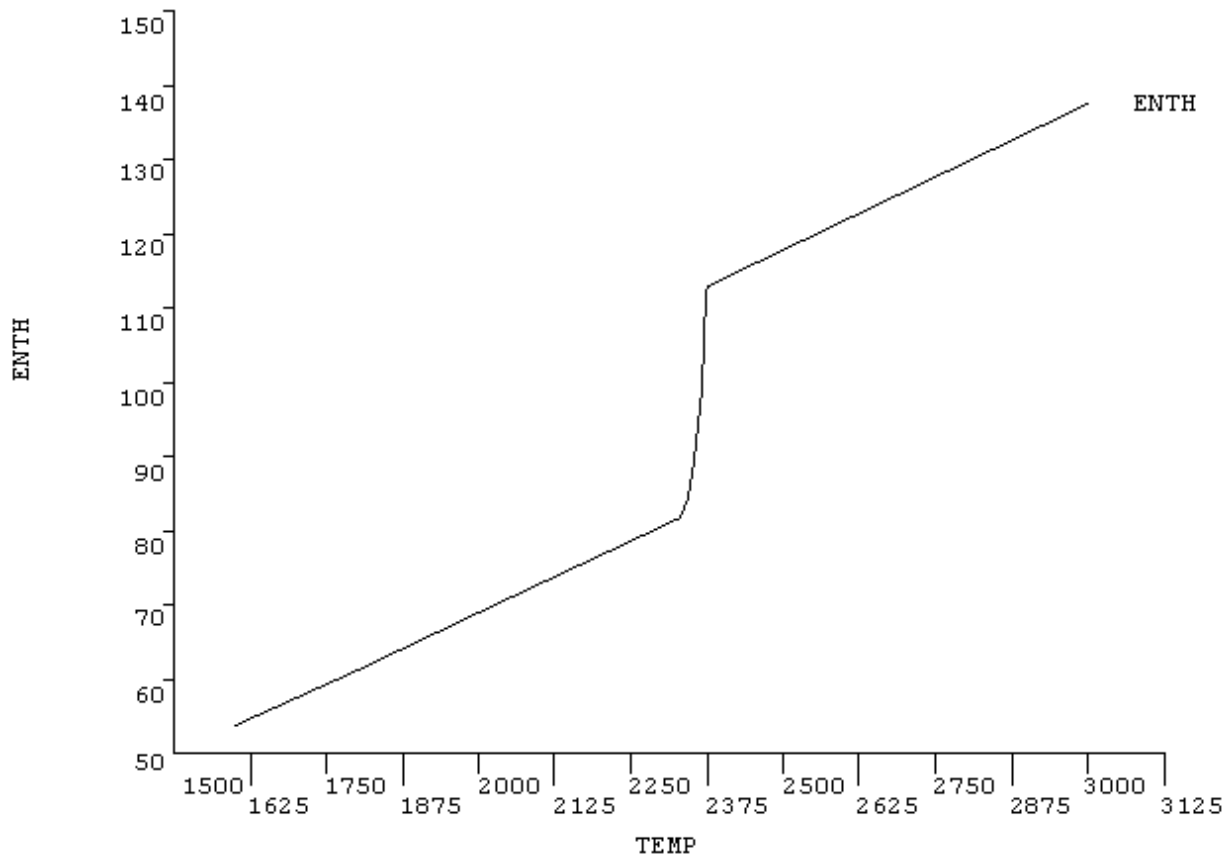
Following are some notes about temperature-dependent material properties:

*   To modify a property data point on an existing curve, simply redefine the desired data point by issuing **MPDATA** with the appropriate location number. For example, to change the ENTH value in location 6 of the above enthalpy-temperature curve from 82.31 to 83.09, the command would be **MPDATA**,ENTH,4,6,83.09

*   To modify a temperature data point on an existing curve, you need two commands: **MPTEMP** with the appropriate location number to specify the new temperature value, and **MPDRES** to associate the new temperature table with the material property. For example, to change the temperature in location 7 of the above enthalpy-temperature curve from 2345 to 2340, the commands would be:

```
MPTEMP,7,2340        ! Modifies location 7, retains other locations
MPDRES,ENTH,4        ! Associates ENTH for material 4 with new temps
```

You need to use the **MPDRES** command to modify stored properties. Whenever you define a temperature-dependent property, the temperature-property data pairs are immediately stored in the database. Modifying the temperature data points affects only material properties that are subsequently defined, not what is already stored. The **MPDRES** command forces modification of what is already stored in the database. Two additional fields on **MPDRES** allow you to modify a stored property and store it under a new label or a new material reference number.

The **MPTRES** command allows you to replace the current temperature table with that of a previously defined material property in the database. You can then use the previous temperature data points for another property.

For temperature-dependent secant coefficients of thermal expansion (ALPX, ALPY, ALPZ), if the base temperature for which they are defined (the *definition* temperature) differs from the reference temperature (the temperature at which zero thermal strains exist, defined by **MP**,REFT or **TREF**), then use the **MPAMOD** command to convert the data to the reference temperature. This conversion is not necessary when you input the thermal strains (THSX, THSY, THSZ) or the instantaneous coefficients of thermal expansion (CTEX, CTEY, CTEZ).

ANSYS accounts for temperature-dependent material properties during solution when element matrices are formulated. The materials are evaluated at once (at or near the centroid of the element) *or* at each of the integration points. For more information about how ANSYS evaluates temperature-dependent material properties, see Linear Material Properties.

You can save linear material properties (whether they are temperature-dependent or constant) to a file or restore them from a text file. (See Section 1.2.4.5: Using Material Library Files for a discussion of material library files.) You also can use **CDWRITE**,*MAT* to write both linear and nonlinear material properties to a file.

> *Note* — If you are using the **CDWRITE** command in any of the ANSYS-derived products (ANSYS Emag, ANSYS Professional, etc.), you must edit the **Jobname.CDB** file that **CDWRITE** creates to remove commands which are not available in the derived product. You must do this before reading the **Jobname.CDB** file.

1.2.4.2. Nonlinear Material Properties

Nonlinear material properties are usually tabular data, such as plasticity data (stress-strain curves for different hardening laws), magnetic field data (B-H curves), creep data, swelling data, hyperelastic material data, etc. The first step in defining a nonlinear material property is to activate a data table using the **TB** command (see Section 1.2.4.4: Material Model Interface for the GUI equivalent). For example, **TB**,BH,2 activates the B-H table for material reference number 2.

To enter the tabular data, use the **TBPT** command. For example, the following commands define a B-H curve:

```
TBPT,DEFI,150,.21
TBPT,DEFI,300,.55
TBPT,DEFI,460,.80
TBPT,DEFI,640,.95
TBPT,DEFI,720,1.0
TBPT,DEFI,890,1.1
TBPT,DEFI,1020,1.15
TBPT,DEFI,1280,1.25
TBPT,DEFI,1900,1.4
```

You can verify the data table through displays and listings using the **TBPLOT** or **TBLIST** commands.

Figure 1.2: "Sample TBPLOT Display" shows a sample **TBPLOT** (of the B-H curve defined above):

Figure 1.2 Sample TBPLOT Display
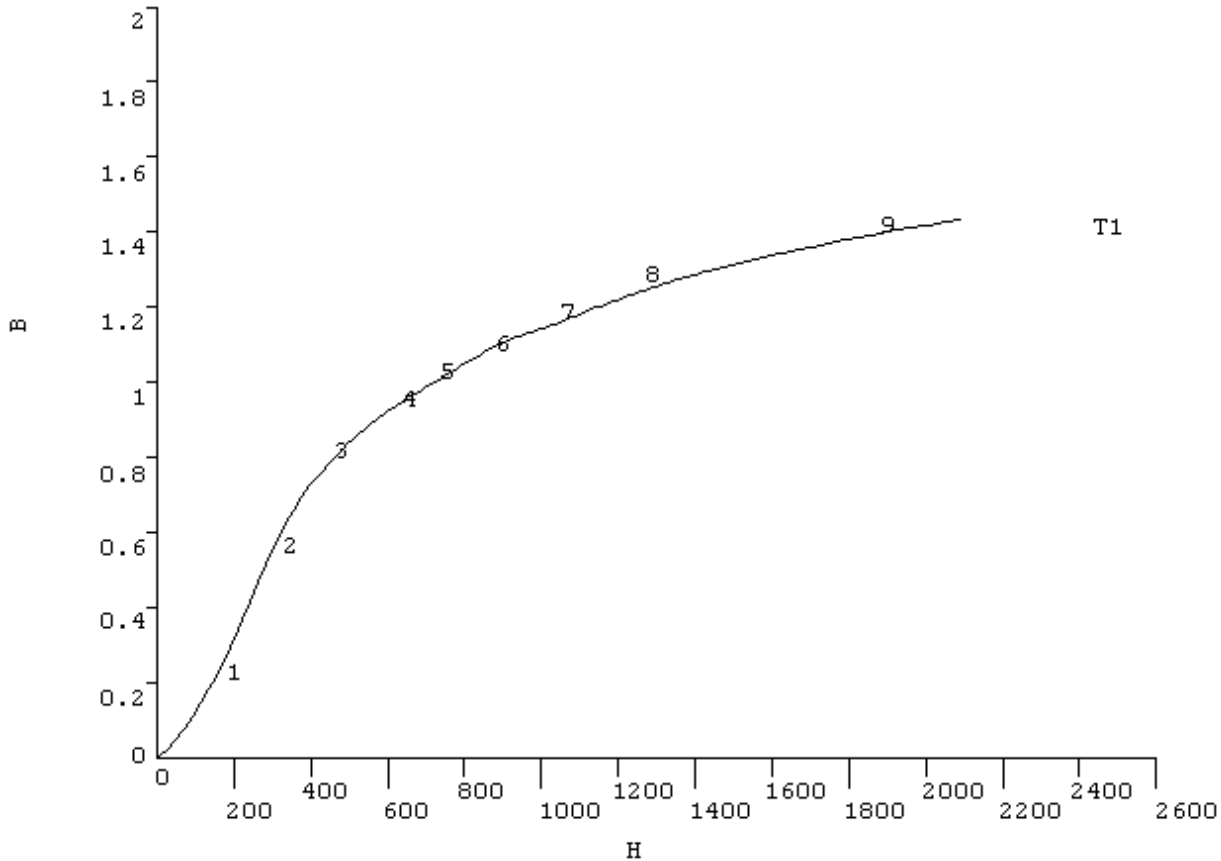
1.2.4.3. Anisotropic Elastic Material Properties

Some element types accept anisotropic elastic material properties, which are usually input in the form of a matrix. (These properties are different from anisotropic plasticity, which requires different stress-strain curves in different directions.) Among the element types that allow elastic anisotropy are SOLID64 (the 3-D anisotropic solid), PLANE13 (the 2-D coupled-field solid), SOLID5 and SOLID98 (the 3-D coupled-field solids).

The procedure to specify anisotropic elastic material properties resembles that for nonlinear properties. You first activate a data table using the **TB** command (with Lab = ANEL) and then define the terms of the elastic coefficient matrix using the **TBDATA** command. Be sure to verify your input with the **TBLIST** command. See *Data Tables - Implicit Analysis* in the *ANSYS Elements Reference* manual and the appropriate element descriptions for more information.

1.2.4.4. Material Model Interface

ANSYS includes an intuitive hierarchical tree structure interface for defining material models. A logical top-down arrangement of material categories guides you in defining the appropriate model for your analysis. You use this material model interface in all ANSYS analyses except for CFD analyses that require the use of any of the **FLDATA** family of commands.

1.2.4.4.1. Accessing the Interface

You access the material model interface from **Main Menu> Preprocessor> Material Props> Material Models**. The **Define Material Model Behavior** dialog box appears, which originally displays the top level of the tree structure, as shown in Figure 1.3: "Material Model Interface Initial Screen".

Figure 1.3 Material Model Interface Initial Screen
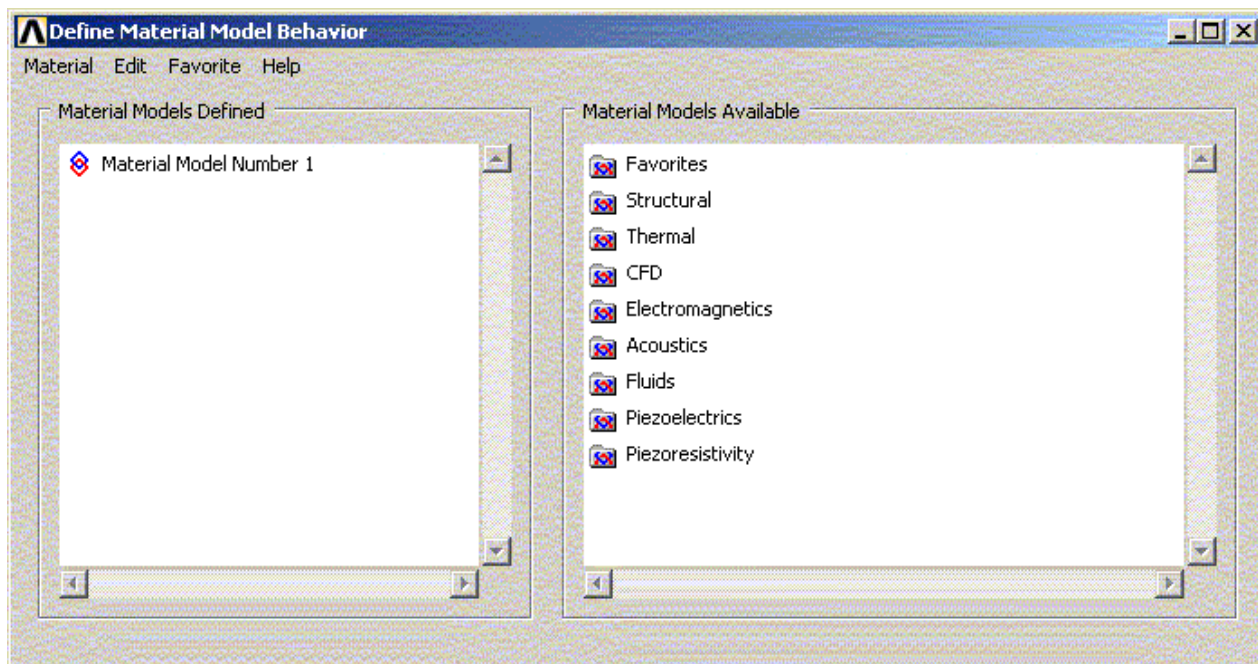
1.2.4.4.2. Choosing Material Behavior

The **Material Models Available** window on the right displays a list of material categories (for example, Favorites, Structural, Thermal, CFD, Electromagnetics).

> *Note* — If you choose an ANSYS LS-DYNA element type, only one category, LS-DYNA appears.

If a category is preceded by a folder icon, there are subcategories available under the main category. When you double-click on the category, the subcategories appear indented, and below the category as shown in Figure 1.4: "Material Model Interface Tree Structure".

Figure 1.4 Material Model Interface Tree Structure
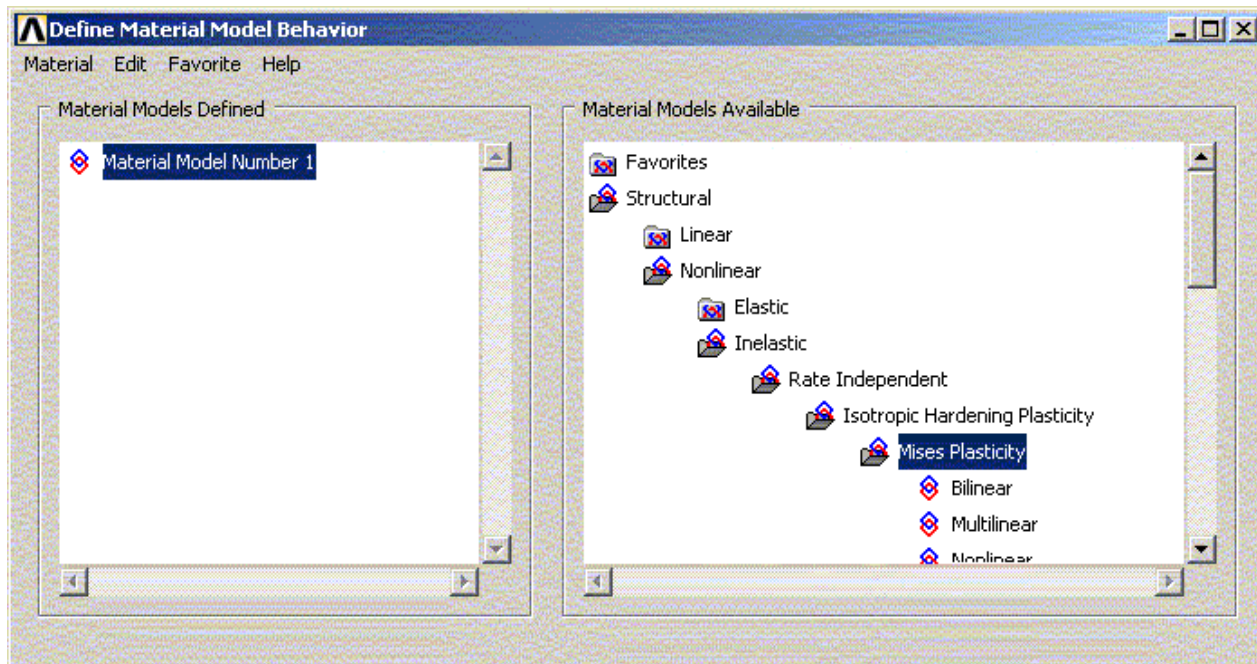
For example, under Structural are categories Linear, Nonlinear, and others. The models are further categorized so that you will eventually see a vertical list of material property sets or material models that are included under that category (for example, under von Mises Plasticity are: Bilinear, Multilinear, and Nonlinear). Once you have decided which material property set or model you will use, you then choose it by double-clicking on the item. A dialog box appears that prompts you for the required input data for *that* particular model or property set. Details of a data input dialog box are presented in Section 1.2.4.4.3: Entering Material Data.

Material Favorites Folder

A Material Favorite is a template of material properties. It is used as a short cut to frequently used properties, instead of navigating through the detailed tree structure each session. You can create a named template based on a currently defined material model through Favorite>New Favorite.You can also delete a named template through Favorite menu. For any consecutive sesisons of ANSYS, you will then be able to access this named template in the Favorites folder shown in the **Material Models Available** window.

1.2.4.4.3. Entering Material Data

Included in a data input dialog box is a table whose rows and columns you can alter depending on the requirements of the specific material property or model you have chosen. A typical data input dialog box is shown in Figure 1.5: "A Data Input Dialog Box".

Figure 1.5 A Data Input Dialog Box
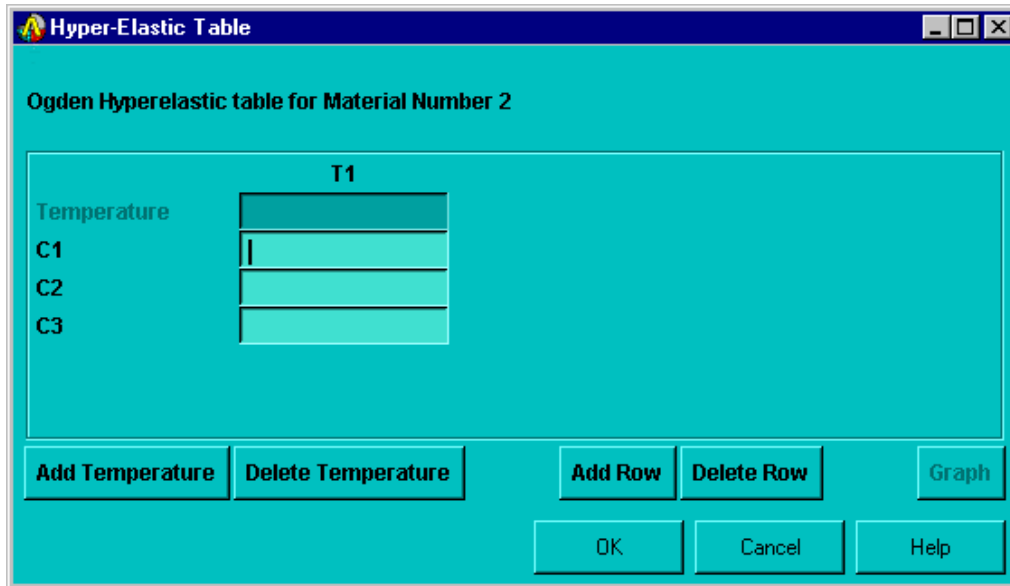
There are two interaction areas within a material data input dialog box: the data input table, and a series of action buttons that appear at the bottom. Depending on the material item you are defining, the labels in the table vary, as do the number of rows and columns that appear initially. The material item also dictates the number of rows and columns that you are allowed to add or delete. In most cases, the columns represent temperatures, and the rows represent data values (for example, density as a linear isotropic property, or constants for a particular non-linear model).

Temperature Dependent Data

Initially, the table is set up for temperature independent data so the temperature field is grayed out. At this point, should you decide to enter data for various temperatures, you can quickly add columns of text fields for the data representing each temperature. You can add or delete the temperature dependent data at any time. You do not need to *predetermine* if the data should be temperature dependent.

Adding and Deleting Columns

To add a column, position the text cursor in any field in the existing column, then click on the **Add Temperature** button. A new column appears to the right of the existing column, and both temperature fields become active, as shown in Figure 1.6: "Data Input Dialog Box - Added Column".

Figure 1.6 Data Input Dialog Box - Added Column
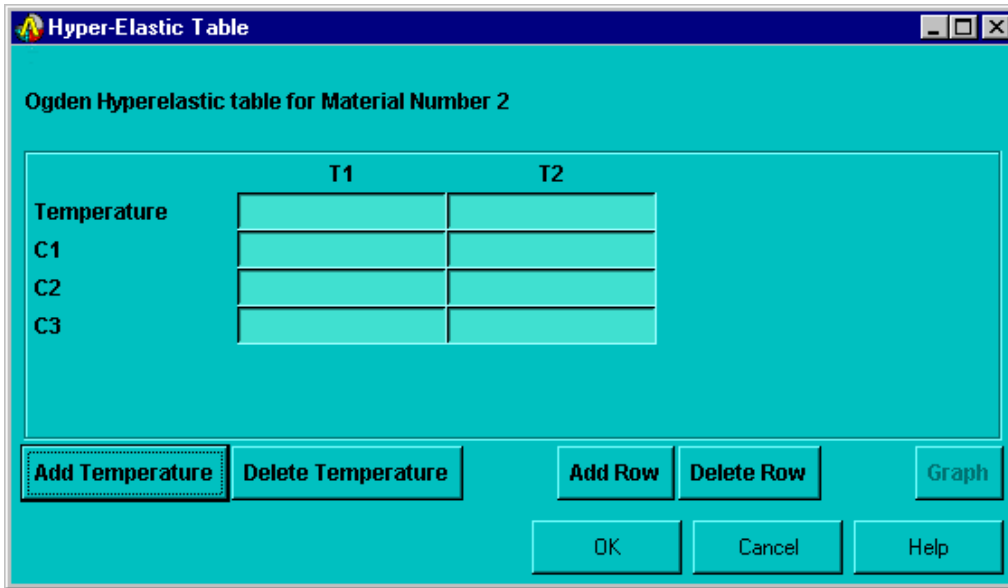
You then enter the two temperatures and the associated data in the rows. You can add more temperature columns, as needed, by following the same procedure. You can insert columns *between* existing columns by clicking the text cursor in a field within a column that is to the left of where you want to insert the new column, then clicking on the **Add Temperature** button. A scroll bar appears across the bottom of the table when the number of columns exceeds the width of the dialog box.

You can delete a temperature column by positioning the text cursor in any field within the column, and clicking on the **Delete Temperature** button.

Adding and Deleting Rows

You may have the need to add another row of constants or other data for a specific temperature. You add or delete rows in a similar way as is described above for adding or deleting columns. To add a row, click the text cursor in any field in an existing row, then click on the **Add Row** (or **Add Point**) button. A new row appears beneath the existing row, as shown in Figure 1.7: "Data Input Dialog Box - Added Row".

Figure 1.7 Data Input Dialog Box - Added Row
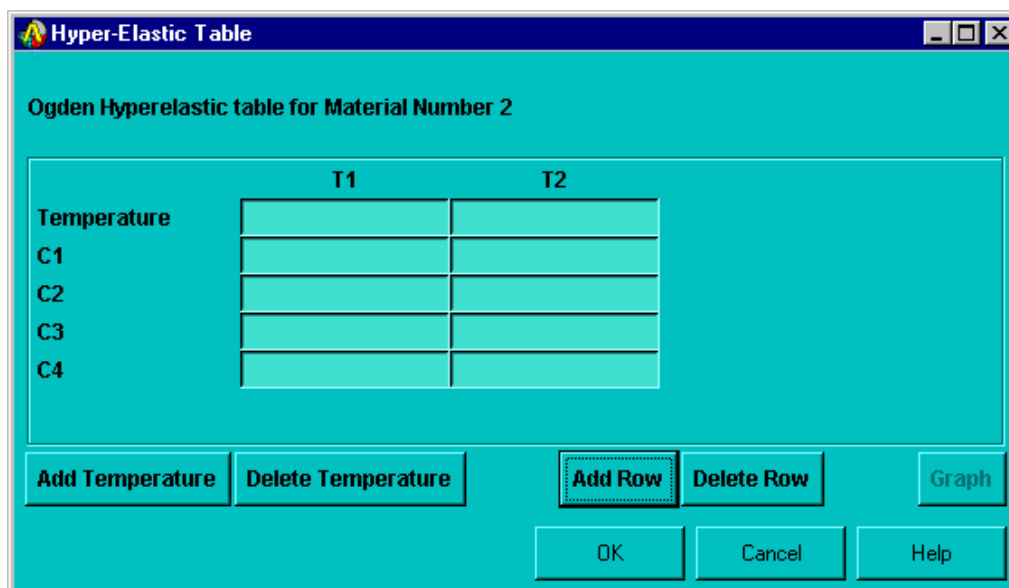
You can add more rows, as needed, by following the same procedure. You can insert rows *between* existing rows by positioning the text cursor in a field in the top row, then clicking on the **Add Row** (or **Add Point**) button. A vertical scroll bar appears in the table when the number of rows exceeds the height of the dialog box.

You can delete a row by positioning the text cursor in any field within the row, and clicking on the **Delete Row** (or **Delete Point**) button.

Entering/Editing Data in Text Fields

When a data dialog box first appears, one of the text fields is selected (black highlight), meaning that the field is ready to accept and display data as you type. You can use the arrow keys to move the selection status to other text fields. Also, pressing the Tab key allows you to move the selection status to the text field positioned to the right of the field that is currently selected.

When you start typing within a text field, the highlight is replaced by the characters that you type. You can use the left and right arrow keys to position the text cursor anywhere within the field should you need to replace or delete characters in that field.

To edit data, you must first select the text field either by clicking on the field, or using the arrow keys to move the selection status to the particular field.

To copy/paste data, select the text fields whose data you want to copy, use Ctrl-c to copy the data to the clipboard, select the empty destination text fields, then paste the data into these fields using Ctrl-v. You select multiple *adjacent* text fields either by dragging the mouse from the first field to the last field, or by clicking on the first field, holding down the Shift key, then clicking on the last field. For selecting multiple *nonadjacent* text fields, click on each field while you hold down the Ctrl key.

Action Buttons

  * **Add Temperature**: Adds a new column of data entry fields to the right of the column where the text cursor is currently positioned. If the button does not appear, the material item has no temperature dependency.
  * **Delete Temperature**: Deletes the column of data entry fields where the text cursor is currently positioned. If the button does not appear, the material item has no temperature dependency.

- **Add Row** (or **Add Point**): Adds a new row of data entry fields beneath the row where the text cursor is currently positioned. If the button does not appear, the material item has no provision for adding more data.
- **Delete Row** (or **Delete Point**): Deletes the row of data entry fields where the text cursor is currently positioned. If the button does not appear, the material item requires that all data entry fields must be completed.
- **Graph**: Displays a graph of the current data in the ANSYS Graphics window. If required, you can change the data in the table and click on the **Graph** button again before clicking on the **OK** button.
- **OK**: Commits all data that you have entered to the ANSYS database and removes this dialog box[1]. Material Model Number # appears in the **Material Models Defined** tree structure window, where # = 1 for the first model, or the number that you specified in the **Define Material ID** dialog box.
- **Cancel**: Cancels all data entered, and removes the dialog box[1].
- **Help**: Displays help information that is specific to the particular material property or material constant.

1.   Click on OK or Cancel to remove the data input dialog box. Pressing the Enter key will *not* remove the dialog box.

If a button appears, but is grayed out, then the function is defined for the particular material property, but you have not yet entered enough data for the function to become active.

Some material data input dialog boxes may include other buttons or interaction components that are necessary for completely defining a material property or model. See Section 4.2.1: A Dialog Box and Its Components in the *ANSYS Operations Guide* if you need help on the use of any of these interaction components.

Considerations for a Structural Analysis

When performing a structural analysis, several inelastic material models (listed by double-clicking on the following in the tree structure: Structural, Nonlinear, Inelastic) require you to input values for *elastic* material properties (elastic modulus and/or Poisson's ratio) in addition to the inelastic constants that are specific to the model (for example, Yield Stress and Tangent Modulus for the Bilinear Isotropic Hardening model). In these instances, you must enter the elastic material properties *before* you enter the inelastic constants. If you try to enter the inelastic constants first, a Note appears stating that you must first enter the elastic properties. After you click on OK in the Note, a data input dialog box appears that prompts you for the elastic material properties. After you enter these properties and click on OK, another data input dialog box appears that prompts you for the inelastic constants associated with the specific model you chose.

1.2.4.4.4. Logging/Editing Material Data

The **Material Models Defined** window (the left window in the **Define Material Model Behavior** dialog box) displays a log of each material model you have specified. After you have chosen OK in the data input dialog box, this window displays a folder icon, and **Material Model Number #** (the first # is 1 by default), followed by the properties defined for this model. You can define additional models with unique numbers by choosing **Material> New Model**, then typing a new number in the **Define Material ID** dialog box. If you double-click on any material model or property (furthest to the right in the tree), the associated data input dialog box appears where you can edit the data, if you choose.

1.2.4.4.5. Example: Defining a Single Material Model

This example and the following two examples show typical uses of the material model interface for use in structural analyses. If your specialty or interest is in performing analyses other than structural analyses, it is recommended that you still read and perform these examples to become familiar with maneuvering within the material model interface. You are then encouraged to try one of your own problems in your particular discipline, or try one of the many sample problems presented throughout the various ANSYS analysis guides. Here is a sampling of these problems:

- Section 2.8: Doing a Steady-State Thermal Analysis (GUI Method) in the *ANSYS Thermal Analysis Guide*.

- Section 6.6: Example of a 3-D Static Edge-Based Analysis (GUI Method) in the *ANSYS Low-Frequency Electromagnetic Analysis Guide*.

- Example problems in the *ANSYS High-Frequency Electromagnetic Analysis Guide*.

- Section 7.8: Sample Thermal-Structural Analysis (GUI Method) in the *ANSYS Coupled-Field Analysis Guide*.

The first example below is intended to show you how to completely define a single material model. It steps you through a procedure that uses the material model interface to define a model for simulating nonlinear isotropic hardening, using the Voce law, in a large strain structural analysis at two temperatures.

1. From the ANSYS Main Menu, click on the following menu path: **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. Enter values for material properties, as required (EX for elastic modulus, and PRXY for Poisson's ratio). Click on **OK**. **Material Model Number 1** properties appear listed in the **Material Models Defined** window.

4. In the **Material Models Available** window, double-click on the following options: **Nonlinear**, **Inelastic**, **Rate Independent**, **Isotropic Hardening Plasticity**, **von Mises Plasticity**, **Nonlinear**. A dialog box appears that includes a table where you can add temperature columns or add rows for material data, as needed for your application. Note that the temperature field is grayed out. This is because ANSYS assumes a temperature independent application, by default, so you would not need to enter a temperature value. Because this example is temperature *dependent* (involving two temperatures), you must first add another temperature column, as described in the next step.

5. Click on the **Add Temperature** button. A second column appears.

6. Enter the first temperature in the **Temperature** row and the **T1** column.

7. Enter the Voce constants required for the first temperature in the rows under the **T1** column (see *Nonlinear Isotropic Hardening* in the *ANSYS Elements Reference*).

8. Enter the second temperature in the **Temperature** row, and the **T2** column.

9. Enter the Voce constants required for the second temperature in the rows under the **T2** column.

   Note that if you needed to input constants for a third temperature, you would position the cursor in the **Temperature** row of the **T2** column, then click on the **Add Temperature** button again. This would cause a third column to appear.

   This material model only requires four constants per temperature. If you were using another model that allowed more constants, the **Add Row** button would be active. For those models, the same functionality is included for adding or inserting rows by using the **Add Row** (or **Add Point**) button.

10. Click on **OK**. The dialog box closes. The properties defined for that material are listed under **Material Model Number 1**.

1.2.4.4.6. Example: Editing Data in a Material Model

This example shows you how to use some of the basic editing features within the material model interface. It assumes that you have completed the previous example (see Section 1.2.4.4.5: Example: Defining a Single Material Model), and that the completed material model is listed in the **Material Models Defined** window.

Editing data typically falls into two general categories: changing data within an existing material property, and copying an entire material property set to form another model with slightly different properties.

Consider a case where you need to change the constants that you assigned to the Nonlinear Isotropic model. To perform this task:

1. Double-click on **Nonlinear Isotropic**. The associated dialog box appears with the existing data displayed in the fields.

2. Edit the constants in the appropriate fields, and click on **OK**.

    Note that if you needed to change any of the other material properties, you would double-click on **Linear Isotropic** in the previous step. This would cause the dialog box associated with linear isotropic properties to appear. You could then edit those properties.

Consider another case where you have the requirement for two material models, where the second model is the same as the first except that it needs to include constants for one more temperature. To perform this task:

1. In the **Define Material Model Behavior** dialog box, click on the following menu path: **Edit> Copy**, then choose 1 for **from Material number**, and enter 2 for **to Material number**. Click on **OK**. The **Material Models Defined** window now includes **Material Model Number 2** in its list. If you double-click on **Material Model Number 2**, the identical material properties appear below **Material Model Number 2** as those listed for **Material Model Number 1**.

2. Double-click on **Nonlinear Isotropic** under **Material Model Number 2**. The associated dialog box appears.

3. Move the text cursor to the **Temperature** row in the column furthest to the right, and click on the **Add Temperature** button. A **T3** column appears.

4. In the new column, enter the new temperature and the four constants associated with this temperature.

5. Click on **OK**. The dialog box closes. If you double-click on **Nonlinear Isotropic** under **Material Model Number 2**, the associated dialog box appears and reflects the new temperature data that you added for **Material Model Number 2**.

1.2.4.4.7. Example: Defining a Material Model Combination

This example is intended to show you how to define a material based on a combination of two material models. It steps you through a procedure that uses the material model interface to define a material for simulating cyclic softening at one temperature. This is accomplished by using the Nonlinear Isotropic model combined with the Chaboche model.

If you performed either of the previous examples in this section, start a new ANSYS session before beginning the following example.

1. From the ANSYS Main Menu, click on the following menu path: **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. Enter values for material properties, as required (EX for elastic modulus, and PRXY for Poisson's ratio). Click on **OK**. **Material Model Number 1** and **Linear Isotropic** appear in the **Material Models Defined** window.

4. In the **Material Models Available** window, double-click on the following options: **Nonlinear**, **Inelastic**, **Rate Independent**, **Combined Kinematic and Isotropic Hardening Plasticity**, **von Mises Plasticity**.

5. Double-click on **Chaboche and Nonlinear Isotropic**. A dialog box appears for defining the constants for the Chaboche model.

6. Enter the first three constants associated with the Chaboche model (click on the Help button for information on these constants).

7. The Chaboche model allows you to specify more constants. If you choose to specify more constants, click on the **Add Row** button, and enter the next constant.

8. Repeat the previous step for all the remaining Chaboche constants that you want to define.

9. Click on **OK**. The dialog box closes and another dialog box appears for defining the constants for the Nonlinear Isotropic model.

10. Enter the constants associated with the Nonlinear Isotropic model (click on the Help button for information on these constants).

11. Click on **OK**. The dialog box closes. Under **Material Model Number 1**, the following are listed: **Linear Isotropic**, **Chaboche**, and **Nonlinear Isotropic**. You can then edit any of the data (see Section 1.2.4.4.6: Example: Editing Data in a Material Model).

1.2.4.4.8. Material Model Interface - Miscellaneous Items

Other characteristics of the material model interface are the following:

- Any batch files you use to enter material data will be converted to material models and will appear listed in the **Material Models Defined** window of the **Define Material Model Behavior** dialog box.

- The material model interface does not import data from the ANSYS material library discussed in Section 1.2.4.5: Using Material Library Files.

- The material model interface does not support input through either **TB**,MOONEY, or the ***MOONEY** command.

1.2.4.5. Using Material Library Files

Although you can define material properties separately for each finite element analysis, ANSYS lets you store a material property set in an archival material library file, then retrieve the set and reuse it in multiple analyses. (Each material property set has its own library file.) The material library files also enable several ANSYS users to share commonly used material property data.

The material library feature offers you other advantages:

- Because the archived contents of material library files are reusable, you can use them to define other, similar material property sets quickly and with fewer errors. For example, suppose that you have defined material properties for one grade of steel and want to create a material property set for another grade of steel that is slightly different. You can write the existing steel material property set to a material library file, read it back into ANSYS under a different material number, and then, within ANSYS, make the minor changes needed to define properties for the second type of steel.

- Using the **/MPLIB** command (**Main Menu> Preprocessor> Material Props> Material Library> Library Path**), you can define a material library read and write path. Doing this allows you to protect your material data resources in a read-only archive, while giving ANSYS users the ability to write their material data locally without switching paths.

- You can give your material library files meaningful names that reflect the characteristics of the data they contain. For example, the name of a material library file describing properties of a steel casting might be **STEELCST.SI_MPL**. (See Section 1.2.4.8: Creating (Writing) a Material Library File for an explanation of file naming conventions.)

- You can design your own directory hierarchy for material library files. This enables you to classify and catalog the files by material type (plastic, aluminum, etc.), by units, or by any category you choose.

The next few paragraphs describe how to create and read material library files. For additional information, see the descriptions of the **/MPLIB**, **MPREAD**, and **MPWRITE** commands in the *ANSYS Elements Reference*.

1.2.4.6. Format of Material Library Files

Material library files are ANSYS command files. The file format supports both linear and nonlinear properties. You can reuse material library files because the commands in them are written so that, once you read a material property set into the ANSYS database, you can associate that set with any material number you wish.

1.2.4.7. Specifying a Default Read/Write Path for Material Library Files

Before you create any material library files, define a default read path and write path for those files:

> **Command(s): /MPLIB**,*R-W_opt*,*PATH*
> **GUI: Main Menu> Preprocessor> Material Props> Material Library> Library Path**

Note — The ANSYS-supplied material library is located at **/ansys81/matlib/**.

In place of *R-W_opt*, specify READ (to set the read path), WRITE (to set the write path), or STAT to see what read and write paths currently are in use. In place of *PATH*, specify the path to be used for material library files.

1.2.4.8. Creating (Writing) a Material Library File

To create an archival material library file, perform these steps:

1. To tell the ANSYS program what system of units you are using, issue the **/UNITS** command. For example, to specify the international system of units, you would issue the command **/UNITS**,SI. You cannot access the **/UNITS** command directly from the GUI.

2. Define a material property using the **MP** command (**Main Menu> Preprocessor> Material Props> Isotropic**). To do so, you must specify a material number and at least one material property value (for example, magnetic permeability or MURX).

3. From the PREP7 preprocessor, issue the command shown below:

```
MPWRITE,Filename,,,LIB,MAT
```

Filename is the name to assign to the material library file. Issue **MPWRITE** (**Main Menu> Preprocessor> Material Props> Material Library> Export Library**) and specify the filename for the material library file.

Issuing **MPWRITE** writes the material data specified by material number MAT into the named file in the current working directory. (If you previously specified a material library write path by issuing the **/MPLIB** command (**Main Menu> Preprocessor> Material Props> Material Library> Library Path**), ANSYS writes the file to that location instead.)

Naming conventions for a material library file are as follows:

*   The name of the file is the name you specify on the **MPWRITE** command. If you do not specify a filename, the default name is **JOBNAME**.

*   The extension of a material library filename follows the pattern **.*xxx*_MPL**, where *xxx* identifies the system of units for this material property sets. For example, if the system of units is the CGS system, the file extension is **.CGS_MPL**. The default extension, used if you do not specify a units system before creating the material library file, is **.USER_MPL**. (This indicates a user-defined system of units.)

ANSYS Basic Analysis Procedures Guide . ANSYS Release 8.1 . 001972 . © SAS IP, Inc.

1.2.4.9. Reading a Material Library File

To read a material library file into the ANSYS database, perform these steps:

1. Use the **/UNITS** command or its GUI equivalent to tell the ANSYS program what system of units you are using.

    Note — The default system of units for ANSYS is SI. The GUI lists only material library files with the currently active units.

2. Specify a new material reference number or an existing number that you wish to overwrite:
    Command(s): MAT
    GUI: Main Menu> Preprocessor> Modeling> Create> Elements> Elem Attributes

    Caution: Overwriting an existing material in the ANSYS database deletes all of the data associated with it.

3. To read the material library file into the database, use one of the following:
    Command(s): MPREAD,*Filename*,,,*LIB*
    GUI: Main Menu> Preprocessor> Material Props> Material Library> Import Library

The *LIB* argument supports a file search hierarchy. The program searches for the named material library file first in the current working directory, then in your home directory, then in the read path directory specified by the **/MPLIB** command, and finally in the ANSYS-supplied directory **/ansys81/matlib**. If you omit the *LIB* argument, the programs searches only in the current working directory.

1.2.5. Creating the Model Geometry

Once you have defined material properties, the next step in an analysis is generating a finite element model - nodes and elements - that adequately describes the model geometry. The graphic below shows some sample finite element models:

Figure 1.8 Sample Finite Element Models
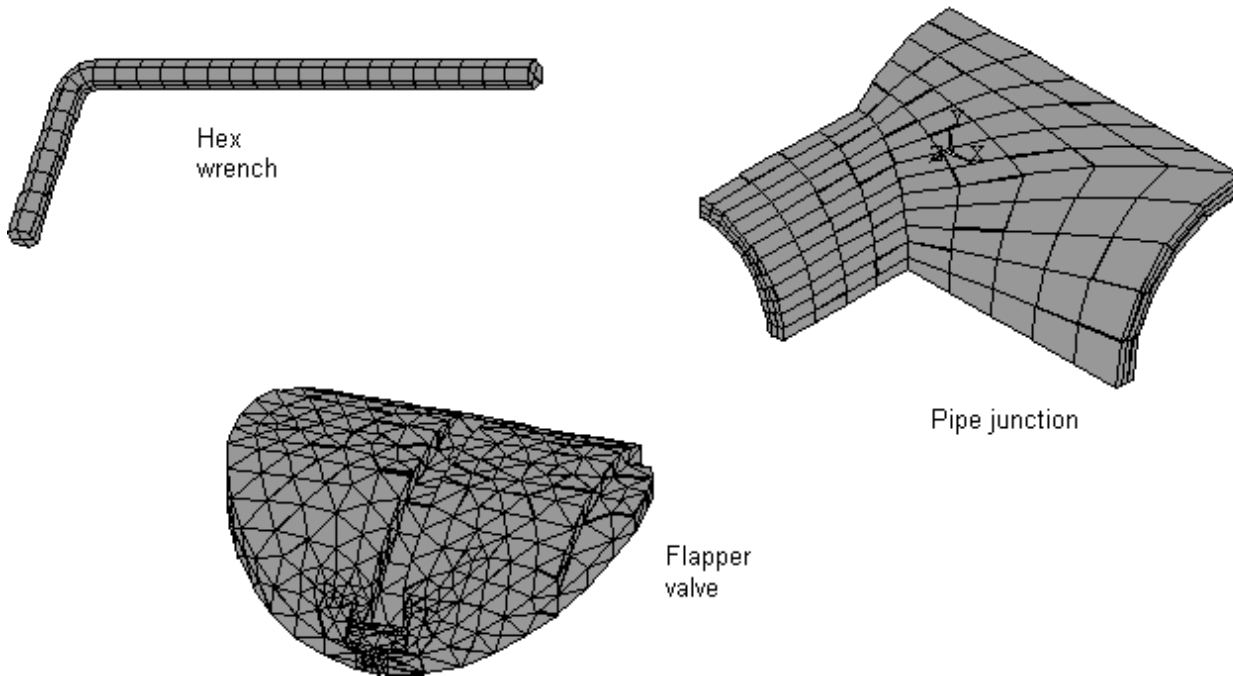
There are two methods to create the finite element model: solid modeling and direct generation. With *solid modeling*, you describe the geometric shape of your model, then instruct the ANSYS program to automatically *mesh* the geometry with nodes and elements. You can control the size and shape in the elements that the program creates. With *direct generation*, you "manually" define the location of each node and the connectivity of each element. Several convenience operations, such as copying patterns of existing nodes and elements, symmetry reflection, etc. are available.

Details of the two methods and many other aspects related to model generation - coordinate systems, working planes, coupling, constraint equations, etc. - are described in the *ANSYS Modeling and Meshing Guide*.

1.2.6. Apply Loads and Obtain the Solution

In this step, you use the SOLUTION processor to define the analysis type and analysis options, apply loads, specify load step options, and initiate the finite element solution. You also can apply loads using the PREP7 preprocessor.

1.2.6.1. Defining the Analysis Type and Analysis Options

You choose the analysis type based on the loading conditions and the response you wish to calculate. For example, if natural frequencies and mode shapes are to be calculated, you would choose a modal analysis. You can perform the following analysis types in the ANSYS program: static (or steady-state), transient, harmonic, modal, spectrum, buckling, and substructuring.

Not all analysis types are valid for all disciplines. Modal analysis, for example, is not valid for a thermal model. The analysis guide manuals in the ANSYS documentation set describe the analysis types available for each discipline and the procedures to do those analyses.

Analysis options allow you to customize the analysis type. Typical analysis options are the method of solution, stress stiffening on or off, and Newton-Raphson options.

To define the analysis type and analysis options, use the **ANTYPE** command (**Main Menu> Preprocessor> Loads> Analysis Type> New Analysis** or **Main Menu> Preprocessor> Loads> Analysis Type> Restart**) and

the appropriate analysis option commands (**TRNOPT**, **HROPT**, **MODOPT**, **SSTIF**, **NROPT**, etc.). For GUI equivalents for the other commands, see their descriptions in the *ANSYS Elements Reference*.

If you are performing a static or full transient analysis, you can take advantage of the **Solution Controls** dialog box to define many options for the analysis. For details about the **Solution Controls** dialog box, see Chapter 3, "Solution".

You can specify either a new analysis or a restart, but a new analysis is the choice in most cases. A singleframe restart that allows you to resume a job at its end point or abort point is available for static (steady-state), harmonic (2-D magnetic only), and transient analyses. A multiframe restart that allows you to restart an analysis at any point is available for static or full transient structural analyses. See Section 3.16: Restarting an Analysis for complete information on performing restarts. The various analysis guides discuss additional details necessary for restarts. You cannot change the analysis type and analysis options after the first solution.

A sample input listing for a structural transient analysis is shown below. Remember that the discipline (structural, thermal, magnetic, etc.) is implied by the *element types* used in the model.

```
ANTYPE,TRANS
TRNOPT,FULL
NLGEOM,ON
```

Once you have defined the analysis type and analysis options, the next step is to apply loads. Some structural analysis types require other items to be defined first, such as master degrees of freedom and gap conditions. The *ANSYS Structural Analysis Guide* describes these items where necessary.

1.2.6.2. Applying Loads

The word *loads* as used in ANSYS documentation includes boundary conditions (constraints, supports, or boundary field specifications) as well as other externally and internally applied loads. Loads in the ANSYS program are divided into six categories:

- DOF Constraints
- Forces
- Surface Loads
- Body Loads
- Inertia Loads
- Coupled-field Loads

You can apply most of these loads either on the solid model (keypoints, lines, and areas) or the finite element model (nodes and elements). For details about the load categories and how they can be applied on your model, see Chapter 2, "Loading" in this manual.

Two important load-related terms you need to know are load step and substep. A *load step* is simply a configuration of loads for which you obtain a solution. In a structural analysis, for example, you may apply wind loads in one load step and gravity in a second load step. Load steps are also useful in dividing a transient load history curve into several segments.

Substeps are incremental steps taken within a load step. You use them mainly for accuracy and convergence purposes in transient and nonlinear analyses. Substeps are also known as *time steps* - steps taken over a period of time.

Note — The ANSYS program uses the concept of *time* in transient analyses as well as static (or steady-state) analyses. In a transient analysis, time represents actual time, in seconds, minutes, or hours. In a static or steady-state analysis, time simply acts as a counter to identify load steps and substeps.

1.2.6.3. Specifying Load Step Options

Load step options are options that you can change from load step to load step, such as number of substeps, time at the end of a load step, and output controls. Depending on the type of analysis you are doing, load step options may or may not be required. The analysis procedures in the analysis guide manuals describe the appropriate load step options as necessary. See Chapter 2, "Loading" for a general description of load step options.

1.2.6.4. Initiating the Solution

To initiate solution calculations, use either of the following:
>**Command(s): SOLVE**
>**GUI: Main Menu> Solution> Solve> Current LS**
>**Main Menu> Solution>** *solution_method*

When you issue this command, the ANSYS program takes model and loading information from the database and calculates the results. Results are written to the results file (**Jobname.RST**, **Jobname.RTH**, **Jobname.RMG**, or **Jobname.RFL**) and also to the database. The only difference is that only one set of results can reside in the database at one time, while you can write all sets of results (for all substeps) to the results file.

You can solve multiple load steps in a convenient manner:
>**Command(s): LSSOLVE**
>**GUI: Main Menu> Solution> Solve> From LS Files**

Chapter 3, "Solution" discusses this and other solution-related topics.

1.2.7. Review the Results

Once the solution has been calculated, you can use the ANSYS postprocessors to review the results. Two postprocessors are available: POST1 and POST26.

You use POST1, the general postprocessor, to review results at one substep (time step) over the entire model or selected portion of the model. The command to enter POST1 is **/POST1** (**Main Menu> General Postproc**), valid only at the Begin level. You can obtain contour displays, deformed shapes, and tabular listings to review and interpret the results of the analysis. POST1 offers many other capabilities, including error estimation, load case combinations, calculations among results data, and path operations.

You use POST26, the time history postprocessor, to review results at specific points in the model over all time steps. The command to enter POST26 is **/POST26** (**Main Menu> TimeHist Postpro**), valid only at the Begin level. You can obtain graph plots of results data versus time (or frequency) and tabular listings. Other POST26 capabilities include arithmetic calculations and complex algebra. Details of POST1 and POST26 capabilities and how to use them are described in chapters later in this document.

Chapter 2: Loading

2.1. Loading Overview

The main goal of a finite element analysis is to examine how a structure or component responds to certain loading conditions. Specifying the proper loading conditions is, therefore, a key step in the analysis. You can apply loads on the model in a variety of ways in the ANSYS program. Also, with the help of load step options, you can control how the loads are actually used during solution.

2.2. What Are Loads?

The word *loads* in ANSYS terminology includes boundary conditions *and* externally or internally applied forcing functions, as illustrated in Figure 2.1: "Loads". Examples of loads in different disciplines are:

Structural: displacements, forces, pressures, temperatures (for thermal strain), gravity

Thermal: temperatures, heat flow rates, convections, internal heat generation, infinite surface

Magnetic: magnetic potentials, magnetic flux, magnetic current segments, source current density, infinite surface

Electric: electric potentials (voltage), electric current, electric charges, charge densities, infinite surface

Fluid: velocities, pressures

Figure 2.1 Loads

Includes boundary conditions as well as other types of loading
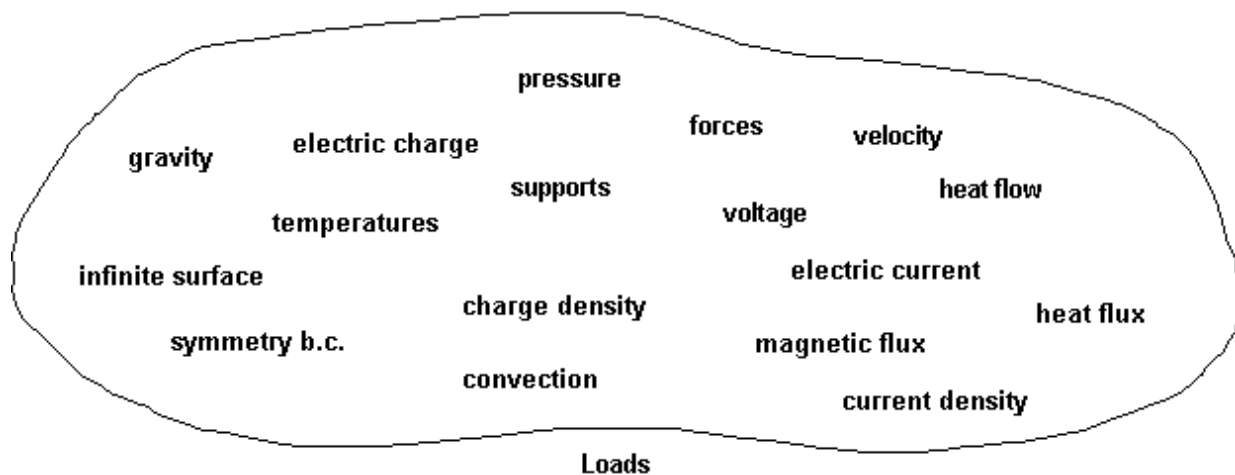
Loads are divided into six categories: DOF constraints, forces (concentrated loads), surface loads, body loads, inertia loads, and coupled-field loads.

- A *DOF constraint* fixes a degree of freedom (DOF) to a known value. Examples of constraints are specified displacements and symmetry boundary conditions in a structural analysis, prescribed temperatures in a thermal analysis, and flux-parallel boundary conditions.

- A *force* is a concentrated load applied at a node in the model. Examples are forces and moments in a structural analysis, heat flow rates in a thermal analysis, and current segments in a magnetic field analysis.

- A *surface load* is a distributed load applied over a surface. Examples are pressures in a structural analysis and convections and heat fluxes in a thermal analysis.

- A *body load* is a volumetric or field load. Examples are temperatures and fluences in a structural analysis, heat generation rates in a thermal analysis, and current densities in a magnetic field analysis.

- *Inertia loads* are those attributable to the inertia (mass matrix) of a body, such as gravitational acceleration, angular velocity, and angular acceleration. You use them mainly in a structural analysis.

- *Coupled-field loads* are simply a special case of one of the above loads, where results from one analysis are used as loads in another analysis. For example, you can apply magnetic forces calculated in a magnetic field analysis as force loads in a structural analysis.

Definitions of other loads-related terms appear below.

2.3. Load Steps, Substeps, and Equilibrium Iterations

A *load step* is simply a configuration of loads for which a solution is obtained. In a linear static or steady-state analysis, you can use different load steps to apply different sets of loads - wind load in the first load step, gravity load in the second load step, both loads and a different support condition in the third load step, and so on. In a transient analysis, multiple load steps apply different segments of the load history curve.

The ANSYS program uses the set of elements which you select for the first load step for all subsequent load steps, no matter which element sets you specify for the later steps. To select an element set, you use either of the following:

Command(s): ESEL
GUI: Utility Menu> Select> Entities

Figure 2.2: "Transient Load History Curve" shows a load history curve that requires three load steps - the first load step for the ramped load, the second load step for the constant portion of the load, and the third load step for load removal.

Figure 2.2 Transient Load History Curve
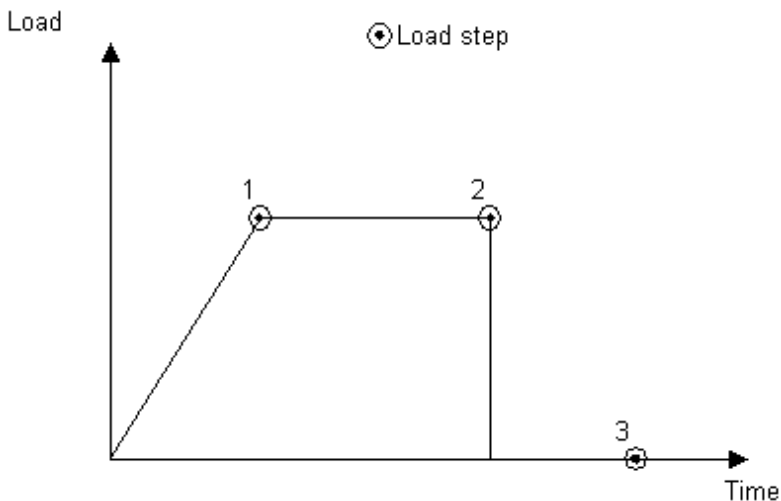
Using multiple load steps to specify

Substeps are points within a load step at which solutions are calculated. You use them for different reasons:

- In a nonlinear static or steady-state analysis, use substeps to apply the loads gradually so that an accurate solution can be obtained.

- In a linear or nonlinear transient analysis, use substeps to satisfy transient time integration rules (which usually dictate a minimum integration time step for an accurate solution).

- In a harmonic response analysis, use substeps to obtain solutions at several frequencies within the harmonic frequency range.

Equilibrium iterations are additional solutions calculated at a given substep for convergence purposes. They are iterative corrections used only in nonlinear analyses (static or transient), where convergence plays an important role.

Consider, for example, a 2-D, nonlinear static magnetic analysis. To obtain an accurate solution, two load steps are commonly used. (Figure 2.3: "Load Steps, Substeps, and Equilibrium Iterations" illustrates this.)

- The first load step applies the loads gradually over five to 10 substeps, each with just one equilibrium iteration.

- The second load step obtains a final, converged solution with just one substep that uses 15 to 25 equilibrium iterations.

Figure 2.3 Load Steps, Substeps, and Equilibrium Iterations
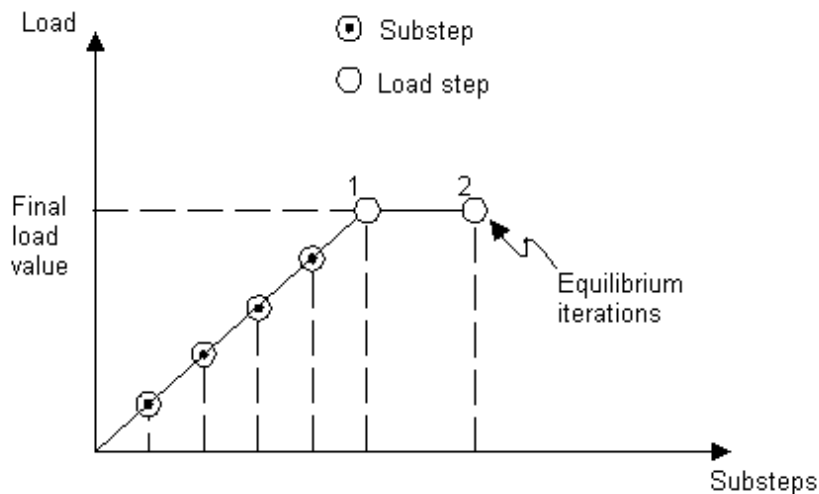
2.4. The Role of Time in Tracking

The ANSYS program uses time as a tracking parameter in *all* static and transient analyses, whether they are or are not truly time-dependent. The advantage of this is that you can use one consistent "counter" or "tracker" in all cases, eliminating the need for analysis-dependent terminology. Moreover, time always increases monotonically, and most things in nature happen over a period of time, however brief the period may be.

Obviously, in a transient analysis or in a rate-dependent static analysis (creep or viscoplasticity), *time* represents actual, chronological time in seconds, minutes, or hours. You assign the time at the end of each load step (using the **TIME** command) while specifying the load history curve. To assign time, use one of the following:

      Command(s): TIME
      GUI: Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Time and Substps
      Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Time - Time Step
      Main Menu> Solution> Analysis Type> Sol'n Control (: Basic Tab)
      Main Menu> Solution> Load Step Opts> Time/Frequenc> Time and Substps

> **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time - Time Step**
> **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time and Substps**
> **Main Menu> Solution> Load Step Opts> Time /Frequenc> Time - Time Step**

In a rate-independent analysis, however, *time* simply becomes a counter that identifies load steps and substeps. By default, the program automatically assigns time = 1.0 at the end of load step 1, time = 2.0 at the end of load step 2, and so on. Any substeps within a load step will be assigned the appropriate, linearly interpolated time value. By assigning your own time values in such analyses, you can establish your own tracking parameter. For example, if a load of 100 units is to be applied incrementally over one load step, you can specify time at the end of that load step to be 100, so that the load and time values are synchronous.

In the postprocessor, then, if you obtain a graph of deflection versus time, it means the same as deflection versus load. This technique is useful, for instance, in a large-deflection buckling analysis where the objective may be to track the deflection of the structure as it is incrementally loaded.

Time takes on yet another meaning when you use the arc-length method in your solution. In this case, *time* equals the value of time at the beginning of a load step, plus the value of the arc-length load factor (the multiplier on the currently applied loads). ALLF does not have to be monotonically increasing (that is, it can increase, decrease, or even become negative), and it is reset to zero at the beginning of each load step. As a result, *time* is not considered a "counter" in arc-length solutions.

The arc-length method is an advanced solution technique. For more information about using it, see Chapter 8, "Nonlinear Structural Analysis" in the *ANSYS Structural Analysis Guide*.

A load step is a set of loads applied over a given *time span*. Substeps are *time points* within a load step at which intermediate solutions are calculated. The difference in time between two successive substeps can be called a *time step* or *time increment*. Equilibrium iterations are iterative solutions calculated at a given time point purely for convergence purposes.

2.5. Stepped Versus Ramped Loads

When you specify more than one substep in a load step, the question of whether the loads should be *stepped* or *ramped* arises.

- If a load is *stepped*, then its full value is applied at the first substep and stays constant for the rest of the load step.

- If a load is *ramped*, then its value increases gradually at each substep, with the full value occurring at the end of the load step.

Figure 2.4 Stepped Versus Ramped Loads

(a) Stepped loads   (b) Ramped loads
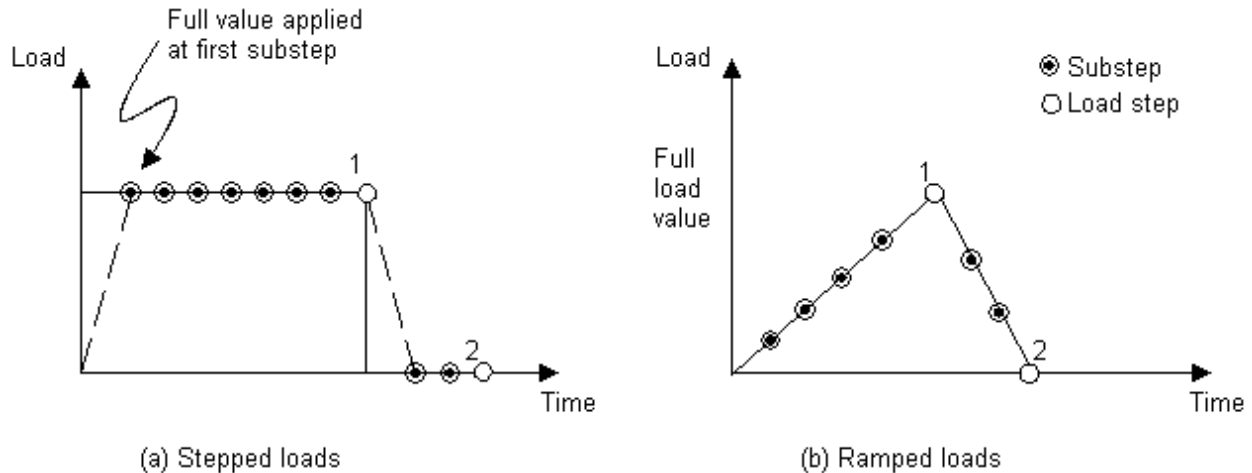
The **KBC** command (**Main Menu> Solution> Analysis Type> Sol'n Control**: **Transient Tab**, **Main Menu> Solution> Load Step Opts> Time/Frequenc> Freq & Substeps** / **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time and Substps** / **Main Menu> Solution> Load Step Opts > Time/Frequenc> Time & Time Step**, or **Main Menu> Solution> Load Step Opts> Time/Frequenc> Freq & Substeps** / **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time and Substps** / **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time & Time Step**) is used to indicate whether loads are ramped or stepped. **KBC**,0 indicates ramped loads, and **KBC**,1 indicates stepped loads. The default depends on the discipline and type of analysis.

Load step options is a collective name given to options that control load application, such as time, number of substeps, the time step, and stepping or ramping of loads. Other types of load step options include convergence tolerances (used in nonlinear analyses), damping specifications in a structural analysis, and output controls.

2.6. How to Apply Loads

You can apply most loads either on the solid model (on keypoints, lines, and areas) or on the finite element model (on nodes and elements). For example, you can specify forces at a keypoint or a node. Similarly, you can specify convections (and other surface loads) on lines and areas or on nodes and element faces. No matter how you specify the loads, the solver expects all loads to be in terms of the finite element model. Therefore, if you specify loads on the solid model, the program automatically transfers them to the nodes and elements at the beginning of solution.

2.6.1. Solid-Model Loads: Advantages and Disadvantages

Advantages

- Solid-model loads are independent of the finite element mesh. That is, you can change the element mesh without affecting the applied loads. This allows you to make mesh modifications and conduct mesh sensitivity studies without having to reapply loads each time.

- The solid model usually involves fewer entities than the finite element model. Therefore, selecting solid model entities and applying loads on them is much easier, especially with graphical picking.

Disadvantages

- Elements generated by ANSYS meshing commands are in the currently active element coordinate system. Nodes generated by meshing commands use the global Cartesian coordinate system. Therefore, the solid model and the finite element model may have different coordinate systems and loading directions.

- Solid-model loads are not very convenient in reduced analyses, where loads are applied at master degrees of freedom. (You can define master DOF only at nodes, not at keypoints.)

- Applying keypoint constraints can be tricky, especially when the constraint expansion option is used. (The expansion option allows you to expand a constraint specification to all nodes between two keypoints that are connected by a line.)

- You cannot display all solid-model loads.

Notes About Solid-Model Loads

As mentioned earlier, solid-model loads are automatically transferred to the finite element model at the beginning of solution. If you mix solid model loads with finite-element model loads, couplings, or constraint equations, you should be aware of the following possible conflicts:

- Transferred solid loads will replace nodal or element loads already present, regardless of the order in which the loads were input. for example, **DL**,,,UX on a line will overwrite any **D**,,UX's on the nodes of that line at transfer time.

- Deleting solid model loads also deletes any corresponding finite element loads. For example, **SFADELE**,,,$PRES$ on an area will immediately delete any **SFE**,,,$PRES$'s on the elements in that area.

- Line or area symmetry or antisymmetry conditions (**DL**,,,$SYMM$, **DL**,,,$ASYM$, **DA**,,,$SYMM$, or **DA**,,,$ASYM$) often introduce nodal rotations that could effect nodal constraints, nodal forces, couplings, or constraint equations on nodes belonging to constrained lines or areas.

2.6.2. Finite-Element Loads: Advantages and Disadvantages

Advantages

- Reduced analyses present no problems, because you can apply loads directly at master nodes.

- There is no need to worry about constraint expansion. You can simply select all desired nodes and specify the appropriate constraints.

Disadvantages

- Any modification of the finite element mesh invalidates the loads, requiring you to delete the previous loads and re-apply them on the new mesh.

- Applying loads by graphical picking is inconvenient, unless only a few nodes or elements are involved.

The next few subsections discuss how to apply each category of loads - constraints, forces, surface loads, body loads, inertia loads, and coupled-field loads - and then explain how to specify load step options.

2.6.3. DOF Constraints

Table 2.1: "DOF Constraints Available in Each Discipline" shows the degrees of freedom that can be constrained in each discipline and the corresponding ANSYS labels. Any directions implied by the labels (such as UX, ROTZ, AY, etc.) are in the nodal coordinate system. For a description of different coordinate systems, see the *ANSYS Modeling and Meshing Guide*.

Table 2.2: "Commands for DOF Constraints" shows the commands to apply, list, and delete DOF constraints. Notice that you can apply constraints on nodes, keypoints, lines, and areas.

Table 2.1 DOF Constraints Available in Each Discipline

Discipline	Degree of Freedom	ANSYS Label
Structural	Translations Rotations	UX, UY, UZ ROTX, ROTY, ROTZ
Thermal	Temperature	TEMP, TBOT, TE2, . . . TTOP
Magnetic	Vector Potentials Scalar Potential	AX, AY, AZ MAG
Electric	Voltage	VOLT
Fluid	Velocities Pressure Turbulent Kinetic Energy Turbulent Dissipation Rate	VX, VY, VZ PRES ENKE ENDS

Table 2.2 Commands for DOF Constraints

Location	Basic Commands	Additional Commands
Nodes	**D**, **DLIST**, **DDELE**	**DSYM**, **DSCALE**, **DCUM**
Keypoints	**DK**, **DKLIST**, **DKDELE**	-
Lines	**DL**, **DLLIST**, **DLDELE**	-
Areas	**DA**, **DALIST**, **DADELE**	-
Transfer	**SBCTRAN**	**DTRAN**

Below are examples of some of the GUI paths you can use to apply DOF constraints:

GUI:

> **Main Menu> Preprocessor> Loads> Define Loads> Apply>** *load type*> **On Nodes**
> **Utility Menu> List> Loads> DOF Constraints> On All Keypoints (or On Picked KPs)**
> **Main Menu> Solution> Define Loads> Apply>** *load type*> **On Lines**

See the *ANSYS Commands Reference* for additional GUI path information and for descriptions of the commands listed in Table 2.2: "Commands for DOF Constraints".

2.6.4. Applying Symmetry or Antisymmetry Boundary Conditions

Use the **DSYM** command to apply symmetry or antisymmetry boundary conditions on a plane of nodes. The command generates the appropriate DOF constraints. See the *ANSYS Commands Reference* for the list of constraints generated.

In a structural analysis, for example, a symmetry boundary condition means that out-of-plane translations and in-plane rotations are set to zero, and an antisymmetry condition means that in-plane translations and out-of-plane rotations are set to zero. (See Figure 2.5: "Symmetry and Antisymmetry Boundary Conditions".) All nodes on the symmetry plane are rotated into the coordinate system specified by the *KCN* field on the **DSYM** command. The use of symmetry and antisymmetry boundary conditions is illustrated in Figure 2.6: "Examples of Boundary Conditions". The **DL** and **DA** commands work in a similar fashion when you apply symmetry or antisymmetry conditions on lines and areas.

You can use the **DL** and **DA** commands to apply velocities, pressures, temperatures, and turbulence quantities on lines and areas for FLOTRAN analyses. At your discretion, you can apply boundary conditions at the endpoints of the lines and the edges of areas.

Note — If the node rotation angles that are in the database while you are using the general postprocessor (POST1) are different from those used in the solution being postprocessed, POST1 may display incorrect results. This condition usually results if you introduce node rotations in a second or later load step by applying symmetry or antisymmetry boundary conditions. Erroneous cases display the following message in POST1 when you execute the **SET** command (**Utility Menu> List> Results> Load Step Summary**):

```
*** WARNING ***
Cumulative iteration 1 may have been solved using
different model or boundary condition data than is
currently stored.  POST1 results may be erroneous
unless you resume from a .db file matching this solution.
```

Figure 2.5 Symmetry and Antisymmetry Boundary Conditions
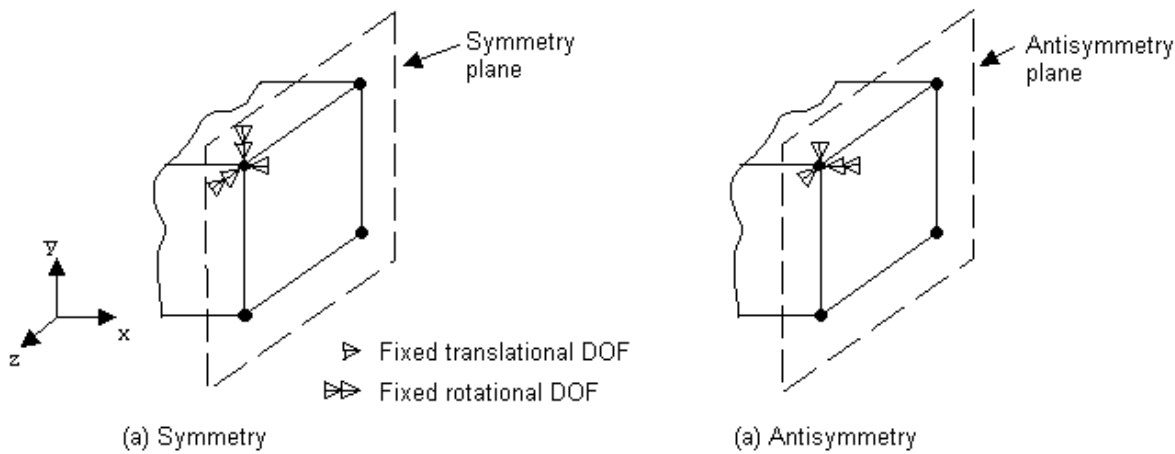
Figure 2.6 Examples of Boundary Conditions
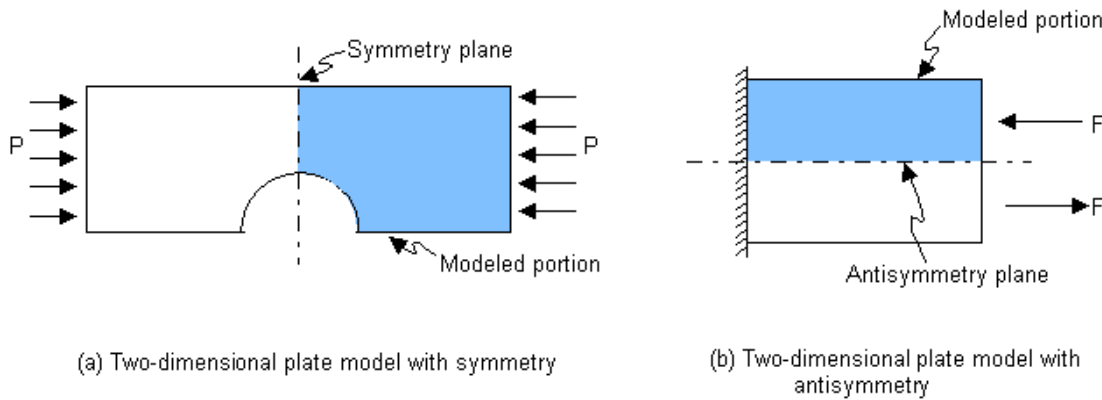
2.6.5. Transferring Constraints

To transfer constraints that have been applied to the solid model to the corresponding finite element model, use one of the following:

> **Command(s): DTRAN**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Transfer to FE> Constraints**
> **Main Menu> Solution> Define Loads> Operate> Transfer to FE> Constraints**

To transfer all solid model boundary conditions, use one of the following:

Command(s): SBCTRAN
GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Transfer to FE> All Solid Lds
Main Menu> Solution> Define Loads> Operate> Transfer to FE> All Solid Lds

2.6.5.1. Resetting Constraints

By default, if you repeat a DOF constraint on the same degree of freedom, the new specification *replaces* the previous one. You can change this default to *add* (for accumulation) or *ignore* with the **DCUM** command (**Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs. Add> Constraints**). For example:

```
NSEL,...            ! Selects a set of nodes
D,ALL,VX,40         ! Sets VX = 40 at all selected nodes
D,ALL,VX,50         ! Changes VX value to 50 (replacement)
DCUM,ADD            ! Subsequent D's to be added
D,ALL,VX,25         ! VX = 50+25 = 75 at all selected nodes
DCUM,IGNORE         ! Subsequent D's to be ignored
D,ALL,VX,1325       ! These VX values are ignored!
DCUM                ! Resets DCUM to default (replacement)
```

See the *ANSYS Commands Reference* for discussions of the **NSEL,D**, and **DCUM** commands.

Any DOF constraints you set with **DCUM** stay set until another **DCUM** is issued. To reset the default setting (replacement), simply issue **DCUM** without any arguments.

2.6.5.2. Scaling Constraint Values

You can scale existing DOF constraint values as follows:

Command(s): DSCALE
GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Scale FE Loads> Constraints
Main Menu> Solution> Define Loads> Operate> Scale FE Loads> Constraints

Both the **DSCALE** and **DCUM** commands work on all selected nodes and also *on all selected DOF labels*. By default, DOF labels that are active are those associated with the element types in the model:

Command(s): DOFSEL
GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Scale FE Loads> Constraints (or Forces)
Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs. Add> Constraints (or Forces)
Main Menu> Solution> Define Loads> Operate> Scale FE Loads> Constraints (or Forces)
Main Menu> Solution> Define Loads> Settings> Replace vs. Add> Constraints (or Forces)

For example, if you want to scale only VX values and not any other DOF label, you can use the following commands:

```
DOFSEL,S,VX         ! Selects VX label
DSCALE,0.5          ! Scales VX at all selected nodes by 0.5
DOFSEL,ALL          ! Reactivates all DOF labels
```

When scaling temperature constraints (TEMP) in a thermal analysis, you can use the *TBASE* field on the **DSCALE** command to scale the temperature offset from a base temperature (that is, to scale |TEMP-*TBASE*|) rather than the actual temperature values. The following figure illustrates this.

Figure 2.7 Scaling Temperature Constraints with DSCALE
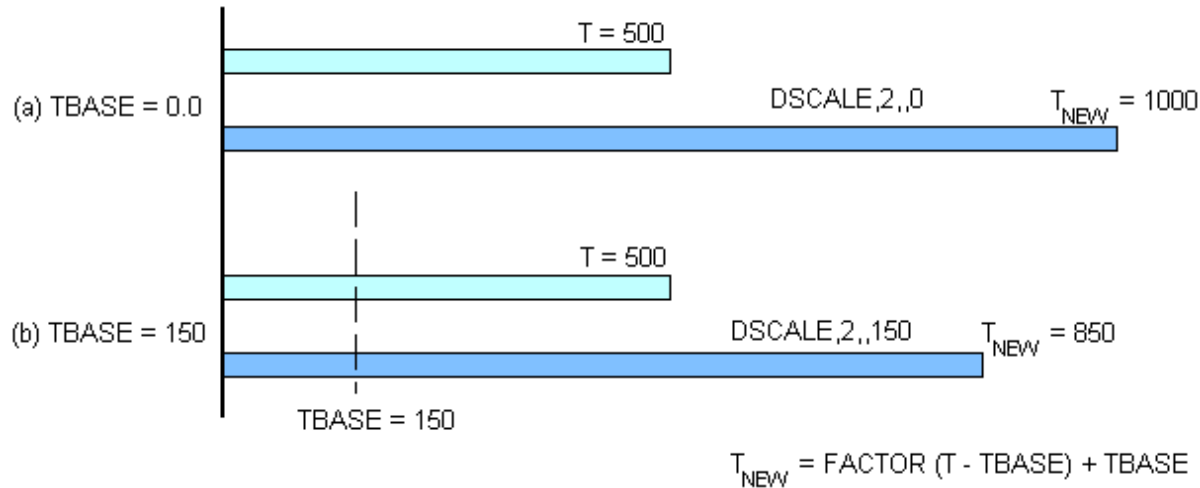
2.6.5.3. Resolution of Conflicting Constraint Specifications

You need to be aware of the possibility of conflicting **DK**, **DL**, and **DA** constraint specifications and how the ANSYS program handles them. The following conflicts can arise:

- A **DL** specification can conflict with a **DL** specification on an adjacent line (shared keypoint).
- A **DL** specification can conflict with a **DK** specification at either keypoint.
- A **DA** specification can conflict with a **DA** specification on an adjacent area (shared lines/keypoints).
- A **DA** specification can conflict with a **DL** specification on any of its lines.
- A **DA** specification can conflict with a **DK** specification on any of its keypoints.

The ANSYS program transfers constraints that have been applied to the solid model to the corresponding finite element model in the following sequence:

1. In ascending area number order, DOF **DA** constraints transfer to nodes on areas (and bounding lines and keypoints).
2. In ascending area number order, SYMM and ASYM **DA** constraints transfer to nodes on areas (and bounding lines and keypoints).
3. In ascending line number order, DOF **DL** constraints transfer to nodes on lines (and bounding keypoints).
4. In ascending line number order, SYMM and ASYM **DL** constraints transfer to nodes on lines (and bounding keypoints).
5. **DK** constraints transfer to nodes on keypoints (and on attached lines, areas, and volumes if expansion conditions are met).

Accordingly, for conflicting constraints, **DK** commands overwrite **DL** commands and **DL** commands overwrite **DA** commands. For conflicting constraints, constraints specified for a higher line number or area number overwrite the constraints specified for a lower line number or area number, respectively. The constraint specification issue order does not matter.

Note — Any conflict detected during solid model constraint transfer produces a warning similar to the following:

```
*** WARNING ***
DOF constraint ROTZ from line 8 (1st value=22) is overwriting a D on
node 18 (1st value=0) that was previously transferred from another
DA, DL, or set of DK's.
```

Changing the value of **DK**, **DL**, or **DA** constraints between solutions may produce many of these warnings at the 2nd or later solid BC transfer. These can be prevented if you delete the nodal **D** constraints between solutions using **DADELE**, **DLDELE**, and/or **DDELE**.

Note — For conflicting constraints on flow degrees of freedom VX, VY, or VZ, zero values (wall conditions) are always given priority over nonzero values (inlet/outlet conditions). "Conflict" in this situation will not produce a warning.

2.6.6. Forces (Concentrated Loads)

Table 2.3: "Forces Available in Each Discipline" shows a list of forces available in each discipline and the corresponding ANSYS labels. Any directions implied by the labels (such as FX, MZ, CSGY, etc.) are in the nodal coordinate system. (See Chapter 3, "Coordinate Systems" in the *ANSYS Modeling and Meshing Guide* for a description of different coordinate systems.) Table 2.4: "Commands for Applying Force Loads" lists the commands to apply, list, and delete forces. Notice that you can apply them at nodes as well as keypoints.

Table 2.3 "Forces" Available in Each Discipline

Discipline	Force	ANSYS Label
Structural	Forces Moments	FX, FY, FZ MX, MY, MZ
Thermal	Heat Flow Rate	HEAT, HBOT, HE2, . . . HTOP
Magnetic	Current Segments Magnetic Flux Electrical Charge	CSGX, CSGY, CSGZ FLUX CHRG
Electric	Current Charge	AMPS CHRG
Fluid	Fluid Flow Rate	FLOW

Table 2.4 Commands for Applying Force Loads

Location	Basic Commands	Additional Commands
Nodes	**F**, **FLIST**, **FDELE**	**FSCALE**, **FCUM**
Keypoints	**FK**, **FKLIST**, **FKDELE**	-
Transfer	**SBCTRAN**	**FTRAN**

Below are examples of some of the GUI paths to use for applying force loads:

GUI:

Main Menu> **Preprocessor**> **Loads**> **Define Loads**> **Apply**> *load type*> **On Nodes**
Utility Menu> **List**> **Loads**> **Forces**> **On All Keypoints (or On Picked KPs)**
Main Menu> **Solution**> **Define Loads**> **Apply**> *load type*> **On Lines**

See the *ANSYS Commands Reference* for descriptions of the commands listed in Table 2.4: "Commands for Applying Force Loads".

2.6.6.1. Repeating a Force

By default, if you repeat a force at the same degree of freedom, the new specification *replaces* the previous one. You can change this default to *add* (for accumulation) or *ignore* by using one of the following:

   Command(s): FCUM
   GUI: Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs Add> Forces
   Main Menu> Solution> Define Loads> Settings> Replace vs. Add> Forces

For example:

```
F,447,FY,3000     ! Applies FY = 3000 at node 447
F,447,FY,2500     ! Changes FY value to 2500 (replacement)
FCUM,ADD          ! Subsequent F's to be added
F,447,FY,-1000    ! FY = 2500-1000 = 1500 at node 447
FCUM,IGNORE       ! Subsequent F's to be ignored
F,25,FZ,350       ! This force is ignored!
FCUM              ! Resets FCUM to default (replacement)
```

See the *ANSYS Commands Reference* for a discussion of the **F** and **FCUM** commands.

Any force set via **FCUM** stays set until another **FCUM** is issued. To reset the default setting (replacement), simply issue **FCUM** without any arguments.

2.6.6.2. Scaling Force Values

The **FSCALE** command allows you to scale existing force values:
   Command(s): FSCALE
   GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Scale FE Loads> Forces
   Main Menu> Solution> Define Loads> Operate> Scale FE Loads> Forces

FSCALE and **FCUM** work on all selected nodes and also *on all selected* force labels. By default, force labels that are active are those associated with the element types in the model. You can select a subset of these with the **DOFSEL** command. For example, to scale only FX values and not any other label, you can use the following commands:

```
DOFSEL,S,FX       ! Selects FX label
FSCALE,0.5        ! Scales FX at all selected nodes by 0.5
DOFSEL,ALL        ! Reactivates all DOF labels
```

2.6.6.3. Transferring Forces

To transfer forces that have been applied to the solid model to the corresponding finite element model, use one of the following:
   Command(s): FTRAN
   GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Transfer to FE> Forces
   Main Menu> Solution> Define Loads> Operate> Transfer to FE> Forces

To transfer all solid model boundary conditions, use the **SBCTRAN** command:

GUI:

   Main Menu> Preprocessor> Loads> Define Loads> Operate> Transfer to FE> All Solid Lds
   Main Menu> Solution> Define Loads> Operate> Transfer to FE> All Solid Lds

2.6.7. Surface Loads

Table 2.5: "Surface Loads Available in Each Discipline" shows surface loads available in each discipline and their corresponding ANSYS labels. The commands to apply, list, and delete surface loads are shown in Table 2.6: "Commands for Applying Surface Loads". You can apply them at nodes and elements, as well as at lines and areas.

Table 2.5 Surface Loads Available in Each Discipline

Discipline	Surface Load	ANSYS Label
Structural	Pressure	PRES[1]
Thermal	Convection Heat Flux Infinite Surface	CONV HFLUX INF
Magnetic	Maxwell Surface Infinite Surface	MXWF INF
Electric	Maxwell Surface Surface Charge Density Infinite Surface	MXWF CHRGS INF
Fluid	Wall Roughness Fluid-Structure Interface Impedance	FSI IMPD
All	Superelement Load Vector	SELV

1. Do not confuse this with the PRES degree of freedom

Table 2.6 Commands for Applying Surface Loads

Location	Basic Commands	Additional Commands
Nodes	**SF**, **SFLIST**, **SFDELE**	**SFSCALE**, **SFCUM**, **SFFUN**, **SFGRAD**
Elements	**SFE**, **SFELIST**, **SFEDELE**	**SFBEAM**, **SFFUN**, **SFGRAD**
Lines	**SFL**, **SFLLIST**, **SFLDELE**	**SFGRAD**
Areas	**SFA**, **SFALIST**, **SFADELE**	**SFGRAD**
Transfer	**SFTRAN**	-

Below are examples of some of the GUI paths to use for applying surface loads.

GUI:

> **Main Menu> Preprocessor> Loads> Define Loads> Apply>** *load type***> On Nodes**
> **Utility Menu> List> Loads> Surface> On All Elements (or On Picked Elements)**
> **Main Menu> Solution> Define Loads> Apply>** *load type***> On Lines**

See the descriptions of the commands listed in Table 2.6: "Commands for Applying Surface Loads" in the *ANSYS Commands Reference* for more information.

> *Note* — The ANSYS program stores surface loads specified on nodes internally in terms of elements and element faces. Therefore, if you use both nodal and element surface load commands for the same surface, only the last specification will be used.

ANSYS applies pressures on axisymmetric shell elements or beam elements on their inner or outer surfaces, as appropriate. In-plane pressure load vectors for layered shells (SHELL91 and SHELL99) are applied on the nodal plane. KEYOPT(11) determines the location of the nodal plane within the shell. When you use flat elements to represent doubly curved surfaces, values which should be a function of the active radius of the meridian will be inaccurate.

2.6.7.1. Applying Pressure Loads on Beams

To apply pressure loads on the lateral faces and the two ends of beam elements, use one of the following:
> **Command(s): SFBEAM**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Apply> Structural> Pressure> On Beams**
> **Main Menu> Solution> Define Loads> Apply> Structural> Pressure> On Beams**

You can apply lateral pressures, which have units of force per unit length, both in the normal and tangential directions. The pressures may vary linearly along the element length, and can be specified on a portion of the element, as shown in the following figure. You can also reduce the pressure down to a force (point load) at any location on a beam element by setting the *JOFFST* field to -1. End pressures have units of force.

Figure 2.8 Example of Beam Surface Loads
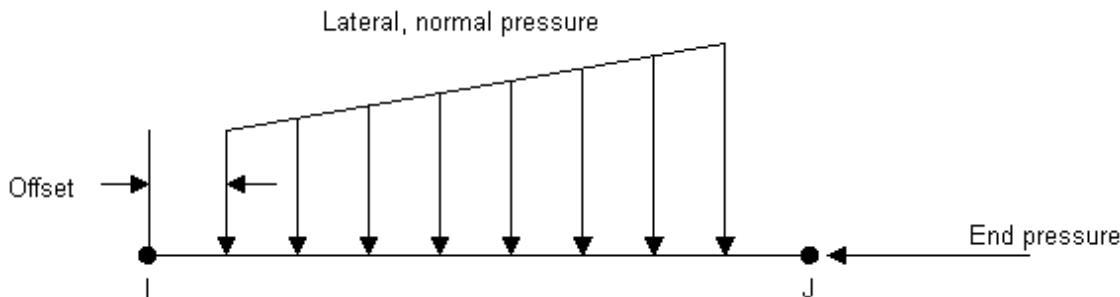
2.6.7.2. Specifying Node Number Versus Surface Load

The **SFFUN** command (**Main Menu> Preprocessor> Loads> Define Loads> Settings> For Surface Ld> Node Function**, or **Main Menu> Solution> Define Loads> Settings> For Surface Ld> Node Function**) specifies a "function" of node number versus surface load to be used when you apply surface loads on nodes or elements. It is useful when you want to apply nodal surface loads calculated elsewhere (by another software package, for instance). You should first define the function in the form of an array parameter containing the load values. The location of the value in the array parameter implies the node number. For example, the array parameter shown below specifies four surface load values at nodes 1, 2, 3, and 4, respectively.

$$ABC = \begin{bmatrix} 400.0 \\ 587.2 \\ 965.6 \\ 740.0 \end{bmatrix}$$

Assuming that these are heat flux values, you would apply them as follows:

```
*DIM,ABC,ARRAY,4              ! Declares dimensions of array parameter ABC
ABC(1)=400,587.2,965.6,740    ! Defines values for ABC
SFFUN,HFLUX,ABC(1)            ! ABC to be used as heat flux function
SF,ALL,HFLUX,100              ! Heat flux of 100 on all selected nodes,
                              !  100 + ABC(i) at node i.
```

See the *ANSYS Commands Reference* for a discussion of the ***DIM**, **SFFUN**, and **SF** commands.

The **SF** command in the example above specifies a heat flux of 100 on all selected nodes. If nodes 1 through 4 are part of the selected set, those nodes are assigned heat fluxes of 100 + ABC(i): 100 + 400 = 500 at node 1, 100 + 587.2 = 687.2 at node 2, and so on.

> *Note* — What you specify with the **SFFUN** command stays active for all subsequent **SF** and **SFE** commands. To remove the specification, simply use **SFFUN** without any arguments.

2.6.7.3. Specifying a Gradient Slope

You can use either of the following to specify that a gradient (slope) is to be used for subsequently applied surface loads:

> **Command(s): SFGRAD**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Settings> For Surface Ld> Gradient**
> **Main Menu> Solution> Define Loads> Settings> For Surface Ld> Gradient**

You can also use this command to apply a linearly varying surface load, such as hydrostatic pressure on a structure immersed in water.

To create the gradient specification, you specify the type of load to be controlled (the Lab argument), the co-ordinate system and coordinate direction the slope is defined in ($SLKCN$ and $Sldir$, respectively), the coordinate location where the value of the load (as specified on a subsequent surface load command) will be in effect ($SLZER$), and the slope ($SLOPE$).

For example, the hydrostatic pressure $(Lab$ = PRES) shown in Figure 2.9: "Example of Surface Load Gradient" is to be applied. Its slope can be specified in the global Cartesian system ($SLKCN$ = 0) in the Y direction ($Sldir$ = Y). The pressure (to be specified as 500 on a subsequent **SF** command) is to have its as-specified value (500) at Y = 0 ($SLZER$ = 0), and will decrease by 25 units per length in the positive Y direction ($SLOPE$ = -25).

Figure 2.9 Example of Surface Load Gradient
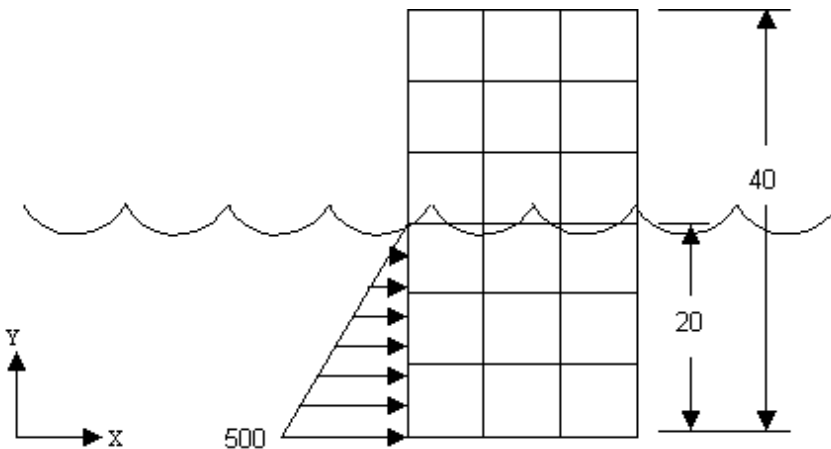
The commands would be as follows:

```
SFGRAD,PRES,0,Y,0,-25    ! Y slope of -25 in global Cartesian
NSEL,...                 ! Select nodes for pressure application
SF,ALL,PRES,500          ! Pressure at all selected nodes:
                         !  500 at Y=0, 250 at Y=10, 0 at Y=20
```

When specifying the gradient in a cylindrical coordinate system ($SLKCN$ = 1, for example), keep some additional points in mind. First, $SLZER$ is in degrees, and $SLOPE$ is in units of load/degree. Second, you need to follow two guidelines:

Guideline 1: Set **CSCIR** (for controlling the coordinate system singularity location) such that the surface to be loaded does *not* cross the coordinate system singularity.

Guideline 2: Choose *SLZER* to be consistent with the **CSCIR** setting. That is, *SLZER* should be between +180° if the singularity is at 180° [**CSCIR**,*KCN*,0], and *SLZER* should be between 0° and 360° if the singularity is at 0° [**CSCIR**,*KCN*,1].

The following example illustrates why these guidelines are suggested. Consider a semicircle shell as shown in Figure 2.10: "Tapered Load on a Cylindrical Shell", located in a local cylindrical system 11. The shell is to be loaded with an external tapered pressure, tapering from 400 at -90° to 580 at +90°. By default, the singularity in the cylindrical system is located at 180°, therefore the θ coordinates of the shell range from -90° to +90°. The following commands will apply the desired pressure load:

```
SFGRAD,PRES,11,Y,-90,1  ! Slope the pressure in the theta direction
                        !   of C.S. 11.  Specified pressure in effect
                        !   at -90°, tapering at 1 unit per degree
SF,ALL,PRES,400         ! Pressure at all selected nodes:
                        !   400 at -90°, 490 at 0°, 580 at +90°.
```

At -90°, the pressure value is 400 (as specified), increasing as θ increases by a slope of 1 unit per degree, to 490 at 0° and 580 at +90°.

Figure 2.10 Tapered Load on a Cylindrical Shell
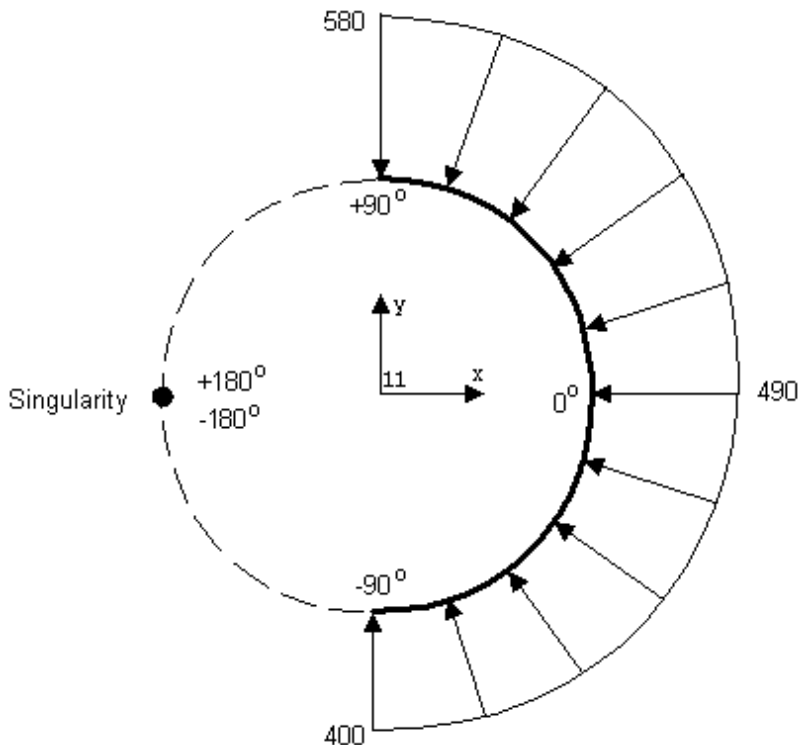
You might be tempted to use 270°, instead of -90°, for *SLZER*:

```
SFGRAD,PRES,11,Y,270,1 ! Slope the pressure in the theta direction
                       !   of C.S. 11.  Specified pressure in effect
                       !   at 270°, tapering at 1 unit per degree
SF,ALL,PRES,400        ! Pressure at all selected nodes:
                       !   400 at -90°, 490 at 0°, 580 at +90°
```

However, as shown on the left in Figure 2.11: "Violation of Guideline 2 (left) and Guideline 1 (right)", this will result in a tapered load much different than intended. This is because the singularity is still located at 180° (the

θ coordinates still range from -90° to +90°), but *SLZER* is not between -180° and +180°. As a result, the program will use a load value of 400 at 270°, and a slope of 1 unit per degree to calculate the applied load values of 220 at +90°, 130 at 0°, and 40 at -90°. You can avoid this behavior by following the second guideline, that is, choosing *SLZER* to be between ±180° when the singularity is at 180°, and between 0° and 360° when the singularity is at 0°.

Figure 2.11 Violation of Guideline 2 (left) and Guideline 1 (right)
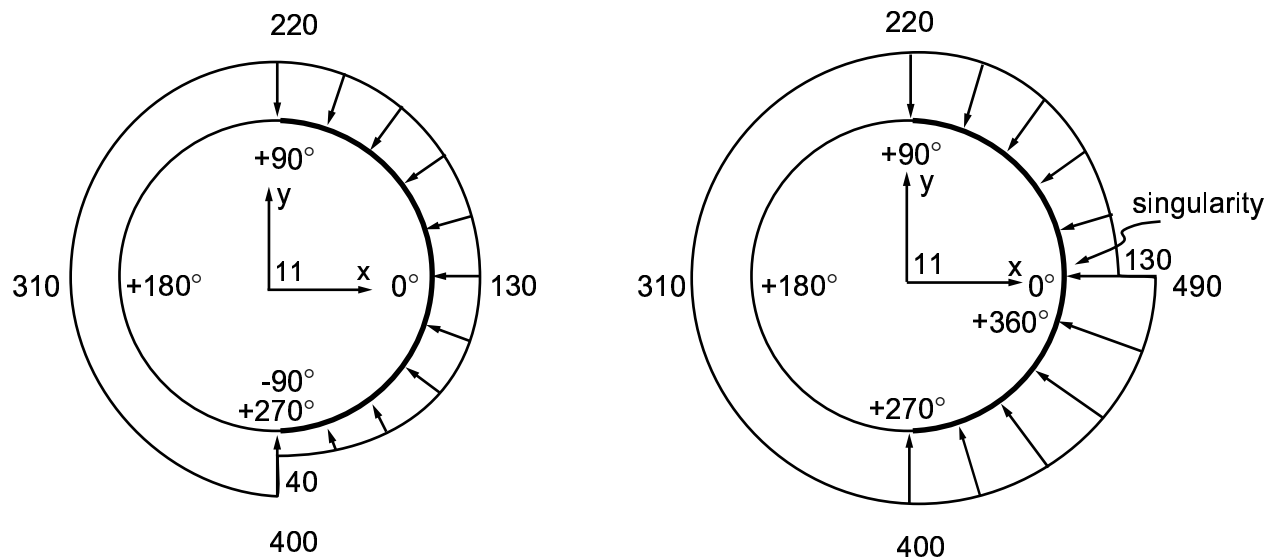
Suppose that you change the singularity location to 0°, thereby satisfying the second guideline (270° is then between 0° and 360°). But then the θ coordinates of the nodes range from 0° to +90° for the upper half of the shell, and 270° to 360° for the lower half. The surface to be loaded crosses the singularity, a violation of Guideline 1:

```
CSCIR,11,1               ! Change singularity to 0°
SFGRAD,PRES,11,Y,270,1   ! Slope the pressure in the theta direction
                         !   of C.S. 11.  Specified pressure in effect
                         !   at 270°, tapering at 1 unit per degree
SF,ALL,PRES,400          ! Pressure at all selected nodes:
                         !   400 at 270°, 490 at 360°, 220 at +90°
                         !   and 130 at 0°
```

Again the program will use a load value of 400 at 270° and a slope of 1 unit per degree to calculate the applied load values of 400 at 270°, 490 at 360°, 220 at 90°, and 130 at 0°. Violating Guideline 1 will cause a singularity in the tapered load itself, as shown on the right in Figure 2.11: "Violation of Guideline 2 (left) and Guideline 1 (right)". Due to node discretization, the actual load applied will not change as abruptly at the singularity as it is shown in the figure. Instead, the node at 0° will have the load value of, in the case shown, 130, while the next node clockwise (say, at 358°) will have a value of 488.

> *Note* — The **SFGRAD** specification stays active for all subsequent load application commands. To remove the specification, simply issue **SFGRAD** without any arguments. Also, if an **SFGRAD** specification is active when a load step file is read, the program erases the specification before reading the file.

Large deflection effects can change the node locations significantly. The **SFGRAD** slope and load value calculations, which are based on node locations, are not updated to account for these changes. If you need this capability, use SURF153 with face 3 loading or SURF154 with face 4 loading.

2.6.7.4. Repeating a Surface Load

By default, if you repeat a surface load at the same surface, the new specification *replaces* the previous one. You can change this default to *add* (for accumulation) or *ignore* using one of the following:

> **Command(s): SFCUM**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs. Add> Surface Loads**
> **Main Menu> Solution> Define Loads> Settings> Replace vs. Add> Surface Loads**

Any surface load you set stays set until you issue another **SFCUM** command. To reset the default setting (replacement), simply issue **SFCUM** without any arguments. The **SFSCALE** command allows you to scale existing surface load values. Both **SFCUM** and **SFSCALE** act only on the selected set of elements. The Lab field allows you to choose the surface load label.

2.6.7.5. Transferring Surface Loads

To transfer surface loads that have been applied to the solid model to the corresponding finite element model, use one of the following:

> **Command(s): SFTRAN**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Transfer to FE> Surface Loads**
> **Main Menu> Solution> Define Loads> Operate> Transfer to FE> Surface Loads**

To transfer all solid model boundary conditions, use the **SBCTRAN** command. (See Section 2.6.3: DOF Constraints for a description of DOF constraints.)

2.6.7.6. Using Surface Effect Elements to Apply Loads

Sometimes, you may need to apply a surface load that the element type you are using does not accept. For example, you may need to apply uniform tangential (or any non-normal or directed) pressures on structural solid elements, radiation specifications on thermal solid elements, etc. In such cases, you can overlay the surface where you want to apply the load with *surface effect* elements and use them as a "conduit" to apply the desired loads. Currently, the following surface effect elements are available: SURF151 and SURF153 for 2-D models and SURF152 and SURF154 for 3-D models.

2.6.8. Body Loads

Table 2.7: "Body Loads Available in Each Discipline" shows all body loads available in each discipline and their corresponding ANSYS labels. The commands to apply, list, and delete body loads are shown in Table 2.8: "Commands for Applying Body Loads". You can apply them at nodes, elements, keypoints, lines, areas, and volumes.

Table 2.7 Body Loads Available in Each Discipline

Discipline	Body Load	ANSYS Label
Structural	Temperature Fluence	TEMP[1] FLUE
Thermal	Heat Generation Rate	HGEN
Magnetic	Temperature Current Density Virtual Displacement Voltage Drop	TEMP[1] JS MVDI VLTG
Electric	Temperature Volume Charge Density	TEMP[1] CHRGD

Discipline	Body Load	ANSYS Label
Fluid	Heat Generation Rate Force Density	HGEN FORC

1.   Do not confuse this with the TEMP degree of freedom.

Table 2.8 Commands for Applying Body Loads

Location	Basic Commands	Additional Commands
Nodes	**BF**, **BFLIST**, **BFDELE**	**BFSCALE, BFCUM, BFUNIF**
Elements	**BFE, BFELIST, BFEDELE**	**BFESCAL, BFECUM**
Keypoints	**BFK**, **BFKLIST**, **BFKDELE**	-
Lines	**BFL**, **BFLLIST**, **BFLDELE**	-
Areas	**BFA, BFALIST, BFADELE**	-
Volumes	**BFV**, **BFVLIST**, **BFVDELE**	-
Transfer	**BFTRAN**	-

For the particular body loads that you can apply, list or delete with any of the commands listed in Table 2.8: "Commands for Applying Body Loads", see the *ANSYS Commands Reference*

Below are examples of some of the GUI paths to use for applying body loads:

GUI:

> **Main Menu> Preprocessor> Loads> Define Loads> Apply>** `load type`**> On Nodes**
> **Utility Menu> List> Loads> Body> On Picked Elems**
> **Main Menu> Solution> Define Loads> Apply>** `load type`**> On Keypoints**
> **Utility Menu> List> Loads> Body> On Picked Lines**
> **Main Menu> Solution> Define Loads> Apply>** `load type`**> On Volumes**

See the *ANSYS Commands Reference* for descriptions of the commands listed in Table 2.8: "Commands for Applying Body Loads".

> *Note* — Body loads you specify on nodes are independent of those specified on elements. For a given element, ANSYS determines which loads to use as follows:

- •   It checks to see if you specified elements for body loads.

- •   If not, it uses body loads specified for nodes.

- •   If no body loads exist for elements or nodes, the body loads specified via the **BFUNIF** command take effect.

2.6.8.1. Specifying Body Loads for Elements

The **BFE** command specifies body loads on an element-by-element basis. However, you can specify body loads at several locations on an element, requiring multiple load values for one element. The locations used vary from element type to element type, as shown in the examples that follow. The defaults (for locations where no body loads are specified) also vary from element type to element type. Therefore, be sure to refer to the element documentation online or in the *ANSYS Elements Reference* before you specify body loads on elements.

- •   For 2-D and 3-D solid elements (PLANE*n* and SOLID*n*), the locations for body loads are usually the corner nodes.

Figure 2.12 BFE Load Locations
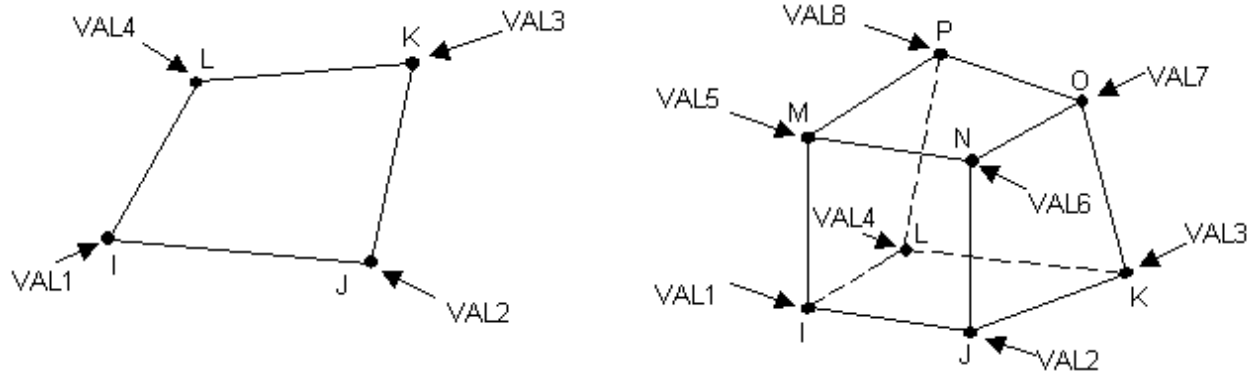
For 2-D and 3-D Solids

- For shell elements (SHELL*n*), the locations for body loads are usually the "pseudo-nodes" at the top and bottom planes, as shown below.

Figure 2.13 BFE Load Locations for Shell Elements
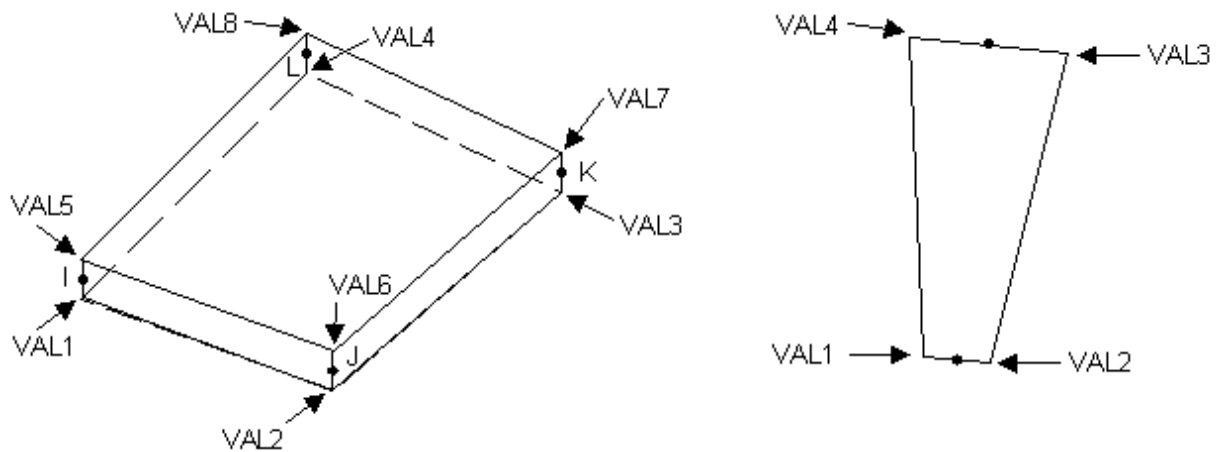
(SHELL63 on the left, SHELL51 on the right)

- Line elements (BEAM*n*, LINK*n*, PIPE*n*, etc.) are similar to shell elements; the locations for body loads are usually the pseudo-nodes at each end of the element.

Figure 2.14 BFE Load Locations for Beam and Pipe Elements
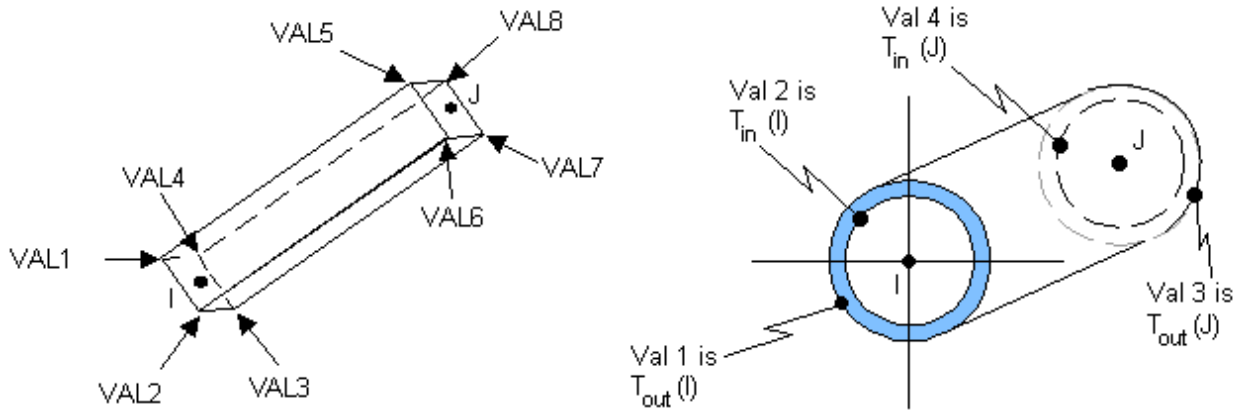
- In all cases, if degenerate (collapsed) elements are involved, you must specify element loads at *all* of its locations, including duplicate values at the duplicate (collapsed) nodes. A simple alternative is to specify body loads directly at the nodes, using the **BF** command.

2.6.8.2. Specifying Body Loads for Keypoints

You can use the **BFK** command to apply body loads at keypoints. If you specify loads at the corner keypoints of an area or a volume, all load values must be equal for the loads to be transferred to the *interior* nodes of the area or volume. If you specify unequal load values, they will be transferred (with linear interpolation) to only the nodes along the lines that connect the keypoints. Figure 2.15: "Transfers to BFK Loads to Nodes" illustrates this:

You can use the **BFK** command to specify table names at keypoints. If you specify table names at the corner keypoints of an area or a volume, all table names must be equal for the loads to be transferred to the *interior* nodes of the area or volume.

Figure 2.15 Transfers to BFK Loads to Nodes
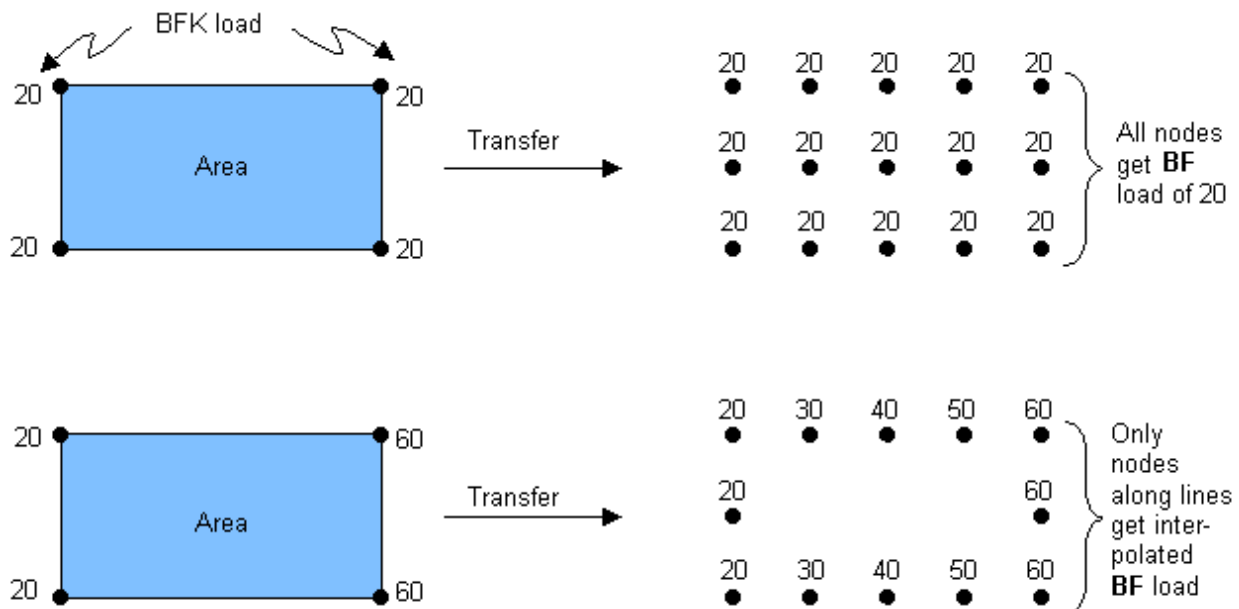
2.6.8.3. Specifying Body Loads on Lines, Areas and Volumes

You can use the **BFL**, **BFA**, and **BFV** commands to specify body loads on lines, areas, and volumes of a solid model, respectively. Body loads on lines of a solid model are transferred to the corresponding nodes of the finite element model. Body loads on areas or volumes of a solid model are transferred to the corresponding elements of the finite element model.

2.6.8.4. Specifying a Uniform Body Load

The **BFUNIF** command specifies a uniform body load at all nodes in the model. Most often, you use this command or path to specify a uniform temperature field; that is, a uniform temperature body load in a structural analysis or a uniform starting temperature in a transient or nonlinear thermal analysis. This is also the default temperature at which the ANSYS program evaluates temperature-dependent material properties.

Another way to specify a uniform temperature is as follows:

> **Command(s): BFUNIF**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Apply>** *Structural or Thermal>* **Temperature> Uniform Temp**
> **Main Menu> Preprocessor> Loads> Define Loads> Settings> Uniform Temp**
> **Main Menu> Solution> Define Loads> Apply>** *Structural or Thermal>* **Temperature> Uniform Temp**
> **Main Menu> Solution> Define Loads> Settings> Uniform Temp**

2.6.8.5. Repeating a Body Load Specification

By default, if you repeat a body load at the same node or same element, the new specification *replaces* the previous one. You can change this default to *ignore* using one of the following:

> **Command(s): BFCUM**, **BFECUM**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs Add> Nodal Body Ld**
> **Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs Add> Elem Body Lds**
> **Main Menu> Solution> Define Loads> Settings> Replace vs Add> Nodal Body Ld**
> **Main Menu> Solution> Define Loads> Settings> Replace vs Add> Elem Body Lds**

The settings you specify with either command or its equivalent GUI paths stay set until they are reuse the command or path. To reset the default setting (replacement), simply issue the commands or choose the paths without any arguments.

2.6.8.6. Transferring Body Loads

To transfer body loads that have been applied to the solid model to the corresponding finite element model, use one of the following:

> **Command(s): BFTRAN**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Transfer to FE> Body Loads**
> **Main Menu> Solution> Define Loads> Operate> Transfer to FE> Body Loads**

To transfer all solid model boundary conditions, use the **SBCTRAN** command. (See Section 2.6.3: DOF Constraints for a description of DOF constraints.)

2.6.8.7. Scaling Body Load Values

You can scale existing body load values using these commands:

Command(s):
   BFSCALE

GUI:

   Main Menu> Preprocessor> Loads> Define Loads> Operate> Scale FE Loads> Nodal Body Ld
   Main Menu> Solution> Define Loads> Operate> Scale FE Loads> Nodal Body Ld

Command(s):
   BFESCAL

GUI:

   Main Menu> Preprocessor> Loads> Define Loads> Operate> Scale FE Loads> Elem Body Lds
   Main Menu> Solution> Define Loads> Operate> Scale FE Loads> Elem Body Lds

BFCUM and **BFSCALE** act on the selected set of nodes, whereas **BFECUM** and **BFESCAL** act on the selected set of elements.

2.6.8.8. Resolution of Conflicting Body Load Specifications

You need to be aware of the possibility of conflicting **BFK**, **BFL**, **BFA**, and **BFV** body load specifications and how the ANSYS program handles them.

BFV, **BFA**, and **BFL** specifications transfer to associated volume, area, and line elements, respectively, where they exist. Where elements do not exist, they transfer to the nodes on the volumes, areas, and lines, including nodes on the region boundaries. The possibility of conflicting specifications depends upon how **BFV**, **BFA**, **BFL** and **BFK** are used as described by the following cases.

CASE A: There are elements for every **BFV**, **BFA**, or **BFL**, and every element belongs to a volume, area or line having a **BFV**, **BFA**, or **BFL**, respectively.

Every element will have its body loads determined by the corresponding solid body load. Any BFK's present will have no effect. No conflict is possible.

CASE B: There are elements for every **BFV**, **BFA**, or **BFL**, but some elements do not belong to a volume area or line having a **BFV**, **BFA**, or **BFL**.

Elements not getting a direct **BFE** transfer from a **BFV**, **BFA**, or **BFL** are unaffected by them, but will have their body loads determined by the following: (1 - highest priority) directly defined **BFE** loads, (2) **BFK** loads, (3) directly defined **BF** loads, or (4) **BFUNIF** loads. No conflict among solid model body loads is possible.

CASE C: At least one **BFV**, **BFA**, or **BFL** cannot transfer to elements.

Elements not getting a direct **BFE** transfer from a **BFV**, **BFA**, or **BFL** will have their body loads determined by the following: (1 - highest priority) directly defined **BFE** loads, (2) **BFK** loads, (3) **BFL** loads on an attached line that did NOT transfer to line elements, (4) **BFA** loads on an attached area that did NOT transfer to area elements, (5) **BFV** loads on an attached volume that did NOT transfer to volume elements, (6) directly defined **BF** loads, or (7) **BFUNIF** loads.

In "Case C" situations, the following conflicts can arise:

- A **BFL** specification can conflict with a **BFL** specification on an adjacent line (shared keypoint).
- A **BFL** specification can conflict with a **BFK** specification at either keypoint.
- A **BFA** specification can conflict with a **BFA** specification on an adjacent area (shared lines/keypoints).

- A **BFA** specification can conflict with a **BFL** specification on any of its lines.

- A **BFA** specification can conflict with a **BFK** specification on any of its keypoints.

- A **BFV** specification can conflict with a **BFV** specification on an adjacent volume (shared areas/lines/keypoints).

- A **BFV** specification can conflict with a **BFA** specification on any of its areas.

- A **BFV** specification can conflict with a **BFL** specification on any of its lines.

- A **BFV** specification can conflict with a **BFK** specification on any of its keypoints.

The ANSYS program transfers body loads that have been applied to the solid model to the corresponding finite element model in the following sequence:

1. In ascending volume number order, **BFV** loads transfer to **BFE** loads on volume elements, or, if there are none, to **BF** loads on nodes on volumes (and bounding areas, lines, and keypoints).

2. In ascending area number order, **BFA** loads transfer to **BFE** loads on area elements, or, if there are none, to **BF** loads on nodes on areas (and bounding lines and keypoints).

3. In ascending line number order, **BFL** loads transfer to **BFE** loads on line elements, or, if there are none, to **BF** loads on nodes on lines (and bounding keypoints).

4. **BFK** loads transfer to **BF** loads on nodes on keypoints (and on attached lines, areas, and volumes if expansion conditions are met).

Accordingly, for conflicting solid model body loads in "Case C" situations, **BFK** commands overwrite **BFL** commands, **BFL** commands overwrite **BFA** commands, and **BFA** commands overwrite **BFV** commands. For conflicting body loads, a body load specified for a higher line number, area number, or volume number overwrites the body load specified for a lower line number, area number, or volume number, respectively. The body load specification issue order does not matter.

> *Note* — Any conflict detected during solid model body load transfer produces a warning similar to the following:

```
***WARNING***
Body load TEMP from line 12 (1st value=77) is overwriting a BF on
node 43 (1st value=99) that was previously transferred from another
BFV, BFA, BFL or set of BFK's.
```

Changing the value of **BFK**, **BFL**, **BFA**, or **BFV** constraints between solutions may produce many of these warnings at the 2nd or later solid BC transfer. These can be prevented if you delete the nodal **BF** loads between solutions using **BFVDELE**, **BFADELE**, **BFLDELE**, and/or **BFDELE**.

2.6.9. Inertia Loads

You use the following commands for inertia loads:

Table 2.9 Inertia Loads Commands

Command	
ACEL	DCGOMG
CGLOC	DOMEGA
CGOMGA	IRLF
CMDOMEGA	OMEGA

Command	
CMOMEGA	

There are no specific commands to list or delete inertia loads. To list them, issue **STAT**, **INRTIA**. To remove an inertia load, set the load value to zero. You can set an inertia load to zero, but you cannot delete it. For ramped load steps, inertia loads are ramped to zero. (This is also true when you apply inertia loads.)

The **ACEL**, **OMEGA**, and **DOMEGA** commands specify acceleration, angular velocity, and angular acceleration, respectively, in global Cartesian directions.

> *Note* — The **ACEL** command applies an acceleration field (not gravity) to a body. Therefore, to apply gravity to act in the negative Y direction, you should specify a positive Y acceleration.

Use the **CGOMGA** and **DCGOMG** commands to specify angular velocity and angular acceleration of a spinning body which is itself revolving about another reference coordinate system. The **CGLOC** command specifies the location of the reference system with respect to the global Cartesian origin. You can use these commands, for example, to include Coriolis effects in a static analysis.

You can also use the **CMOMEGA** and **CMDOMEGA** commands to specify the rotational velocity and acceleration effects for element components you define. You either specify an axis and the scalar vector quantity, or define the three components of the rotational value and the point in space you are considering. You can use these commands for Element components only.

Inertia loads are effective only if your model has some mass, which is usually supplied by a density specification. (You can also supply mass to the model by using mass elements, such as MASS21, but density is more commonly used and is more convenient.) As with all other data, the ANSYS program requires you to use consistent units for mass. If you are accustomed to the British system of units, you might sometimes wish to use *weight density* (lb/in^3) instead of *mass density* ($lb\text{-}sec^2/in/in^3$), for convenience.

Use weight density in place of mass density only under these conditions:

- The model will only be used in a static analysis.
- No angular velocity or angular acceleration is applied.
- Gravitational acceleration is unity (g = 1.0).

A handy way to specify density so that you can use it readily in either a "convenient," weight-density form or "consistent," mass-density form is to define a parameter for gravitational acceleration, g:

Table 2.10 Ways of Specifying Density

Convenient Form	Consistent Form	Description
g = 1.0	g = 386.0	Parameter definition
MP,DENS,1,0.283/g	MP,DENS,1,0.283/g	Density of steel
ACEL,,g	ACEL,,g	Gravity load

2.6.10. Coupled-Field Loads

A coupled-field analysis usually involves applying results data from one analysis as loads in a second analysis. For example, you can apply the nodal temperatures calculated in a thermal analysis as body loads in a structural analysis (for thermal strain). Similarly, you can apply magnetic forces calculated in a magnetic field analysis as nodal forces in a structural analysis. To apply such coupled-field loads, use one of the following:

Command(s): LDREAD
GUI: **Main Menu> Preprocessor> Loads> Define Loads> Apply>** *load type*> **From source**
Main Menu> Solution> Define Loads> Apply> *load type*> **From source**

See the *ANSYS Coupled-Field Analysis Guide* for details about how to use this command in different types of coupled-field analyses.

2.6.11. Axisymmetric Loads and Reactions

For constraints, surface loads, body loads, and Y-direction accelerations, you define loads exactly as they would be for any nonaxisymmetric model. However, for concentrated forces the procedure is a little different. For these quantities, input load values of force, moment, etc. are on a "360° basis." That is, the load value is entered in terms of *total load around the circumference*. For example, if an axisymmetric axial load of 1500 pounds per inch of circumference were applied to a 10" diameter pipe (Figure 2.16: "Concentrated Axisymmetric Loads"), the total load of 47,124 lb. (1500*2 π*5 = 47,124) would be applied to node *N* as follows:

```
F,N,FY,-47124
```

Axisymmetric results are interpreted in the same fashion as their corresponding input loads. That is, reaction forces, moments, etc. are reported on a total load (360°) basis.

Axisymmetric harmonic elements require that their loads be supplied in a form that the program can interpret as a Fourier series. The **MODE** command (**Main Menu> Preprocessor> Loads> Load Step Opts> Other> For Harmonic Ele** or **Main Menu> Solution> Load Step Opts> Other> For Harmonic Ele**), together with other load commands (**D**, **F**, **SF**, etc.), is required for these elements. See the *ANSYS Commands Reference* for details.

Figure 2.16 Concentrated Axisymmetric Loads

(a) 3-D structure   (b) 2-D model
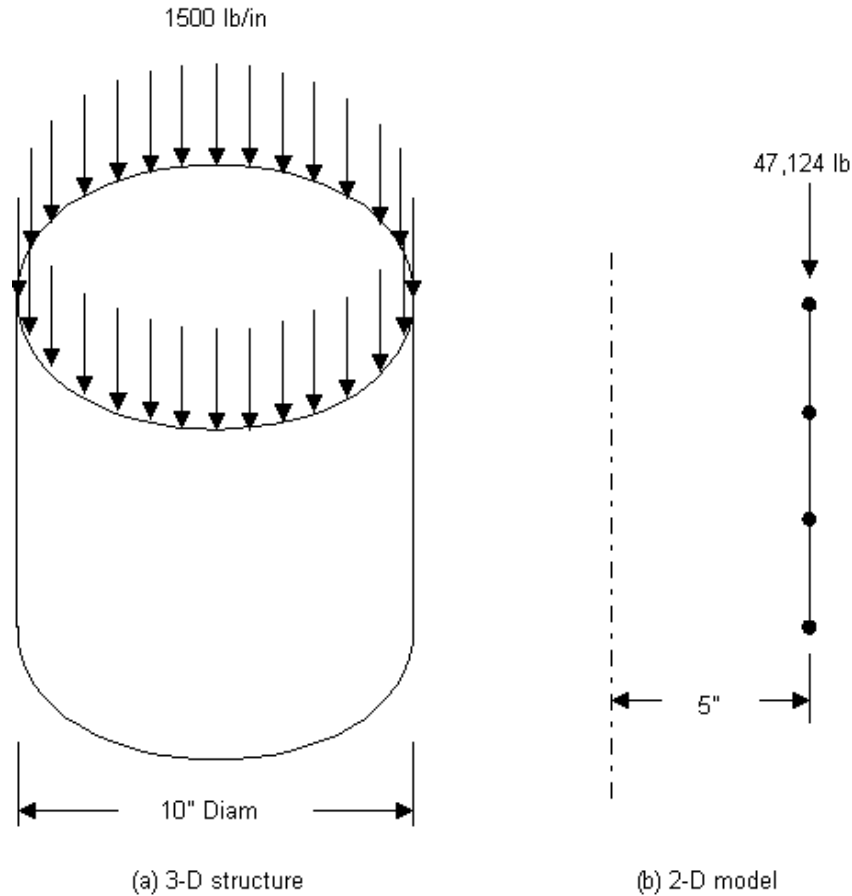
Defined on a 360° basis

2.6.11.1. Further Hints and Restrictions

Be careful to specify a sufficient number of constraints to prevent unwanted rigid-body motions, discontinuities, or singularities. For instance, for an axisymmetric model of a solid structure such as a solid bar, a lack of UX constraint along the axis of symmetry can potentially allow spurious "voids" to form in a structural analysis. (See Figure 2.17: "Central Constraint for Solid Axisymmetric Structure".)

Figure 2.17 Central Constraint for Solid Axisymmetric Structure
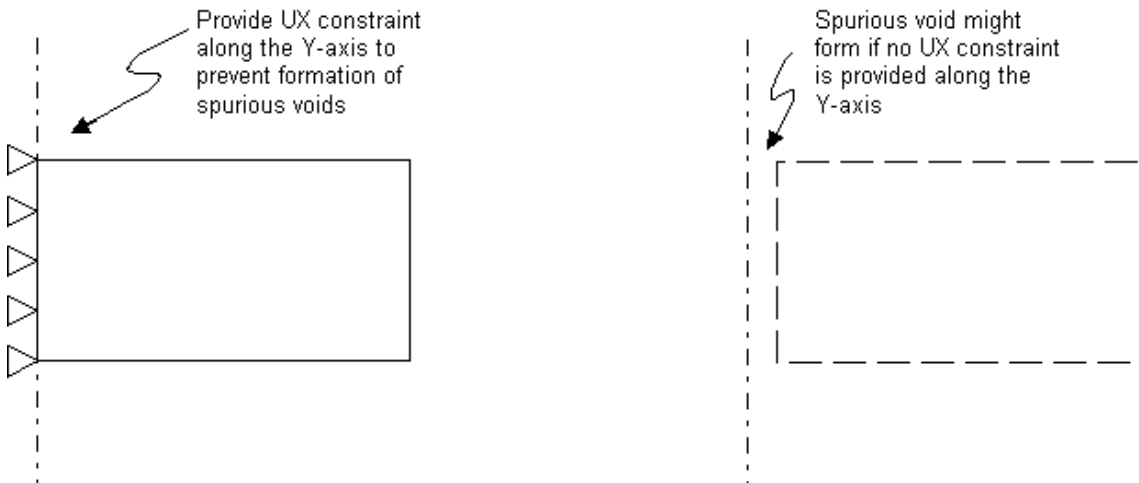
2.6.12. Loads to Which the DOF Offers No Resistance

If an applied load acts on a DOF which offers no resistance to it (i.e. perfectly zero stiffness), the ANSYS program ignores the load. For example, consider a series of connected colinear LINK1 elements. Loads normal to the line of the links are ignored when you apply them to interior DOFs. If, however, the links are under tension and stress stiffening is being used, the loads are not ignored because there is resistance (stiffness) in the direction of the loads. The same principle applies to membrane shell elements.

2.6.13. Initial Stress Loading

You can specify initial stress as a loading in ANSYS when doing a structural analysis. Initial stress loading is only allowed in a static or full transient analysis (the analysis can be linear or nonlinear). Initial stresses can be applied only in the first load step of an analysis.

> *Note* — Initial stresses are treated as if they are applied on elastic materials.

Constant stresses can be specified using the **ISTRESS** command. Use the **ISFILE** command to read initial stresses from an input file, and to list and delete initial stresses. Command arguments allow the initial stresses to be limited to certain material types.

To apply initial stresses for a specific layer of the layered SHELL181 element, you can use the specific layer's material ID number. Use the $MATx$ fields on the initial stress commands to apply the required stress based on the material ID number of that layer. Alternatively, you may apply the initial stresses to all layers of the SHELL181 element and then delete those initial stresses on unwanted layers with the **ISFILE**, delete, element ID, layer number command. You can list and delete the initial stresses specification on a specific layer of the layered SHELL181 element with the initial stress commands.

This initial stress capability is only supported by the following element types: PLANE2, PLANE42, SOLID45, PLANE82, SOLID92, SOLID95, LINK180, SHELL181, PLANE182, PLANE183, SOLID185, SOLID186, SOLID187, BEAM188, and BEAM189. To specify initial stress for BEAM188 or BEAM189, use only the SX, SXY, and SXZ components on the **ISTRESS** command or in the initial stress file read by **ISFILE**.

The **ISWRITE** command can be used to write the resulting initial stresses to a file. These commands can be used in the /SOLUTION processor only. See the *ANSYS Commands Reference* for a description of all of the initial stress commands.

Only one method (**ISFILE**, **ISTRESS**, or the USTRESS user subroutine) can be used to input initial stresses for an element. Issuing the initial stress commands more than once will overwrite the previous specification of the initial stresses.

2.6.13.1. Initial Stress File Format

The *LOC* argument of the **ISFILE** command lets you specify the location of the initial stresses that are read in from the file. Initial stresses can be specified either at the centroid of the elements or at the element integration points. A default value of 0 for *LOC* corresponds to the element centroid. *LOC* = 1 corresponds to the element integration points. You can also specify different initial stress locations for each of the individual elements in the mesh by setting *LOC* = 2. In this case, the initial stress location for each element will be indicated by a local location flag in the individual element records in the initial stress file. *LOC* = 3 specifies that the initial stress state is the same for every element in the mesh. For this case, only one stress tensor needs to be specified for all the elements. The locations and number of the element integration points for the various element types are given in Section 13.1 and Chapter 14 in the *ANSYS, Inc. Theory Reference*. The initial stresses for BEAM and SHELL elements must be specified at all the domain and section integration points.

To use the **ISFILE** command, initial stresses must be listed in an external ASCII file. Comments in the initial stress file can be specified by using the "!" mark as the first character on the comment line. The first line for each element record should start with the character string "EIS" followed by the element number and an optional local location flag. These items must be separated by commas. The local location flag will be ignored if the *LOC* argument of the **ISFILE** command has a value of 0, 1, or 3. If *LOC* = 2, then a local location flag must be specified for every element. The local location flag must have one of the following values: 0 for element centroid (default), or 1 for integration points. Any other value will generate an error and cause the **ISFILE** command to be ignored.

This first line for each element is followed by subsequent lines that specify the element stress records for each stress point in the element. When *LOC* = 0, only one stress record at the centroid of each element must be specified. When *LOC* = 1, the number of stress records for each element is equal to the number of element integration points for the element. ANSYS expects six stress tensor components in each stress record. Based on the element type, ANSYS will read only the relevant stress tensor components from each stress record. When *LOC* = 3, the stress record for the first element in the initial stress file will be used to specify the uniform initial stresses for all elements. The initial stresses must be specified in the element local coordinate system. If an element coordinate system (ESYS) is defined for an element, the initial stresses must be specified in this coordinate system.

The initial stress file written by the **ISWRITE** command has the same format as described for the input file.

2.6.13.2. A Sample Initial Stress Problem Using ISFILE

The following example problem shows how you would define an initial stress file and use **ISFILE** to read in the file.

The model is a cantilever beam of length 10 units and depth 1 unit. The beam is modeled by five PLANE42 elements. At X = 0, all the nodes are constrained to have zero displacement.

The following file, istress.ist, contains the initial stresses to be read into ANSYS. Each element has four integration points in the domain of the element, so each element must have four stress records corresponding to the four integration points.

```
! ************ STRESS INITIALIZATION FILE FOR ANSYS ************
!
! This file, istress.ist, contains initial stress data
! for 5 PLANE42 elements.
!
!  Stress for element 1
!    Sx           Sy           Sz         Sxy         Syz   Sxz
eis,    1
```

```
 -3117691.453622 , -87.56555196445 , 0.0, -99999.99999990 , 0.0, 0.0
 -3117691.453622 , -326.7990889251 , 0.0, -99999.99999990 , 0.0, 0.0
  3117691.453622 , -326.7990889248 , 0.0, -99999.99999990 , 0.0, 0.0
  3117691.453622 , -87.56555196422 , 0.0, -99999.99999990 , 0.0, 0.0
!
!  Stress for element 2
!     Sx             Sy            Sz        Sxy         Syz   Sxz
eis,    2
 -2424871.130595 ,  23.46311893908 , 0.0, -99999.99999991 , 0.0, 0.0
 -2424871.130595 ,  1219.630803748 , 0.0, -99999.99999991 , 0.0, 0.0
  2424871.130595 ,  1219.630803748 , 0.0, -99999.99999991 , 0.0, 0.0
  2424871.130595 ,  23.46311893943 , 0.0, -99999.99999991 , 0.0, 0.0
!
!  Stress for element 3
!     Sx             Sy            Sz        Sxy         Syz   Sxz
eis,    3
 -1732050.807568 , -6.286923759268 , 0.0, -99999.99999989 , 0.0, 0.0
 -1732050.807568 , -4551.724125935 , 0.0, -99999.99999990 , 0.0, 0.0
  1732050.807568 , -4551.724125935 , 0.0, -99999.99999989 , 0.0, 0.0
  1732050.807568 , -6.286923743552 , 0.0, -99999.99999989 , 0.0, 0.0
!
!  Stress for element 4
!     Sx             Sy            Sz        Sxy         Syz   Sxz
eis,    4
 -1039230.484542 ,  1.684576148808 , 0.0, -99999.99999990 , 0.0, 0.0
 -1039230.484542 ,  16987.26570008 , 0.0, -99999.99999991 , 0.0, 0.0
  1039230.484542 ,  16987.26570014 , 0.0, -99999.99999991 , 0.0, 0.0
  1039230.484541 ,  1.684576134896 , 0.0, -99999.99999990 , 0.0, 0.0
!
!  Stress for element 5
!     Sx             Sy            Sz        Sxy         Syz   Sxz
eis,    5
  -346410.1615140 ,-0.4513808515767 , 0.0, -100000.0000001 , 0.0, 0.0
  -346410.1615140 , -63397.33867442 , 0.0, -100000.0000001 , 0.0, 0.0
   346410.1615140 , -63397.33867442 , 0.0, -100000.0000001 , 0.0, 0.0
   346410.1615140 ,-0.4513808514312 , 0.0, -100000.0000001 , 0.0, 0.0
!
! End of initial stress file
!**************************************************************
```

In the following ANSYS input listing, initial stresses are read in as loading (using the **ISFILE** command) in the first load step from the file istress.ist. These initial stresses result in a deflection corresponding to a tip loaded cantilever beam with a tip load of 1e5 units.

```
/prep7
/title, Example of Initial stress import into ANSYS
et,1,PLANE42    ! Plane stress PLANE42 element
mp,ex,1,1.0e9
mp,nuxy,1,0.3
!
! Define the nodes
!
n,1
n,2,2.0
n,3,4.0
n,4,6.0
n,5,8.0
n,6,10.0
n,7,,1.0
n,8,2.0,1.0
n,9,4.0,1.0
n,10,6.0,1.0
n,11,8.0,1.0
n,12,10.0,1.0
!
! Define the 5 elements
!
e,1,2,8,7
e,2,3,9,8
e,3,4,10,9
e,4,5,11,10
```

```
e,5,6,12,11
! Constrain all dofs on all nodes at x=0 to be zero
nsel,s,loc,x,
d,all,all
nall
finish
!
/solu
! Read in the initial stresses from istress.ist file
! as loading in the 1st load step.
! Input stresses correspond to the element integration
! point location. This is done by setting
! the LOC field on ISFILE to 1.
!
isfile,read,istress,ist,,1
! List the initial stresses
isfile,list
outres,all,all
solve
finish
!
/post1
set,last
prnsol,u
finish
```

2.6.13.3. A Sample Initial Stress Problem Using ISTRESS

Constant stresses can be applied to all selected elements using the **ISTRESS** command. The **ISFILE** command can be used to delete stress from individual elements after the stress is applied. **ISFILE** can also be used to list the applied stresses. The following code sample shows how these commands are used.

```
solution
!
! Apply a constant state of the initial stresses.
!
istress,1322.34,2022.21,302.43,4040.32,5076.32,6021.456
!
! Verify the applied stresses then delete those of element #1
!
isfile,list
isfile,dele,1
!
! Verify the final result, then solve.
!
isfile,list
solve
FINISH
```

2.6.13.4. A Sample Output from ISWRITE

The initial stress file written by the **ISWRITE** command has the same format as described for the input file. The stresses in the file are those calculated at the integration points when the convergence happens in the nonlinear analysis. If the analysis type is linear, the stresses are those calculated when the solution is finished. A sample initial stress file resulting from this command looks as follows:

```
!
! ******* INITIAL STRESS FILE   ev181-64s.ist            <----  File Name
!
!
!INITIAL STRESS RECORD FOR ELEMENT       3               <---- Element Records.
!    SX           SY           SZ          SXY           SYZ           SXZ
eis,      3,1
    78.6886        122.471      0.000000E+00   55.7557      0.000000E+00 0.000000E+00
    47.2161        73.4892      0.000000E+00   33.4552      0.000000E+00 0.000000E+00
    0.730024E-02  0.167903E-01  0.000000E+00  0.442187E-02  0.000000E+00 0.000000E+00
   -47.2015       -73.4556      0.000000E+00  -33.4464      0.000000E+00 0.000000E+00
   -78.6740      -122.437       0.000000E+00  -55.7469      0.000000E+00 0.000000E+00
```

```
    !
    !
    !INITIAL STRESS RECORD FOR ELEMENT        2
    !    SX          SY           SZ           SXY         SYZ         SXZ
    eis,      2,1
       111.393       -10.2521     0.000000E+00  -5.40234     0.000000E+00 0.000000E+00
       66.8419       -6.14723     0.000000E+00  -3.24392     0.000000E+00 0.000000E+00
      0.149626E-01  0.101159E-01  0.000000E+00 -0.630216E-02 0.000000E+00 0.000000E+00
      -66.8120        6.16746     0.000000E+00   3.23132     0.000000E+00 0.000000E+00
      -111.363       10.2724      0.000000E+00   5.38973     0.000000E+00 0.000000E+00
    !
    !
    !INITIAL STRESS RECORD FOR ELEMENT        1
    !    SX          SY           SZ           SXY         SYZ         SXZ
    eis,      1,1
      -486.946      -146.209      0.000000E+00  -82.7928     0.000000E+00 0.000000E+00
      -292.163       -87.7187     0.000000E+00  -49.6735     0.000000E+00 0.000000E+00
      0.121788E-01  0.172217E-01  0.000000E+00  0.534346E-02 0.000000E+00 0.000000E+00
       292.187        87.7531     0.000000E+00   49.6842     0.000000E+00 0.000000E+00
       486.971       146.244      0.000000E+00   82.8035     0.000000E+00 0.000000E+00
    !
    !
```

2.6.14. Applying Loads Using TABLE Type Array Parameters

To apply loads using TABLE parameters, you use the appropriate loading commands or menu paths for your analysis. However, instead of specifying an actual value for a particular load, you specify the name of a table array parameter. Not all boundary conditions support tabular loads; please refer to the documentation on the specific loads you are working with to determine if tabular loads are supported.

> *Note* — When defining loads via commands, you must enclose the table name in % symbols: %tabname%. For example, to specify a table of convection values, you would issue a command similar to the following:

```
SF,all,conv,%sycnv%,tbulk
```

If your data cannot be conveniently expressed as a table, you may want to use function boundary conditions. See Section 2.6.15: Applying Loads Using Function Boundary Conditions.

If working interactively, you can define a new table at the time you apply the loads by selecting the "new table" option. You will be asked to define the table through a series of dialog boxes. You can also define a table before you apply loads by choosing the menu path **Utility Menu> Parameters> Array Parameters> Define/Edit**, or by using the ***DIM** command. Tabular loads can be defined in both the global Cartesian (default) or a local co-ordinate system you define with the **LOCAL** command (only Cartesian, spherical and cylindrical coordinate systems are valid). If working in batch mode, you need to define the table before issuing any of the loading commands.

For more information on defining table array parameters (both interactively and via command), see Section 3.11.3: TABLE Type Array Parameters of the *ANSYS APDL Programmer's Guide*.

2.6.14.1. Defining Primary Variables

When you define the table array parameter, you can define various primary variables, depending on the type of analysis you are doing. Table 2.11: "Boundary Condition Type and Corresponding Primary Variable" lists boundary conditions and their associated primary variables for supported types of analyses. Additional primary variables are available using function boundary conditions. See Section 2.6.15.1: Using the Function Editor for more information. Primary variables are shown as the valid labels used by the ***DIM** command. You can apply tabular loads according to a local coordinate system defined via **LOCAL**, and specified in ***DIM**.

When defining the tables, the primary variables must be in ascending order in the table indices (as in any table array).

Table 2.11 Boundary Condition Type and Corresponding Primary Variable

Boundary Condition	Primary Variable	Command
Thermal Analyses		
Fixed Temperature	TIME, X, Y, Z	**D**,,(TEMP, TBOT, TE2, TE3, . . ., TTOP)
Heat Flow	TIME, X, Y, Z, TEMP	**F**,,(HEAT, HBOT, HE2, HE3, . . ., HTOP)
Film Coefficient (Convection)	TIME, X, Y, Z, TEMP, VELOCITY	**SF**,,CONV
Bulk Temperature (Convection)	TIME, X, Y, Z	**SF**,,,TBULK
Heat Flux	TIME, X, Y, Z, TEMP	**SF**,,HFLU
Heat Generation	TIME, X, Y, Z, TEMP	**BFE**,,HGEN
Uniform Heat Generation	TIME	**BFUNIF**,TEMP
Structural Analyses		
Displacements	TIME, X, Y, Z, TEMP	**D**,(UX, UY, UZ, ROTX, ROTY, ROTZ)
Forces and moments	TIME, X, Y, Z, TEMP, SECTOR	**F**,(FX, FY, FZ, MX, MY, MZ)
Pressures	TIME, X, Y, Z, TEMP, SECTOR	**SF**,,PRES
Temperature	TIME	**BF**,,TEMP
Electrical Analyses		
Voltage	TIME, X, Y, Z	**D**,,VOLT
Current	TIME, X, Y, Z	**F**,,AMPS
Fluid Analyses		
Pressure	TIME, X, Y, Z	**D**,,PRES
Flow	TIME, X, Y, Z	**F**,,FLOW
FLOTRAN Analyses		
Nodal DOF	TIME, X, Y, Z, TEMP, VELOCITY, PRESSURE	**D**,,(VX, VY, VZ, PRES, TEMP, ENKE, ENDS, SP01-SP06)
Nodal DOF for ALE formulation	TIME, X, Y, Z, TEMP, VELOCITY, PRES, Xr, Yr, Zr	**D**,,(UX, UY, UZ)
Heat Flux Film Coefficient	TIME, X, Y, Z, TEMP, VELOCITY, PRESSURE	**SF**,,HFLU **SF**,,CONV
Element Heat Generation	TIME, X, Y, Z, TEMP, VELOCITY, PRESSURE	**BFE**,,HGEN
Nodal Heat Generation	TIME, X, Y, Z, TEMP, VELOCITY, PRESSURE	**BF**,,HGEN
Nodal Body Force	TIME, X, Y, Z, TEMP, VELOCITY, PRESSURE	**BF**,,FORCE
Radiation	TIME, X, Y, Z, TEMP, VELOCITY, PRESSURE	**SF**,,RAD

See the ***DIM** command for more information on defining your labels.

The *VELOCITY* label refers to the magnitude of the velocity degrees of freedom or the computed fluid velocity in FLUID116 elements.

In addition, some real constants for elements SURF151, SURF152, and FLUID116 can have associated primary variables.

Table 2.12 Real Constants and Corresponding Primary Variable

Real Constants	Primary Variables
SURF151, SURF152	
Rotational Speed	TIME, X. Y, Z
FLUID116	
Rotational Speed	TIME, X, Y, Z
Slip Factor	TIME, X, Y, Z

2.6.14.2. Defining Independent Variables

If you need to specify a variable other than one of the primary variables listed, you can do so by defining an independent parameter. To specify an independent parameter, you define an additional table for the independent parameter. That table must have the same name as the independent parameter, and can be a function of either a primary variable or another independent parameter. You can define as many independent parameters as necessary, but all independent parameters must relate to a primary variable.

For example, consider a convection coefficient (HF) that varies as a function of rotational speed (RPM) and temperature (TEMP). The primary variable in this case is TEMP. The independent parameter is RPM, which varies with time. In this scenario, you need two tables: one relating RPM to TIME, and another table relating HF to RPM and TEMP.

```
*DIM,SYCNV,TABLE,3,3,,RPM,TEMP
SYCNV(1,0)=0.0,20.0,40.0
SYCNV(0,1)=0.0,10.0,20.0,40.0
SYCNV(0,2)=0.5,15.0,30.0,60.0
SYCNV(0,3)=1.0,20.0,40.0,80.0
*DIM,RPM,TABLE,4,1,1,TIME
RPM(1,0)=0.0,10.0,40.0,60.0
RPM(1,1)=0.0,5.0,20.0,30.0
SF,ALL,CONV,%SYCNV%
```

When defining the tables, the independent variables must be in ascending order in the table indices (as in any table array).

2.6.14.3. Operating on Table Parameters

For convenience, you can multiply table parameters by constants, add one table to another, and add a constant increment for offset. To do so, use the ***TOPER** command (**Utility Menu> Parameters> Array Operations> Table Operations**). The two tables must have the same dimensions and must have the same variable names for the rows and columns. The tables must also have identical index values for rows, columns, etc.

2.6.14.4. Verifying Boundary Conditions

If you use table array parameters to define boundary conditions, you may want to verify that the correct table and the correct values from the table were applied. You can do so in several ways:

- You can look in the Output window. If you apply tabular boundary conditions on finite element or solid model entities, the name of the table, not the numerical value, is echoed in the Output window.

- You can list boundary conditions. If you list the boundary conditions during /PREP7, table names are listed. However, if you list boundary conditions during any of the solution or postprocessing phases at a particular entity or time point, the actual numerical value at the location or time is listed.

- You can look at the graphical display. Where tabular boundary conditions were applied, the table name and any appropriate symbols (face outlines, arrows, etc.) can be displayed using the standard ANSYS graphic display capabilities (**/PBC**, **/PSF**, etc.), provided that table numbering is on (**/PNUM**,*TABNAM*,ON).

- You can look at the numerically-substituted table of values (**/PNUM**,*SVAL*) in POST1.

- You can retrieve a value of a table parameter at any given combination of variables using the ***STATUS** command (**Utility Menu> List> Other> Parameters**).

2.6.14.5. Example Analysis Using 1-D Table Array

An example of how to run a steady-state thermal analysis using tabular boundary conditions is described in Section 2.9: Doing a Thermal Analysis Using Tabular Boundary Conditions.

2.6.14.6. Example Analysis Using 5-D Table Array

This example shows how to run an analysis using a 5-D table. Note that 4- and 5-D tables cannot be defined interactively; you must use the command method.

This problem consists of a thermal-stress analysis with a pressure that varies as a function of (x,y,z,time,temp). The table and table values are first defined. The table is applied as a pressure boundary condition to the faces of a rectangular beam. Time and temperature are prescribed for two load steps and solved.

```
/batch,list
/title, Illustrate use of 5D table for SF command (pressure) loading
!!!!
!!!!
                                !!!! create 5D table for applied pressure
X1=2                            !!!! X dimensionality
Y1=2                            !!!! Y dimensionality
Z1=10                           !!!! Z dimensionality
D4=5                            !!!! time dimensionality
D5=5                            !!!! temperature dimensionality
len=10                          !!!! cantilever beam length
wid=1                           !!!! cantilever beam width
hth=2                           !!!! cantilever beam height


*dim,xval,array,X1              !!!! create 1D arrays to load 5D table
xval(1)=0,20                    !!!! variations per dimension same
*dim,yval,array,Y1              !!!! but will give different values on each
yval(1)=0,20                    !!!! book and shelf
*dim,zval,array,10
zval(1)=10,20,30,40,50,60,70,80,90,100
*dim,tval,array,5
tval(1)=1,.90,.80,.70,.60
*dim,tevl,array,5
tevl(1)=1,1.20,1.30,1.60,1.80

*dim,ccc,tab5,X1,Y1,Z1,D4,D5,X,Y,Z,TIME,TEMP
*taxis,ccc(1,1,1,1,1),1,0,wid    !!! X-Dim
*taxis,ccc(1,1,1,1,1),2,0,hth    !!! Y-Dim
*taxis,ccc(1,1,1,1,1),3,1,2,3,4,5,6,7,8,9,10 !!! Z-Dim
*taxis,ccc(1,1,1,1,1),4,0,10,20,30,40  !!! Time
*taxis,ccc(1,1,1,1,1),5,0,50,100,150,200 !!! Temp
*do,ii,1,2
    *do,jj,1,2
        *do,kk,1,10
            *do,ll,1,5
                *do,mm,1,5
                    ccc(ii,jj,kk,ll,mm)=(xval(ii)+yval(jj)+zval(kk))*tval(ll)*tevl(mm)
```

```
        *enddo
      *enddo
    *enddo
  *enddo
*enddo

/prep7
block,,wid,,hth,,len          !!!! create beam volume
et,1,5                        !!!! use SOLID5

esize,0.5                     !!!! element size
mshkey,1                      !!!! mapped mesh
vmesh,all

mp,ex,1,1e7                   !!!! material properties
mp,nuxy,1,.3
mp,kxx,1,1

nsel,s,loc,z,0                !!!! fix end of beam
d,all,all
fini
save                          !!!! save problem for future restart
/solu
antyp,trans
timint,off

asel,u,loc,z,0
sfa,all,1,pres,%ccc%          !!!! apply pressure to all selected areas
alls
time,1e-3                     !!!! first solution at time = "0"
nsub,1
outres,all,all                !!!! output everything to results file
d,all,temp,0                  !!!! for first problem, temp = 0
solve

time,30     !!!! second solution, time=30
d,all,temp,150    !!!! second solution, temp=150
solve
finish
/post1
/view,1,1,1,1
/psf,press,norm,3,0,1
/pbc,all,0
set,1,1
/title, Pressure distribution; time=0, temp=0
eplot
set,2,1
/title, Pressure distribution; time=30, temp=150
eplot
finish
```

The following plots illustrate the pressure distribution for the two load cases.

Figure 2.18 Pressure Distribution for Load Case 1
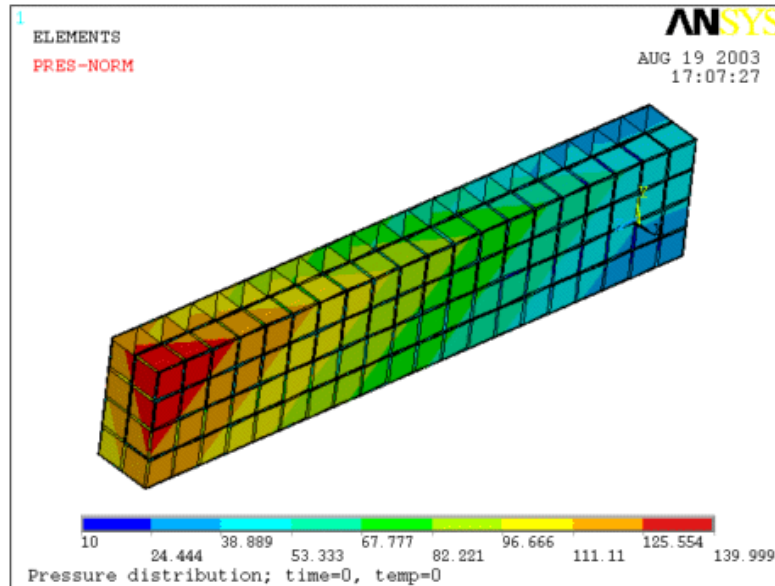
Figure 2.19 Pressue Distribution for Load Case 2
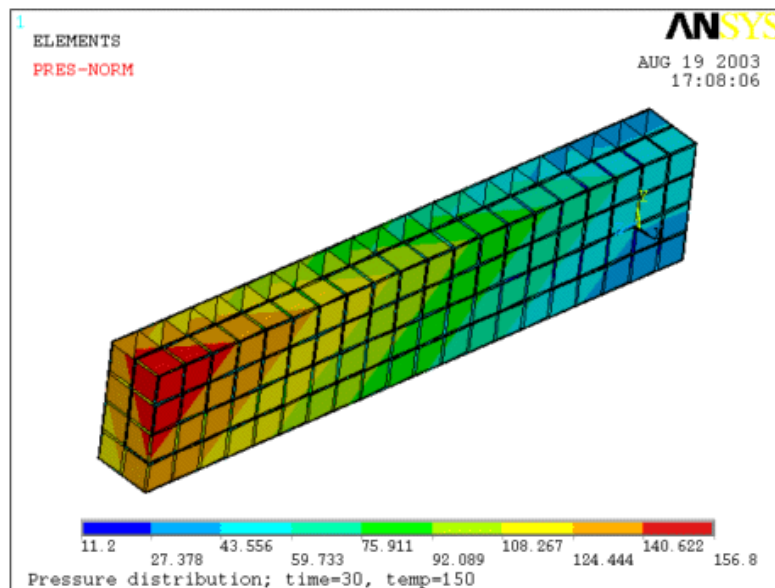
Note the difference in the pressure load in the second load case.

2.6.15. Applying Loads Using Function Boundary Conditions

You can use the Function Tool to apply complicated boundary conditions on a model. The Function Tool has two parts: the Function Editor creates an arbitrary equation or function (multiple equations), and a Function Loader retrieves the functions and loads them as TABLE arrays. They are then applied to a model using the tabular boundary condition process described in Section 2.6.14: Applying Loads Using TABLE Type Array Parameters.

You must define your function and load it as a TABLE array *before* you try to add it as a load. You can access the Function Editor either via **Utility Menu> Parameters> Functions> Define/Edit**, or via **Main Menu> Solution> Define Loads> Apply> Functions> Define/Edit**. You can access the Function Loader either via **Utility Menu>**

Parameters> Functions> Read from file, or via **Main Menu> Solution> Define Loads> Apply> Functions> Read file**

You should understand these additional points as you work with function boundary conditions:

- If your data can be conveniently expressed as a table, we recommend that you use tabular boundary conditions. For more information, see Section 2.6.14: Applying Loads Using TABLE Type Array Parameters.

- Functions are stored in a TABLE array in equation format, not as discrete table values.

- You cannot use function boundary conditions to circumvent the restrictions on boundary conditions and their corresponding primary variables as supported by tabular boundary conditions. For example, in a structural analysis, the primary variables supported with a pressure load are TIME, X, Y, Z, and TEMP; therefore, when using a function boundary condition, the only primary variables allowed in the equation are TIME, X, Y, Z, and TEMP. The list in Section 2.6.15.1: Using the Function Editor shows which primary variables are available for each type of operation.

You should be familiar with some specific terms when using the Function Tool.

- Function: A set of equations that together define an advanced boundary condition.

- Primary Variable: Independent variables evaluated and used by the program during solution.

- Regime: A portion of an operating range or design space characterized by a single regime variable. Regimes are partitioned according to lower and upper bounds of the regime variable. The regime variable must be continuous across the entire regime. Each regime contains a unique equation to evaluate the function.

- Regime Variable: The defining variable that governs which of the set of equations is used to evaluate the function.

- Equation Variable: A user-specified variable in an equation. The value for this variable is defined when the function is loaded.

2.6.15.1. Using the Function Editor

The Function Editor defines the regimes and equations. You use a set of primary variables, equation variables, and mathematical functions to build the equations. You can build a single equation or you can build a function, which is a series of equations, each of which applies to a particular regime. The equations defined for each regime, taken together, define the function, and the function as a whole is applied as the boundary condition. This process is described in detail in this section.

The function editor works like a calculator. To build the equation, you can:

- Select a primary variable from the drop-down list. The primary variables you can choose from are:

  - Time* (TIME)

  - X location* (X) in global Cartesian coordinates

  - Y location* (Y) in global Cartesian coordinates

  - Z location* (Z) in global Cartesian coordinates

  - Temperature* (TEMP degree of freedom)

  - Fluid temperature (TFLUID) (computed fluid temperature in FLUID116 elements for SURF151 or SURF152 elements)

  - Velocity* (VELOCITY) (magnitude of the Velocity degrees of freedom or the computed fluid velocity in FLUID116 elements)

- – Applied surface pressure* (PRES)

- – Tsurf* (TS) (element surface temperature for SURF151 or SURF152 elements)

- – Density (ρ) (material property DENS)

- – Specific heat (material property C)

- – Thermal conductivity (material property kxx)

- – Thermal conductivity (material property kyy)

- – Thermal conductivity (material property kzz)

- – Viscosity (material property μ)

- – Emissivity (material property ε)

- – Reference location* (Xr) (ALE formulations only)

- – Reference location* (Yr) (ALE formulations only)

- – Reference location* (Zr) (ALE formulations only)

- – Contact gap (GAP)

- – Rotational speed (OMEGS) (rotational speed for SURF151 or SURF152 elements)

- – Rotational speed (OMEGF) (rotational speed for FLUID116 elements)

- – Slip factor (SLIP) (slip factor for FLUID116 elements)

  Primary variables marked with an asterisk (*) are also available for tabular boundary conditions. The remaining primary variables can be used only with function boundary conditions.

- • Type any variable name into the Function Editor; it is interpreted as an equation variable. You can use any name you want, but we recommend that you not use the same name as one of the primary variables. You can use up to ten user-defined equation variables in a function (up to six regimes). You will define the values for these variables when you actually load the function (described in Section 2.6.15.2: Using the Function Loader).

- • Click a button on the keypad. The keypad includes the numbers 0-9, parentheses, and a set of mathematical operators. In addition to the default set of operators, you can also click the INV key to access an alternate set of operators.

As you build the equation, it is shown in standard mathematical syntax in the equation box above the keypad. The various components (primary variables, equation variables, mathematical operators, and numbers) are shown in different colors. In this way, you can verify the equation you are entering. You can also graph or list the equation using the GRAPH/LIST button in the Function Editor dialog box. See Section 2.6.15.4.1: Graphing BC Functions for details on this feature.

Note — The Function Editor DOES NOT validate the equation construction. You are responsible for ensuring the mathematical validity of any equation you define and that parentheses, operators, etc., are correctly posed. For example, you must ensure that you do not inadvertently create a divide by zero situation.

If you intend to use an equation or part of an equation later in the function (such as in another regime), you can click the STO button to store it. When you click the STO button, the numbers on the keypad change to a series of memory buffers. Click one of them to store the equation in that memory buffer. For example, click STO and then M1 to store the equation in the Memory1 buffer. To retrieve a stored equation, click the INV button and then the INS MEM button followed by the appropriate memory button. The contents of that memory buffer are then displayed in the equation box. You can also recall an abbreviated form of the contents by clicking the RCL button. If you pause the cursor over a memory button, a tool tip displays the contents of that memory buffer.

You must define your function and load it as a TABLE array *before* you try to add it as a load.

A detailed procedure for using the Function Editor follows.

1. Open the Function Editor by choosing **Utility Menu> Parameters> Functions> Define/Edit** or **Main Menu> Solution> Define Loads> Apply> Functions> Define/Edit**.

2. Select the function type. Choose either a single equation or a multivalued function. If you choose the latter, you must type in the name of your regime variable. This is the variable that governs the equations in the function. When you select a multivalued function, the six regime tabs become active.

3. Choose degrees or radians. This selection determines only how the equation is evaluated. This setting has no effect on ***AFUN** settings.

4. Define the result equation (if a single equation) or the equation describing the regime variable (if a multivalued function) using primary variables, equation variables, and the keypad. If you are defining a single-equation function, go to Step 10 to comment and save the equation. If you are defining a multi-valued function, continue with Step 5.

5. Click on the **Regime 1** tab. Type in the appropriate lower and upper limits for the regime variable you defined under the **Function** tab.

6. Define the equation for this regime.

7. Click on the **Regime 2** tab. Note that the lower limit for the regime variable is already defined and cannot be changed. This feature ensures that the regimes remain continuous, with no gaps. Define the upper limit for this regime.

8. Define the equation for this regime.

9. Continue this process for up to six regimes. You do not have to store or save the individual equations in each regime, unless you wish to reuse the equation in another regime.

10. Enter a comment to describe the function (optional). Choose **Editor> Comment** and type your comment in the area provided.

11. Save the function. Choose **Editor> Save** and type in a name. The filename must have a **.func** extension.

Once the function is defined and saved, it can be used in any applicable ANSYS analysis, or by any user with access to the file. For example, you could create a corporate library of functions and place them in a common directory that all users can access via a network.

To use the function, you must load it, assign values to any equation variables, and provide a table parameter name for use in a particular analysis. All of these tasks are done with the Function Loader.

2.6.15.2. Using the Function Loader

When you are ready to apply specific values to the equation variables, specify a table parameter name, and use the function in an analysis, you need to load the function into the Function Loader.

1. Open the Function Loader by choosing **Utility Menu> Parameters> Functions> Read from file**.

2. Navigate to the directory where you saved the function, select the appropriate file, and open it.

3. In the Function Loader dialog box, enter a table parameter name. This is the name you will use (%tab-name%) when you specify this function as a tabular boundary condition.

4. On the bottom half of the dialog box, you will see a **Function** tab and a **Regime** tab for each regime defined for the function. Click on the **Function** tab. You will see a data entry area for each equation variable you specified. You will also see a data entry area for material IDs if you used any variable that requires a material ID. Enter the appropriate values in these data entry areas.

Note — Only numeric data is supported for the constant values in the Function Loader dialog box. Character data and expressions are not supported as constant values.

5. Repeat the process for each regime you defined.

6. Click on Save. You will not be able to save this as a TABLE array parameter until you have provided values for all variables in all regimes in the function.

Once you have saved the function as a named TABLE array parameter using the Function Loader, you can apply it as a tabular boundary condition. See Section 2.6.14: Applying Loads Using TABLE Type Array Parameters for detailed information on using tabular boundary conditions in your analysis.

Note — The function is loaded into the table as a coded equation. This coded equation is processed in ANSYS when the table is called for evaluation.

2.6.15.3. Example

The following example shows you how to create and apply a boundary condition using a function representation. In this example, the convection heat transfer coefficient from a fluid flowing over a flat plate is applied as a function boundary condition, using the correlation for laminar heat transfer coefficient. The figure below shows the flat plate with the applied boundary conditions.
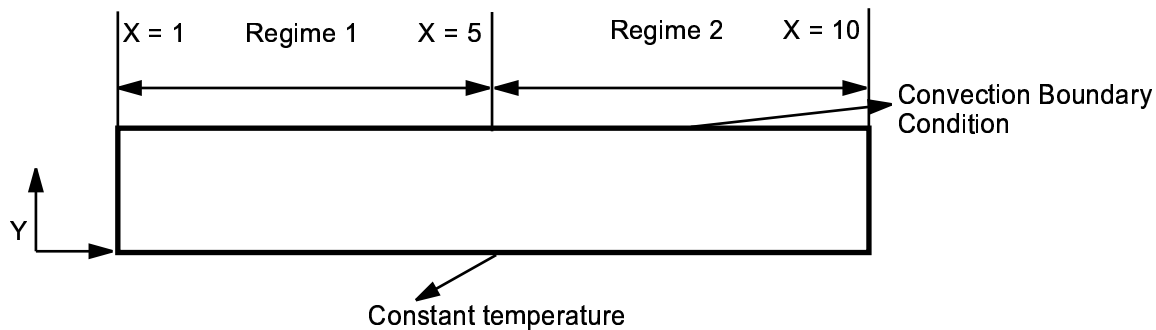
The bottom of the plate is fixed at a constant temperature. The top of the plate, where the convection boundary condition is being applied, is split into two regimes:

Regime 1 is defined for X between $1 \leq X < 5$, and the convection heat transfer coefficient is given by:

h(x) = 0.332 * (kxx/x) * Re**(1/2) * Pr**(1/3)

Regime 2 is defined for X between $5 < X \leq 10$, and the convection heat transfer coefficient is given by:

h(x) = 0.566 * (kxx/x) * Re**(1/2) * Pr**(1/3)

In the above equations, the Reynolds number Re is given by:

Re = (dens*vel*x)/visc

and the Prandtl number PR is given by:

Pr = (visc*c)/kxx

The properties of the fluid over the flat plate are:

Density (dens) = 1, thermal conductivity (kxx) = 10, specific heat (c) = 10, and viscosity (visc) = 0.01

The velocity of the fluid (vel) over the flat plate is equal to 100 for Regime 1 and 50 for Regime 2. Bulk temperature for the fluid for both regimes is 100 degrees.

1. First, create a rectangle and assign element type PLANE55, define your material properties, and mesh:

```
/prep7
rect,1,10,,.5
et,1,55
!Define Fluid Properties
mp,KXX,1.10 !Thermal conductivity
mp,DENS,1,1 !Density
mp,C,1,10 !Specific heat
mp,VISC,1,0.01 !Viscosity
!Define Plate Properties
mp,kxx,2,10
mp,dens,2,10
mp,c,2,5
mat,2
esize,,25
amesh,all
```

Next, you will define the convection boundary condition as a function. This is a two step process. First you define the function using the function editor, and then you apply the function as a table parameter using the function loader.

2. Define the function. Choose **Utility Menu> Parameters> Functions> Define/Edit** to bring up the function editor. The function boundary condition being applied in this example is a multivalued function, its final value being dependent on the X location in the domain. In the Function Editor dialog box, click on the radio button for "Multivalued function based on regime variable" and type *xloc* as the name of the regime variable in the text entry box. The name xloc appears as the name of the regime variable. To define xloc, select "X" from the drop down box on the lower half of the dialog box. Your dialog box should look like this:
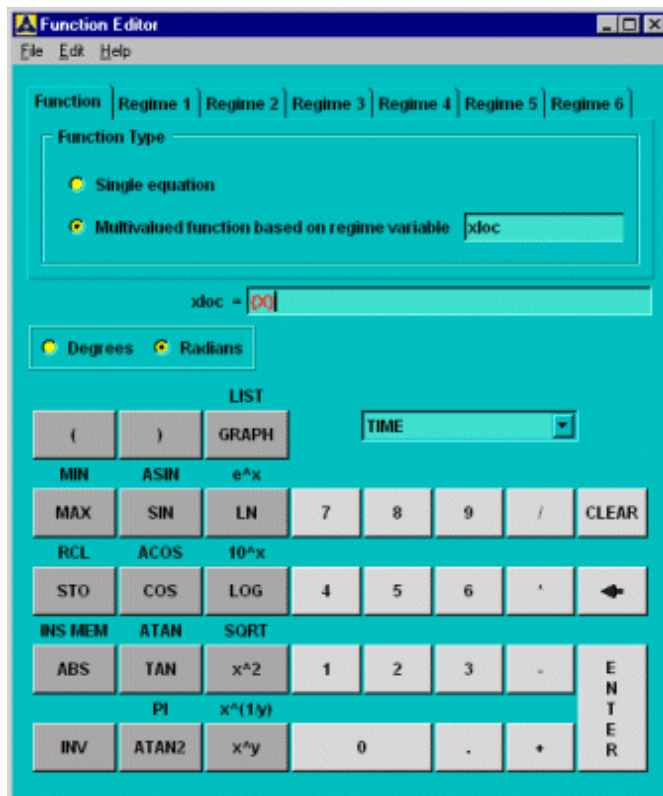
3. Now you will define the equations for the heat transfer coefficient in the two regimes. Click on the **Regime 1** tab. Under this tab, you will define the equation for the first regime, $1 \leq X \leq 5$. Type "1" and "5" in the Regime 1 Limits text entry boxes.

4. For ease of use, you may want to define those expressions in the equations that you will use more than once or that are part of a very long equation, and store them in memory. In this example, expressions for the Reynold's number and Prandtl number are used repeatedly in both equations; therefore, these are good examples of expressions that can be stored and used throughout the function editor, in all regimes.

    To store the Reynold's number, fill in the Result box as shown below. Select the terms DENS, X, and VISC (shown in {brackets}) from the drop down list on the lower half of the dialog box. Use the keypad to insert the math functions such as * and /. Your dialog box should look like this:
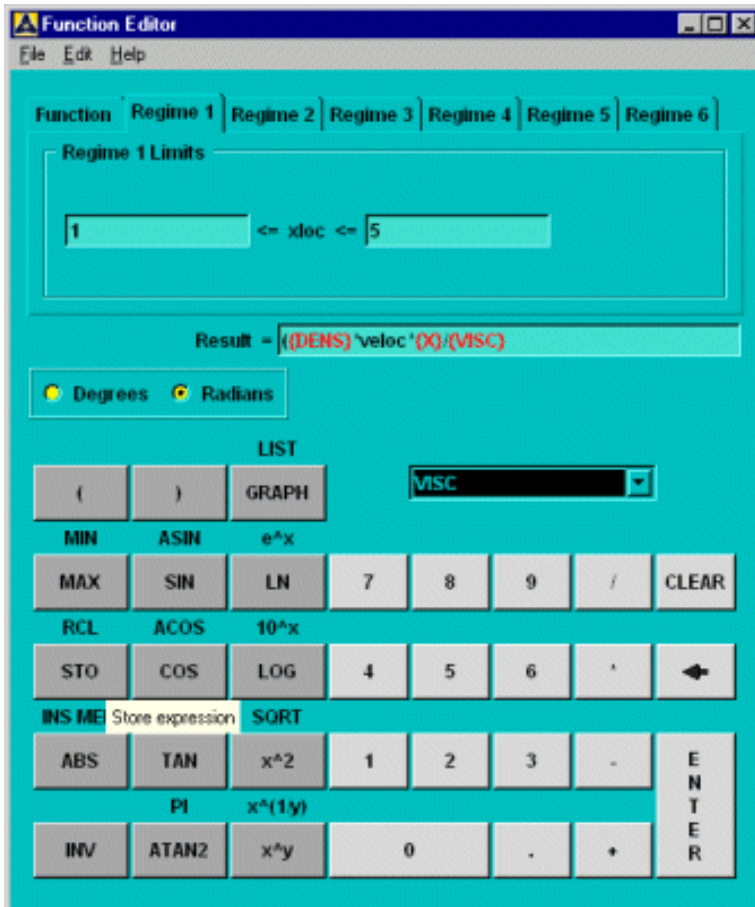
    Now click on **STO**, then on **M0** on the number pad to store the expression in memory location 0.

5. To store the Prandtl number, clear the Results box by clicking the **Clear** button and then fill it again as shown below. Select the terms VISC, SPHT, and KXX from the drop down list. Then, click on **STO** and then on the **M1** button. Your dialog box should look like this:
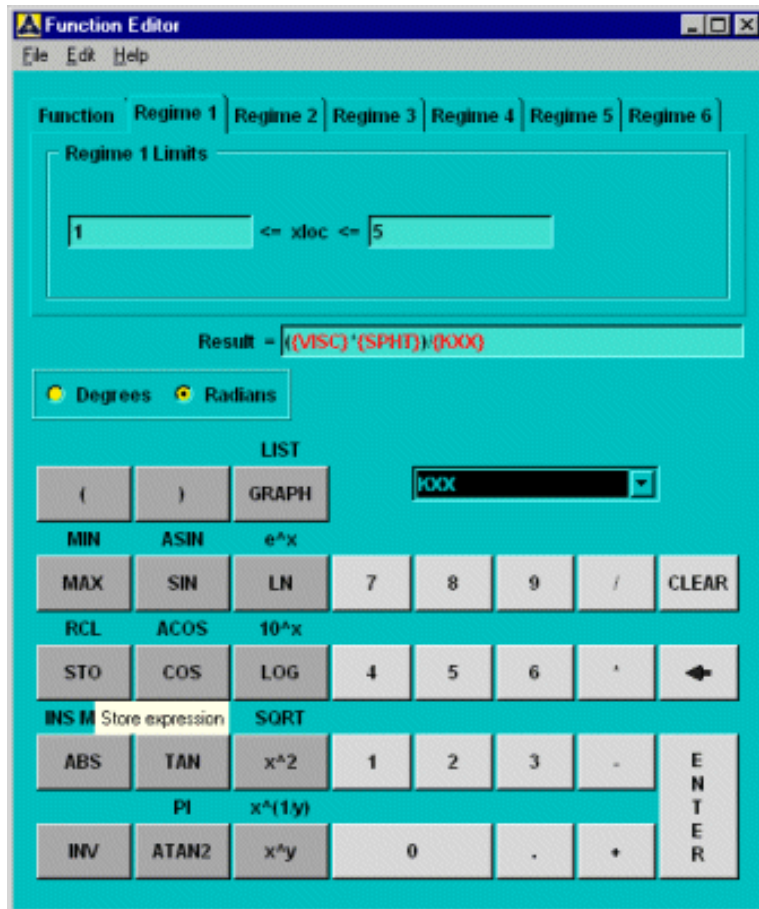
6. Now define an expression for the heat transfer coefficient for Regime 1. Click on the **Clear** button to clear the contents of the text entry box. Type in the expression for the heat transfer coefficient for Regime 1 as shown below. Items in brackets ({KXX} and {X}) are selected from the drop down list. The terms M0 and M1 are the terms you stored in memory earlier. To place them in the equation, click on the **INV** button, and then **RCL**, then **M0** and **M1** respectively.

7.   Now you will define the equation for Regime 2. Click on the **Regime2** tab. First, enter "10" as the upper limit for the regime variable for which this equation is valid. Notice that the lower limit for this regime is already set as the upper limit from Regime 1. This feature ensures continuity between the regimes. Next, type in the expression for the heat transfer coefficient as shown below. You can use the same stored memory locations M0 and M1 to replace expressions for Reynold's number and Prandtl number, respectively. Your dialog box should look like this:

8. You can now enter comments for this function. To enter comments, choose **File> Comments**.

9. Save the function by choosing **File> Save**. *You must save the function.* Function are saved with a **.func** extension.

   After you have saved the function, you can then load it as a table parameter into ANSYS.

10. Load the function and apply specific values to the equation variables. To do this, choose **Utility Menu> Parameters> Functions> Read from File**. Select the .func file that you saved in the previous step. The Function Loader dialog box appears.

11. Provide a table parameter name that you will use when applying the function as a boundary condition. For this example, type "heatcf" (this parameter name must be seven characters or less). You must then provide values for any variables you defined in the function editor. In the case of this example, the only variable is the velocity "veloc." Click on the **Regime 1** tab and enter "1" for the material ID and enter 100 for the velocity. You are asked the number for material ID only if you have used a material property in your expression. Your dialog box should look like this:
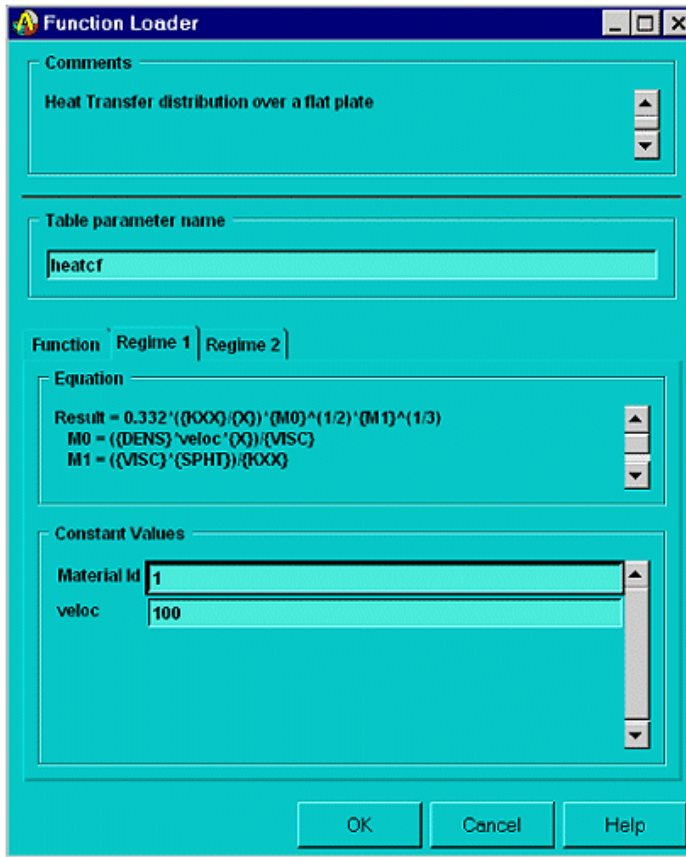
Note — Only numeric data is supported for the constant values in the Function Loader dialog box. Character data and expressions are not supported as constant values.

12. Click on the **Regime 2** tab and enter "1" for the material ID and enter 50 for the velocity. Notice that the **OK** button is not active until all required variables have been entered. Once the **OK** is active, click OK.

13. You can now finish the analysis. Use the table name you assigned when you apply this function as a boundary condition.

```
nsel,s,loc,y,0
d,all,temp,25
nsel,s,loc,y,0.5
sf,all,conv,%heatcf%,100    Apply the function as a boundary condition
finish
/solu
time,1
deltim,.1
outres,all,all
allsel
solve
finish
/post1
set,last
/psf,conv,hcoe,2,0.e+00,1
/replot !show surface load symbols
finish
```

2.6.15.4. Graphing or Listing the Boundary Condition Functions

You can graph the function you enter and see a visual representation of the current boundary condition function, or you can list results of the equation. These allow you to verify that your equation is behaving as you expect.

As shown in the figure below, for either graphing or listing, you need to choose a variable to graph the result against, and you must set an x-axis range and the number of points to graph.
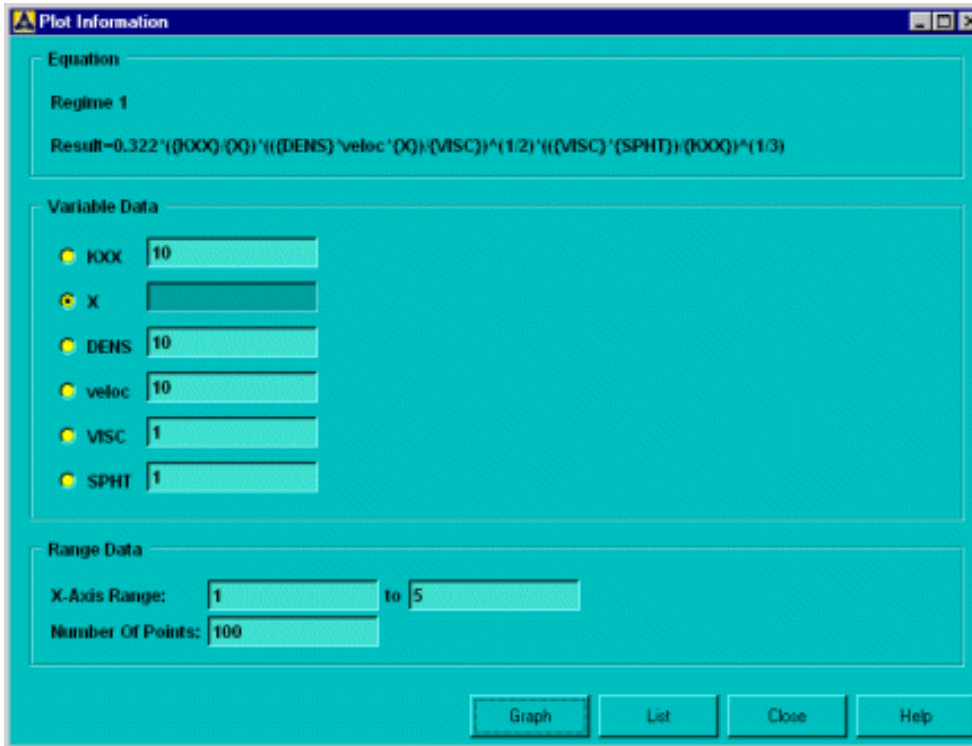
2.6.15.4.1. Graphing BC Functions

From the Plot Information dialog box, choose the Graph button after you set up your plot. An example of a plot is shown below.
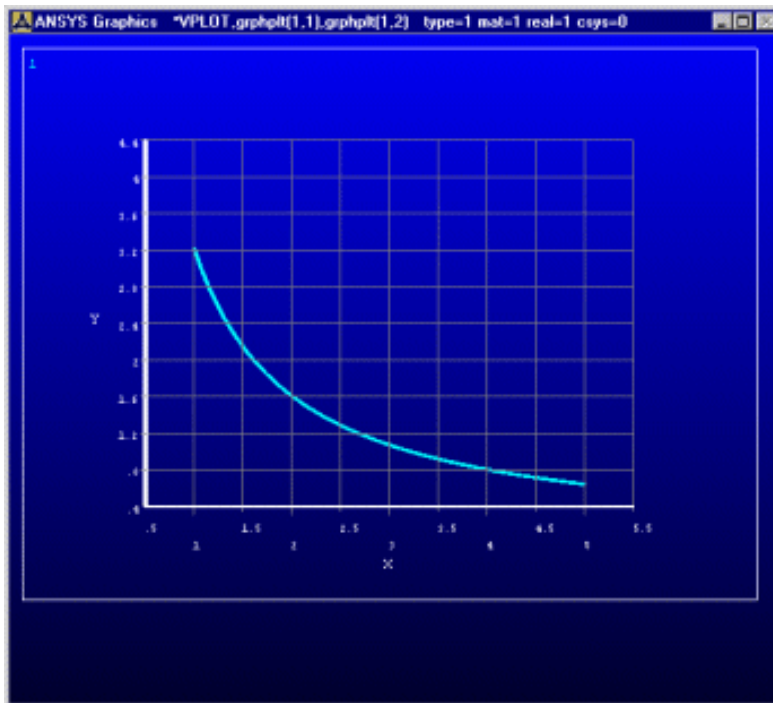
You can apply any of the standard graph functionality (for example, fill in under curve using the command input window or the **Utility Menu> PlotCtrls> Style> Graphs**). You can also save an image for later use (Image Capture).

2.6.15.4.2. Listing BC Functions

When you choose the List option from the Plot Information dialog box, ANSYS generates a table displaying the plot point values. The settings you chose in the Plot Information dialog box are used to generate these values. An example of a list is shown below.
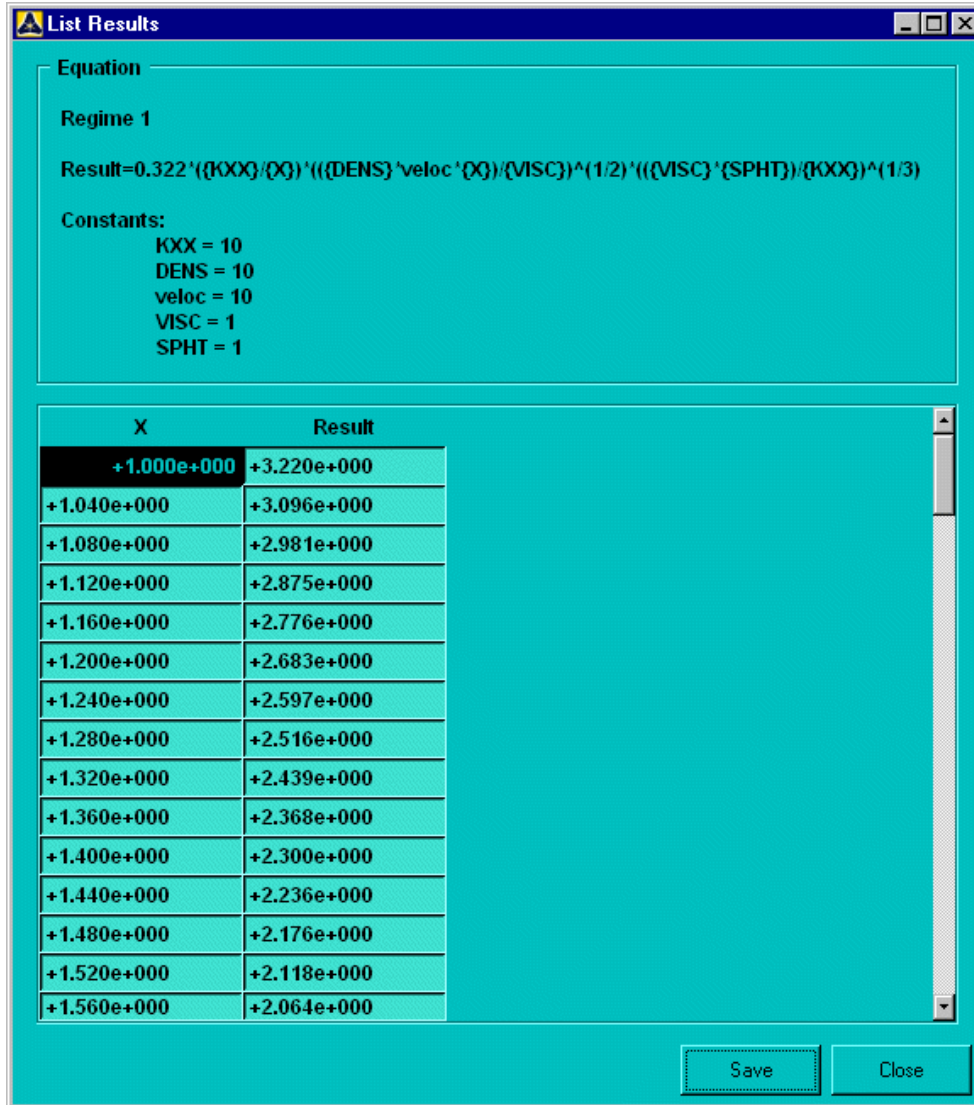
You cannot edit this table, but you can copy and past it into a spreadsheet if needed, or you can save the information to a text file which will contain all the equation data and the calculated coordinates.

2.7. How to Specify Load Step Options

As mentioned earlier, *load step options* is a collective name for options that control how loads are used during solution and other options such as output controls, damping specifications, and response spectrum data. Load step options can vary from load step to load step. There are six categories of load step options:

- General Options
- Dynamics Options

- Nonlinear Options

- Output Controls

- Biot-Savart Options

- Spectrum Options

2.7.1. General Options

These include such options as time at the end of a load step in transient and static analyses, number of substeps or the time step size, stepping or ramping of loads, and reference temperature for thermal strain calculations. A brief description of each option follows.

2.7.1.1. Solution Controls Dialog Box

If you are performing a static or full transient analysis, you can use the **Solution Controls** dialog box to set many of the load step options described on the following pages. Where applicable, the menu path to the **Solution Controls** dialog box is included. For details about using the **Solution Controls** dialog box, see Chapter 3, "Solution".

2.7.1.2. The Time Option

The **TIME** command specifies time at the end of a load step in transient and static analyses. In transient and other rate-dependent analyses, **TIME** specifies actual, chronological time, and you are required to specify a time value. In other, rate-independent analyses, time acts as a tracking parameter. You can never set time to zero in an ANSYS analysis. If you issue **TIME**,0 or **TIME**,(*blank*), or if you do not issue the **TIME** command at all, ANSYS uses the default time value: 1.0 for the first load step, and 1.0 + previous time for other load steps. To start your analysis at "zero" time, such as in a transient analysis, specify a very small value such as **TIME**,1E-6.

2.7.1.3. Number of Substeps and Time Step Size

For a nonlinear or transient analysis, you need to specify the number of substeps to be taken within a load step. This is done as follows:

Command(s):
    DELTIM

GUI:

    Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Time & Time Step
    Main Menu> Solution> Load Step Opts> Sol'n Control (: Basic Tab)
    Main Menu> Solution> Load Step Opts> Time/Frequenc> Time & Time Step
    Main Menu> Solution> Load Step Opts> Time/Frequenc> Time & Time Step

Command(s):
    NSUBST

GUI:

    Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Freq & Substeps (or Time and Substps)
    Main Menu> Solution> Load Step Opts> Sol'n Control (: Basic Tab)
    Main Menu> Solution> Load Step Opts> Time/Frequenc> Freq & Substeps (or Time and Substps)
    Main Menu> Solution> Unabridged Menu> Time/Frequenc> Freq & Substeps (or Time and Substps)

    ANSYS Basic Analysis Procedures Guide . ANSYS Release 8.1 . 001972 . © SAS IP, Inc.

NSUBST specifies the number of substeps, and **DELTIM** specifies the time step size. By default, the ANSYS program uses one substep per load step.

2.7.1.4. Automatic Time Stepping

The **AUTOTS** command activates automatic time stepping. Its equivalent GUI paths are:

GUI:

> **Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Time & Time Step (or Time and Substps)**
> **Main Menu> Solution> Load Step Opts> Sol'n Control (: Basic Tab)**
> **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time & Time Step (or Time and Substps)**
> **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time & Time Step (or Time and Substps)**

In automatic time stepping, the program calculates an optimum time step at the end of each substep, based on the response of the structure or component to the applied loads. When used in a nonlinear static (or steady-state) analysis, **AUTOTS** determine the size of load increments between substeps.

2.7.1.5. Stepping or Ramping Loads

When specifying multiple substeps within a load step, you need to indicate whether the loads are to be ramped or stepped. The **KBC** command is used for this purpose: **KBC**,0 indicates ramped loads, and **KBC**,1 indicates stepped loads. The default depends on the discipline and type of analysis.

> **Command(s): KBC**
> **GUI: Main Menu> Solution> Load Step Opts> Sol'n Control (: Transient Tab)**
> **Main Menu> Solution> Load Step Opts> Time/Frequenc> Freq & Substeps (or Time and Substps or Time & Time Step)**
> **Main Menu> Solution> Load Step Opts> Time/Frequenc> Freq & Substeps (or Time and Substps or Time & Time Step)**

Some notes about stepped and ramped loads are:

- If you specify stepped loads, the program handles all loads (constraints, forces, surface loads, body loads, and inertia loads) in the same manner. They are step-applied, step-changed, or step-removed, as the case may be.

- If you specify ramped loads, then:

  - All loads applied in the first load step, except film coefficients, are ramped (either from zero or from the value specified via **BFUNIF** or its GUI equivalent, depending on the type of load; see Table 2.13: "Handling of Ramped Loads (KBC = 0) Under Different Conditions"). Film coefficients are step-applied.

    Note — The concept of stepped versus ramped loading does *not* apply to temperature-dependent film coefficients (input as -*N* on a convection command). These are always applied at the value dictated by their temperature function.

  - All loads changed in later load steps are ramped from their previous values. Note that in a full harmonic analysis (**ANTYPE**,HARM with **HROPT**,FULL), surface and body loads ramp as they do in the first load step and *not* from their previous values, *except* for PLANE2, SOLID45, SOLID92, and SOLID95, which *do* ramp from their previous values.

  - For tabular boundary conditions, loads are never ramped but rather evaluated at the current time. If a load is specified using the tabular format for one load step and then changed to a non-tabular for

the next, the load is treated as a newly introduced load and ramped from zero or from **BFUNIF** and not from the previous tabular value.

– All loads newly introduced in later load steps are ramped (either from zero or from **BFUNIF**, depending on the type of load; see Table 2.13: "Handling of Ramped Loads (KBC = 0) Under Different Conditions").

– All loads deleted in later load steps are step-removed, except body loads and inertia loads. Body loads are ramped to **BFUNIF**. Inertia loads, which you can delete only by setting them to zero, are ramped to zero.

– Loads should not be deleted and respecified in the same load step. Ramping may not work the way the user intended in this case.

Table 2.13 Handling of Ramped Loads (KBC = 0) Under Different Conditions

Load Type	Applied in Load Step 1	Introduced in Later Load Steps
DOF Constraints		
Temperatures	Ramped from **TUNIF**[2]	Ramped from **TUNIF**[3]
Others	Ramped from zero	Ramped from zero
Forces	Ramped from zero	Ramped from zero
Surface Loads		
TBULK	Ramped from **TUNIF**[2]	Ramped from **TUNIF**
HCOEF	Stepped	Ramped from zero[4]
Others	Ramped from zero	Ramped from zero
Body Loads		
Temperatures	Ramped from **TUNIF**[2]	Ramped from previous **TUNIF**[3]
Others	Ramped from **BFUNIF**[5]	Ramped from previous **BFUNIF**[3]
Inertia Loads[1]	Ramped from zero	Ramped from zero

1. For OMEGA loads, note that the OMEGA itself is linearly ramped; therefore, the resulting force will vary quadratically over the load step.

2. The **TUNIF** command specifies a uniform temperature at all nodes.

3. In this case, the **TUNIF** or **BFUNIF** value from the *previous* load step is used, not the current value.

4. Temperature-dependent film coefficients are always applied at the value dictated by their temperature function, regardless of the **KBC** setting.

5. The **BFUNIF** command is just a generic form of **TUNIF**, meant to specify a uniform body load at all nodes.

2.7.1.6. Other General Options

You can also specify the following general options:

• The reference temperature for thermal strain calculations, which defaults to zero degrees. Specify this temperature as follows:

> **Command(s): TREF**
> **GUI: Main Menu> Preprocessor> Loads> Load Step Opts> Other> Reference Temp**
> **Main Menu> Preprocessor> Loads> Define Loads> Settings> Reference Temp**
> **Main Menu> Solution> Load Step Opts> Other> Reference Temp**
> **Main Menu> Solution> Define Loads> Settings> Reference Temp**

- Whether a new triangularized matrix is required for each solution (that is, each equilibrium iteration). You can do this only in a static (steady-state) or transient analysis, using one of these methods:

  > **Command(s): KUSE**
  > **GUI: Main Menu> Preprocessor> Loads> Load Step Opts> Other> Reuse Tri Matrix**
  > **Main Menu> Solution> Load Step Opts> Other> Reuse Tri Matrix**

  By default, the program decides whether a new matrix is required, based on such things as changes in DOF constraints, temperature-dependent material properties, and the Newton-Raphson option. If **KUSE** is set to 1, the program reuses the previous triangularized matrix. This setting is useful during a singleframe restart (it cannot be used during a multiframe restart). If you are restarting an analysis for additional load steps and you know that the existing triangularized matrix (in the file **Jobname.TRI**) can be reused, you can save a significant amount of computer time by setting **KUSE** to 1. The command **KUSE**,-1 forces the triangularized matrix to be reformulated at every equilibrium iteration. Analyses rarely require this; you will use it mainly for debugging purposes.

- A mode number (the number of harmonic waves around the circumference) and whether the harmonic component is symmetric or antisymmetric about the global X axis. When you use axisymmetric harmonic elements (axisymmetric elements with nonaxisymmetric loading), the loads are specified as a series of harmonic components (a Fourier series). To specify the mode number, use one of the following:

  > **Command(s): MODE**
  > **GUI: Main Menu> Preprocessor> Loads> Load Step Opts> Other> For Harmonic Ele**
  > **Main Menu> Solution> Load Step Opts> Other> For Harmonic Ele**

  See the *ANSYS Elements Reference* for a description of harmonic elements.

- The type of scalar magnetic potential formulation to be used in a 3-D magnetic field analysis, specified via one of the following:

  > **Command(s): MAGOPT**
  > **GUI: Main Menu> Preprocessor> Loads> Load Step Opts> Magnetics>** *potential formulation method*
  > **Main Menu> Solution> Load Step Opts> Magnetics>** *potential formulation method*

- The type of solution to be expanded in the expansion pass of a reduced analysis, specified via one of the following:

  > **Command(s): NUMEXP**, **EXPSOL**
  > **GUI: Main Menu> Preprocessor> Loads> Load Step Opts> ExpansionPass> Single Expand> Range of Solu's**
  > **Main Menu> Solution> Load Step Opts> ExpansionPass> Single Expand> Range of Solu's**
  > **Main Menu> Preprocessor> Loads> Load Step Opts> ExpansionPass> Single Expand> By Load Step**
  > **Main Menu> Preprocessor> Loads> Load Step Opts> ExpansionPass> Single Expand> By Time/Freq**
  > **Main Menu> Solution> Load Step Opts> ExpansionPass> Single Expand> By Load Step**
  > **Main Menu> Solution> Load Step Opts> ExpansionPass> Single Expand> By Time/Freq**

2.7.2. Dynamics Options

These are options used mainly in dynamic and other transient analyses. They include the following:

Table 2.14 Dynamic and Other Transient Analyses Commands

Command	GUI Menu Paths	Purpose
TIMINT	**Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Time Integration** **Main Menu> Solution> Load Step Opts> Sol'n Control (: Basic Tab)** **Main Menu> Solution> Load Step Opts> Time/Frequenc> Time Integration** **Main Menu> Solution> Unabridged Menu> Time/Frequenc> Time Integration**	Activates or deactivates time integration effects
HARFRQ	**Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Freq & Substeps** **Main Menu> Solution> Load Step Opts> Time/Frequenc> Freq & Substeps**	Specifies the frequency range of the loads in a harmonic response analysis
ALPHAD	**Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Damping** **Main Menu> Solution> Load Step Opts> Sol'n Control (: Transient Tab)** **Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping** **Main Menu> Solution> Unabridged Menu> Time/Frequenc> Damping**	Specifies damping for a structural dynamic analysis
BETAD	**Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Damping** **Main Menu> Solution> Load Step Opts> Sol'n Control (: Transient Tab)** **Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping** **Main Menu> Solution> Unabridged Menu> Time/Frequenc> Damping**	Specifies damping for a structural dynamic analysis
DMPRAT	**Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Damping** **Main Menu> Solution> Time/Frequenc> Damping**	Specifies damping for a structural dynamic analysis
MDAMP	**Main Menu> Preprocessor> Loads> Load Step Opts> Time/Frequenc> Damping** **Main Menu> Solution> Load Step Opts> Time/Frequenc> Damping**	Specifies damping for a structural dynamic analysis
TRNOPT	**Main Menu> Preprocessor> Loads> Analysis Type> Analysis Options** **Main Menu> Preprocessor> Loads> Analysis Type> New Analysis** **Main Menu> Solution> Analysis Type> Analysis Options** **Main Menu> Solution> Analysis Type> New Analysis**	Specifies transient analysis options

2.7.3. Nonlinear Options

These are options used mainly in nonlinear analyses. They include the following:

Table 2.15 Nonlinear Analyses Commands

Command	GUI Menu Paths	Purpose
NEQIT	**Main Menu> Preprocessor> Loads> Load Step Opts> Nonlinear> Equilibrium Iter** **Main Menu> Solution> Load Step Opts> Sol'n Control (: Nonlinear Tab)** **Main Menu> Solution> Load Step Opts> Nonlinear> Equilibrium Iter** **Main Menu> Solution> Unabridged Menu> Nonlinear> Equilibrium Iter**	Specifies the maximum number of equilibrium iterations per substep (default = 25)
CNVTOL	**Main Menu> Preprocessor> Loads> Load Step Opts> Nonlinear> Convergence Crit** **Main Menu> Solution> Load Step Opts> Sol'n Control (: Nonlinear Tab)** **Main Menu> Solution> Load Step Opts> Nonlinear> Convergence Crit** **Main Menu> Solution> Unabridged Menu> Nonlinear> Convergence Crit**	Specifies convergence tolerances
NCNV	**Main Menu> Preprocessor> Loads> Load Step Opts> Nonlinear> Criteria to Stop** **Main Menu> Solution> Sol'n Control (: Advanced NL Tab)** **Main Menu> Solution> Load Step Opts> Nonlinear> Criteria to Stop** **Main Menu> Solution> Unabridged Menu> Nonlinear> Criteria to Stop**	Provides options for terminating analyses

2.7.4. Output Controls

Output controls, as their name indicates, control the amount and nature of output from an analysis. There are two primary output controls:

Table 2.16 Output Controls Commands

Command	GUI Menu Paths	Purpose
OUTRES	**Main Menu> Preprocessor> Loads> Load Step Opts> Output Ctrls> DB/Results File** **Main Menu> Solution> Load Step Opts> Sol'n Control (: Basic Tab)** **Main Menu> Solution> Load Step Opts> Output Ctrls> DB/Results File** **Main Menu> Solution> Load Step Opts> Output Ctrls> DB/Results File**	Controls what ANSYS writes to the database and results file and how often it is written.
OUTPR	**Main Menu> Preprocessor> Loads> Load Step Opts> Output Ctrls> Solu Printout** **Main Menu> Solution> Load Step Opts> Output Ctrls> Solu Printout** **Main Menu> Solution> Load Step Opts> Output Ctrls> Solu Printout**	Controls what is printed (written to the solution output file, **Jobname.OUT**) and how often it is written.

The example below illustrates using **OUTRES** and **OUTPR**:

```
OUTRES,ALL,5    ! Writes all data every 5th substep
OUTPR,NSOL,LAST  ! Prints nodal solution for last substep only
```

You can issue a series of **OUTPR** and **OUTRES** commands (up to 50 of them combined) to meticulously control the solution output, but be aware that the order in which they are issued is important. For example, the commands shown below will write all data to the database and results file every 10th substep and nodal solution data every fifth substep.

```
OUTRES,ALL,10
OUTRES,NSOL,5
```

However, if you reverse the order of the commands (as shown below), the second command essentially overrides the first, resulting in all data being written every 10th substep and nothing every 5th substep.

```
OUTRES,NSOL,5
OUTRES,ALL,10
```

As another example,

```
OUTRES,NSOL,10
OUTRES,NSOL,ALL,TIP
```

writes the solution at all DOFs every 10th substep and the solution at the node component "TIP" every substep. Again, if you reverse these you will only obtain output at all DOF every 10th substep.

> *Note* — The program default for writing out solution data for all elements depends on analysis type; see the description of **OUTRES** in the *ANSYS Elements Reference*. To restrict the solution data that is written out, use **OUTRES** to selectively suppress (FREQ = NONE) the writing of solution data, or first suppress the writing of all solution data (**OUTRES**,ALL,NONE) and then selectively turn on the writing of solution data with subsequent **OUTRES** commands.

A third output control command, **ERESX**, allows you to review element integration point values in the postprocessor.

> **Command(s): ERESX**
> **GUI: Main Menu> Preprocessor> Loads> Load Step Opts> Output Ctrls> Integration Pt**
> **Main Menu> Solution> Load Step Opts> Output Ctrls> Integration Pt**
> **Main Menu> Solution> Load Step Opts> Output Ctrls> Integration Pt**

By default, the ANSYS program extrapolates nodal results that you review in the postprocessor from integration point values for all elements except those with active material nonlinearities (for instance, nonzero plastic strains). By issuing **ERESX**,NO, you can turn off the extrapolation and instead *copy* integration point values to the nodes, making those values available in the postprocessor. Another option, **ERESX**,YES, forces extrapolation for *all* elements, whether or not they have active material nonlinearities.

2.7.5. Biot-Savart Options

These are options used in a magnetic field analysis. The two commands in this category are as follows:

Table 2.17 Biot-Savart Commands

Command	GUI Menu Paths	Purpose
BIOT	**Main Menu> Preprocessor> Loads> Load Step Opts> Magnetics> Options Only> Biot-Savart** **Main Menu> Solution> Load Step Opts> Magnetics> Options Only> Biot-Savart**	Calculates the magnetic source field intensity due to a selected set of current sources.
EMSYM	**Main Menu> Preprocessor> Loads> Load Step Opts> Magnetics> Options Only> Copy Sources** **Main Menu> Solution> Load Step Opts> Magnetics> Options Only> Copy Sources**	Duplicates current sources that exhibit circular symmetry.

The *ANSYS Low-Frequency Electromagnetic Analysis Guide* explains the use of these commands where appropriate.

2.7.6. Spectrum Options

There are many commands in this category, all meant to specify response spectrum data and power spectral density (PSD) data. You use these commands in spectrum analyses, as described in the *ANSYS Structural Analysis Guide*.

2.8. Creating Multiple Load Step Files

All loads and load step options put together form a *load step*, for which the program can calculate the solution. If you have multiple load steps, you can store the data for each load step on a file, called the *load step file*, and read it in later for solution.

The **LSWRITE** command writes the load step file (one file per load step, identified as **Jobname.S01**, **Jobname.S02**, **Jobname.S03**, etc.). Use one of these methods:

> **Command(s): LSWRITE**
> **GUI: Main Menu> Preprocessor> Loads> Load Step Opts> Write LS File**
> **Main Menu> Solution> Load Step Opts> Write LS File**

If you are using the **Solution Controls** dialog box to set your analysis and load step options, you define each load step using the **Basic** tab. (You can use the **Solution Controls** dialog box for static and full transient analyses only. For details, see Chapter 3, "Solution".)

After all load step files are written, you can use one action command to read in the files sequentially and obtain the solution for each load step (see Chapter 3, "Solution").

The sample set of commands shown below defines multiple load steps:

```
/SOLU                ! Enter SOLUTION
0
!  Load Step 1:
D, ...        ! Loads
SF, ...
 ...
NSUBST, ...          ! Load step options
KBC, ...
OUTRES, ...
OUTPR, ...
 ...
LSWRITE          ! Writes load step file: Jobname.S01

!  Load Step 2:
D, ...        ! Loads
SF, ...
 ...
NSUBST, ...          ! Load step options
KBC, ...
OUTRES, ...
OUTPR, ...
 ...
LSWRITE          ! Writes load step file: Jobname.S02
0
```

See the *ANSYS Commands Reference* for descriptions of the **NSUBST**, **KBC**, **OUTRES**, **OUTPR**, and **LSWRITE** commands.

Some notes about the load step file:

- The load step data are written to the file in terms of ANSYS commands.

- The **LSWRITE** command does not capture changes to real constants (**R**) or material properties (**MP**).

- The **LSWRITE** command automatically transfers solid-model loads to the finite element model, so all loads are written in the form of finite-element load commands. In particular, surface loads are always written in terms of **SFE** (or **SFBEAM**) commands, regardless of how they were applied.

- To modify data on load step file number *n*, issue the command **LSREAD**,*n* to read in the file, make the desired changes, and then issue **LSWRITE**,*n* (which will overwrite the old file *n*). You can also directly edit the load step file using your system editor, but this is generally not recommended. The GUI equivalents of the **LSREAD** command are:

  GUI:

     Main Menu> Preprocessor> Loads> Load Step Opts> Read LS File
     Main Menu> Solution> Load Step Opts> Read LS File

- The **LSDELE** command allows you to delete load step files from within the ANSYS program. The GUI equivalents of **LSDELE** are:

  GUI:

     Main Menu> Preprocessor> Loads> Define Loads> Operate> Delete LS Files
     Main Menu> Solution> Define Loads> Operate> Delete LS Files

- Another useful load step related command is **LSCLEAR**, which allows you to delete all loads and reset all load step options to their defaults. You can use it, for example, to "clean up" the load step data before reading in a load step file for modifications.

GUI equivalents for **LSCLEAR** are:

GUI:

   Main Menu> Preprocessor> Loads> Define Loads> Delete> All Load Data> *data type*
   Main Menu> Preprocessor> Loads> Reset Options
   Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs Add
   Main Menu> Solution> Reset Options
   Main Menu> Solution> Define Loads> Settings> Replace vs Add> Reset Factors

2.9. Defining Pretension in a Joint Fastener

Preloads in bolts and other structural components often have significant effect on deflections and stresses. Two ANSYS features, the PRETS179 pretension element and the **PSMESH** pretension meshing command, can be used for this type of analysis. If the fastener has been meshed in two separate pieces, the pretension elements can be inserted between the pieces using the **EINTF** command.

The pretension load is used to model a pre-assembly load in a joint fastener. The fastener can be made up of any 2-D or 3-D structural, low- or high-order solid, beam, shell, pipe, or link elements. When using the **PSMESH** command, the pretension section, across which the pretension load is applied, must be defined inside the fastener (shown in Figure 2.20: "Pretension Definition" for a bolted joint).

2.9.1. Applying Pretension to a Fastener Meshed as a Single Piece

The easiest way to apply pretension elements to a fastener is via the **PSMESH** command. You can use the command only if the fastener is *not* meshed in separate pieces. The command defines the pretension section and generates

the pretension elements. It automatically cuts the meshed fastener into two parts and inserts the pretension elements. If you decide that you want to remove the pretension elements, they can do so automatically by deleting the pretension section (**Main Menu> Preprocessor> Sections> Delete Section**). This feature also allows you to "undo" the cutting operation by merging nodes.

Figure 2.20 Pretension Definition
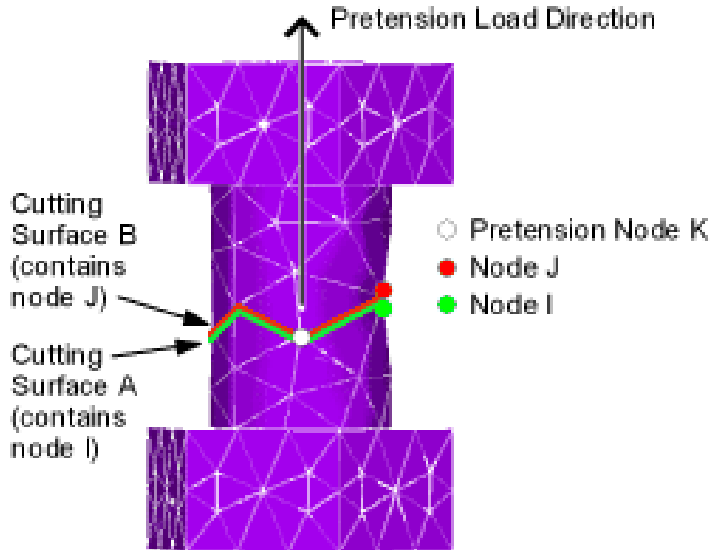
The normal direction is specified via the **PSMESH** command and is part of the section data. This is in contrast to the previous method (the **PTSMESH** command), which used real constants to specify the normal direction.

The meshed pretension section does not need to be flat. The elements underlying the pretension section can have almost any shape: line, triangle, quadrilateral, tetrahedron, wedge, or hexahedron. However, there must be coincident nodes on the two sides (A and B) of the pretension section. Sides A and B on the pretension section are connected by one or more pretension elements, one for each coincident node pair.

A pretension node (K) is used to control and monitor the total tension loads. The pretension load direction of the pretension section can be specified relative to side A when the section is created by the **PSMESH** command. All pretension elements on a specific pretension section must use the same section, and must have the same pretension node K. Node K is the third position for the pretension element definition.

2.9.2. Applying Pretension to a Fastener Meshed as Two Pieces

If the fastener has been meshed in two separate pieces (such as in an existing, legacy model), the pretension elements (PRETS179) can be inserted between the pieces using **EINTF**,*TOLER*,*K* (**Main Menu> Preprocessor> Create> elements> pretension> At Coincident Nodes ...**). If K is not defined, ANSYS will create it automatically. Before using the **EINTF** command, the element type ID and section properties must be defined properly. The connecting surfaces (A and B) must have matching mesh patterns with coincident nodes. If some node pairs between the two surfaces are not connected with pretension elements, the resulting analysis can be inaccurate.

2.9.3. Example Pretension Analysis

The following example describes the typical procedure used to perform a pretension analysis using the **PSMESH** command.

1. Mesh the bolt joint, then cut the mesh and insert the pretension elements to form the pretension section. For example, the following creates a pretension section called "example" by cutting the mesh and inserting the section into volume 1. Note that a component is created as well (npts) that aids in plotting or selecting the pretension elements.

   ```
   psmesh,,example,,volu,1,0,z,0.5,,,,npts
   ```

2. In the first load step, apply a force or displacement to node K. In this case, the load is applied as a force. The force "locks" on the second load step, allowing you to add additional loads. The effect of the initial load is preserved as a displacement after it is locked. This is shown in the following example.

   ```
   sload,1,9,tiny,forc,100,1,2
   ```

3. Apply other external loads as required using the **SLOAD** command.

The following example will help you to understand how the pretension procedure works.

Figure 2.21 Initial Meshed Structure


```
Sample application of PSMESH
```
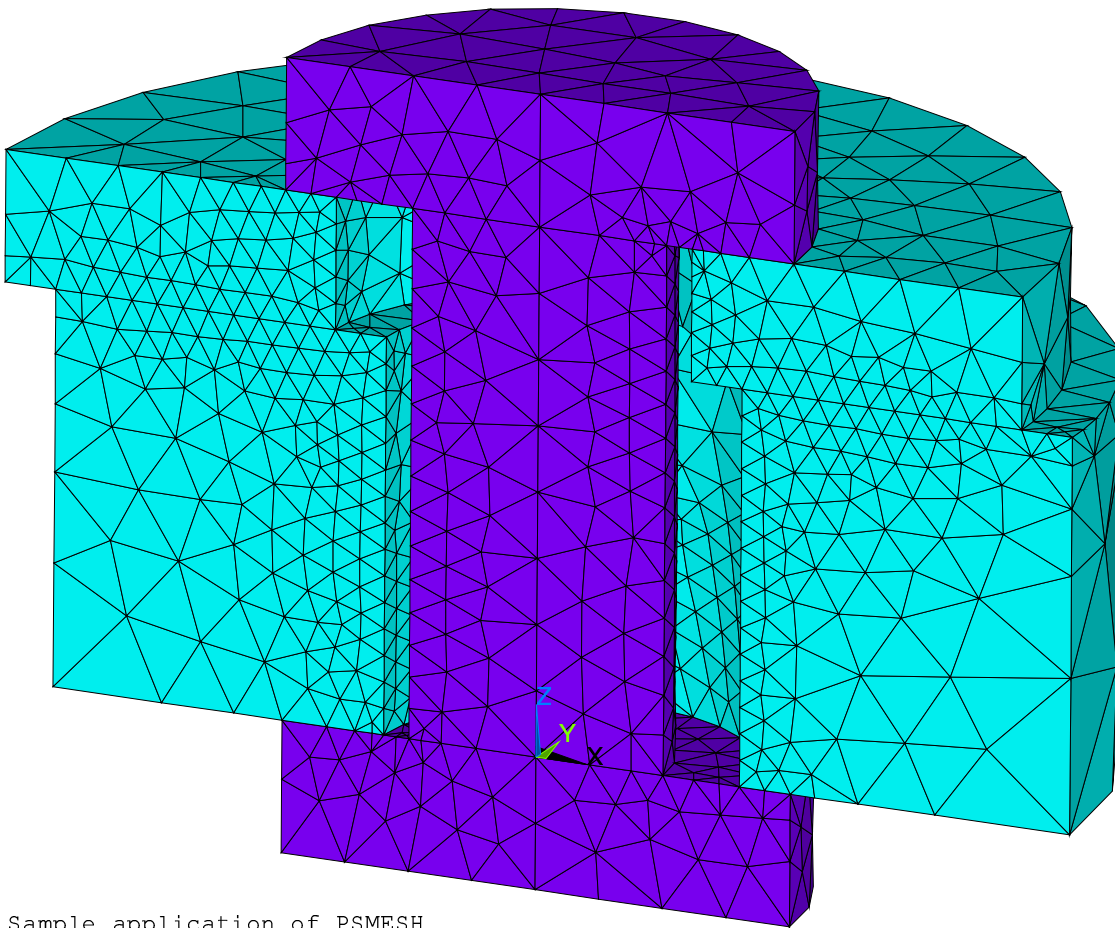
The model represents a 180° slice of two annular plates and a single bolt assembled with an offset. The bolt is carbon steel, and the plates are aluminum. (See Figure 2.21: "Initial Meshed Structure".)

Figure 2.22 Pretension Section
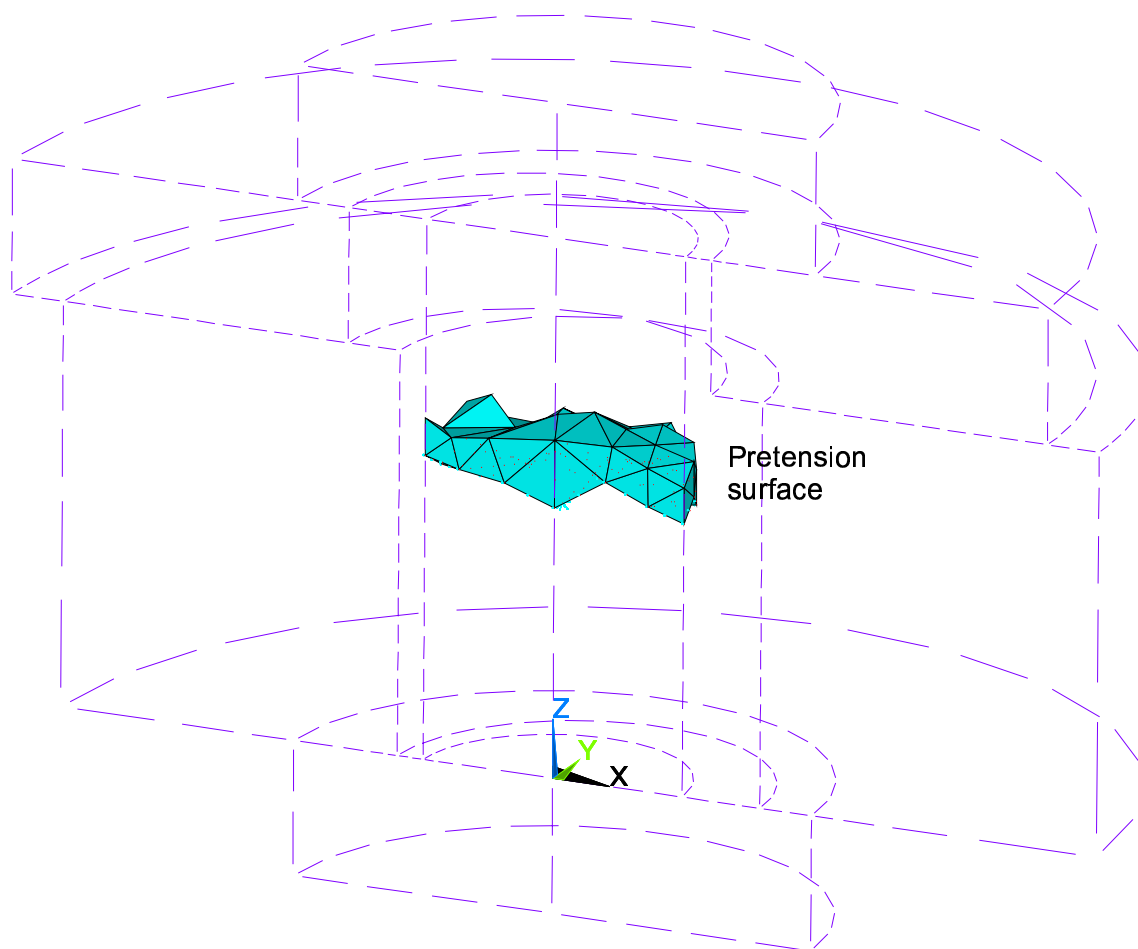
Sample application of PSMESH

We use the **PSMESH** operation to separate the elements of the bolt into two unconnected groups, tied together with PRETS179 pretension elements. We then plot the element and node components on the pretension interface. (See Figure 2.22: "Pretension Section".)

Figure 2.23 Pretension Stress
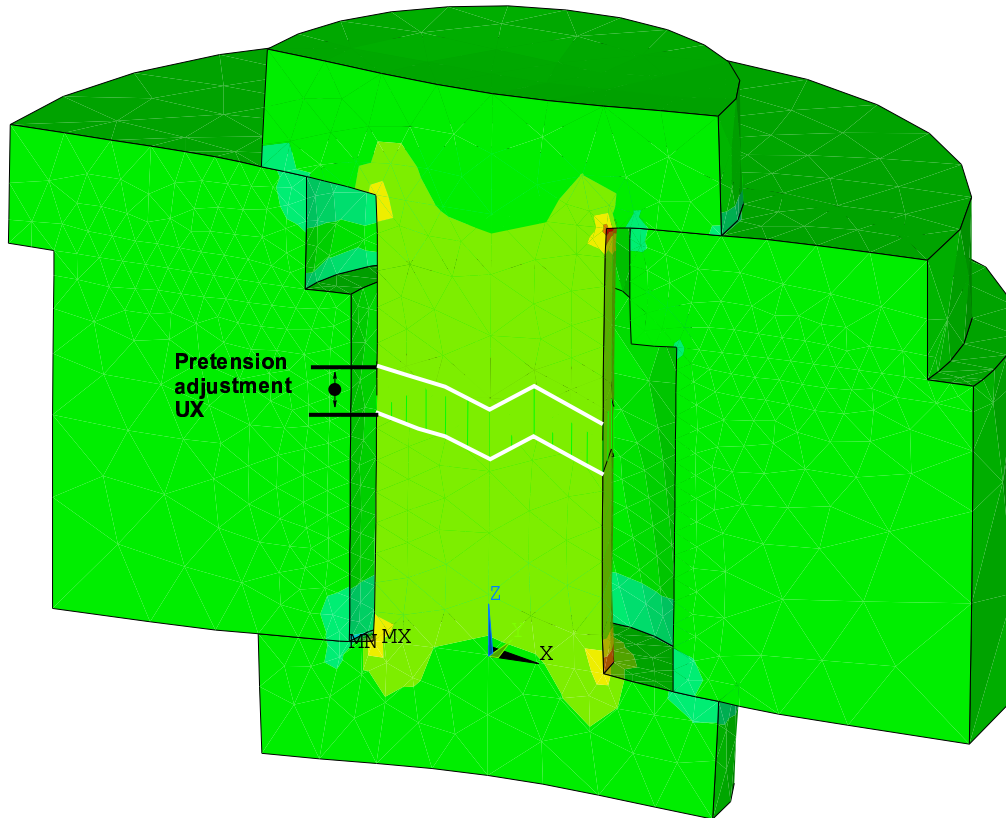

```
Sample application of PSMESH - preload only
```

We apply constraints for symmetry and to prevent rigid body motion. Note that the uniform temperature defaults to the reference temperature of 70°F. We apply half the load (this is a half model) to the pretension node created by **PSMESH**, solve, and plot the normal stress in the axial direction. As we should expect, the axial stress is tensile in the bolt, and compressive in the portion of the plates compressed by the bolt heads. (See Figure 2.23: "Pretension Stress".)

```
/prep7
/title, Sample application of PSMESH

et,1,92
mp,ex,1,1e7
mp,alpx,1,1.3e-5
mp,prxy,1,0.30
mp,ex,2,3e7
mp,alpx,2,8.4e-6
mp,prxy,2,0.30
tref,70

/foc,,-.09,.34,.42
/dist,,.99
/ang,,-55.8
/view,,.39,-.87,.31
/pnum,volu,1
/num,1
cylind,0.5,,  -0.25,0,  0,180
cylind,0.5,,  1,1.25,   0,180
cylind,0.25,, 0,1,      0,180
wpoff,.05
cylind,0.35,1, 0,0.75, 0,180
wpoff,-.1
```

```
cylind,0.35,1, 0.75,1, 0,180
wpstyle,,,,,,,,0
vglue,all
numc,all
vplot
mat,1
smrt,off
vmesh,4,5
mat,2
vmesh,1,3
/pnum,mat,1
eplot
psmesh,,example,,volu,1,0,z,0.5,,,,elems
CM,lines,LINE
/dist,,1.1
cmplot
/solu
eqslve,pcg,1e-8
asel,s,loc,y
da,all,symm
asel,all
dk,1,ux
dk,12,ux
dk,1,uz
sload,1,9,,force,100,1,2
/title,Sample application of PSMESH - preload only
solve
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Finally, we construct the actual solution of interest. We want to
!know what happens to the preload in the bolt, and the stress field around
!it, when the assembly temperature rises to 150° F.
!Both the preload and the stresses increase because, for a uniform
!temperature rise, there is greater thermal expansion in the aluminum plates
!than in the steel bolt. Any method for applying preload that did not
!allow the load to change would be unable to predict this result.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
/post1
plnsol,s,z
/solu
antype,,restart
tunif,150
/title,Sample application of PSMESH - uniform 150°
solve

/post1
plnsol,s,z
```

2.9.4. Example Pretension Analysis (GUI Method)

This section presents a sample pretension analysis using the ANSYS GUI.

2.9.4.1. Set the Analysis Title

1. Choose **Utility Menu> File> Change Title**

2. Enter the text, "Sample Application of PSMESH" and click **OK**.

2.9.4.2. Define the Element Type

Define SOLID92 as the element type.

1. Select **Main Menu> Preprocessor> Element Type> Add/Edit/Delete**. The Element Types dialog box appears.

2. Click **Add**. The Library of Elements dialog box appears.

3. In the scroll box on the left, select Structural, Solid.

4. Select Tet 10 node 92 in the scroll box on the right and click **OK**.

5. Click **Close** in the Element Types dialog box.

2.9.4.3. Define Material Properties

1. Select **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 Structural, Linear, Elastic, and Isotropic. A dialog box appears.

3. Enter 1E7 for EX, 0.3 for PRXY and click **OK**. Linear Isotropic appears under Material Model Number 1 in the Material Models Defined window.

4. Under Structural in the Material Models Available window, double click on Thermal Expansion Coef, Isotropic. A dialog box appears.

5. Enter 1.3E-5 for ALPX and click **OK**. Thermal Expansion (iso) appears under Material Model Number 1 in the Material Models Defined window.

6. Choose **Material> New Model**, then enter 2 for the new material ID and click **OK**. Material Model 2 appears in the Material Models Defined window on the left.

7. Double click on Isotropic under Structural, Linear, Elastic in the Material Models Available window. A dialog box appears.

8. Enter 3E7 for EX, 0.3 for PRXY and click **OK**. Linear Isotropic appears under Material Model Number 2 in the Material Models Defined window.

9. Double click on Isotropic under Structural, Thermal Expansion Coef in the Material Models Available Window. A dialog box appears.

10. Enter 8.4E-6 for ALPX and click **OK**. Thermal Expansion (iso) appears under Material Model Number 2 in the Material Models Defined window.

11. Choose **Material> Exit** to close the Define Material Behavior dialog box.

12. Select **Main Menu> Preprocessor> Loads> Define Loads> Settings> Reference Temp**.

13. Enter 70 as the reference temperature and click **OK**.

2.9.4.4. Set Viewing Options

1. Select **Utility Menu> PlotCtrls> View Settings> Focus Point**. The Focus Point dialog box appears.

2. Choose User Specified.

3. Enter -.09, .34, and .42 as the User specified locate and click **OK**.

4. Select **Utility Menu> PlotCtrls> View Settings> Magnification**. The Magnification dialog box appears

5. Choose User Specified.

6. Enter .99 as the User specified distance and click **OK**.

7. Select **Utility Menu> PlotCtrls> View Settings> Angle of Rotation**. The Angle of Rotation dialog box appears.

8. Enter -55.8 as the Angle in degrees value and click **OK**.

9. Select **Utility Menu> PlotCtrls> View Settings> Viewing Direction**. The Viewing Direction dialog box appears.

10. Enter .39, -.87, and .31 as the XV, YV, and ZV values, respectively and click **OK**.

11. Select **Utility Menu> PlotCtrls> Numbering**. Turn on Volume numbers.

12. Choose Numbering shown with Colors only and click **OK**.

2.9.4.5. Create Geometry

1. Select **Main Menu> Preprocessor> Modeling> Create> Volumes>** *By Dimensions*. The Create Cylinder by Dimensions dialog box appears.

2. Enter the following values:

   Outer radius (RAD1): 0.5
   Z-coordinates (Z1, Z2): -0.25, 0
   Ending angle (THETA2): 180

3. Click **Apply** to create the cylinder and keep the Create Cylinder by Dimensions dialog box open.

4. Enter the following values:

   Outer radius (RAD1): 0.5
   Z-coordinates (Z1, Z2): 1, 1.25
   Ending angle (THETA2): 180

5. Click **Apply** to create the cylinder and keep the Create Cylinder by Dimensions dialog box open.

6. Enter the following values:

   Outer radius (RAD1): 0.25
   Z-coordinates (Z1, Z2): 0, 1
   Ending angle (THETA2): 180

7. Click **OK** to create the cylinder and close the Create Cylinder by Dimensions dialog box.

8. Select **Utility Menu> WorkPlane> Offset WP by increments**

9. Enter 0.05 in X, Y, Z Offset, press enter, and click **OK**. This offsets the working plane 0.05 units in the working plane x-direction.

10. Select **Main Menu> Preprocessor> Modeling> Create> Volumes> Cylinder> By Dimensions**. The Create Cylinder by Dimensions dialog box appears.

11. Enter the following values:

   Outer radius (RAD1): 1
   Optional inner radius (RAD2): 0.35
   Z-coordinates (Z1, Z2): 0, 0.75
   Ending angle (THETA2): 180

12. Click **OK** to create the cylinder and close the Create Cylinder by Dimensions dialog box.

13. Select **Utility Menu> WorkPlane> Offset WP by increments**.

14. Enter -0.10 in X, Y, Z Offset, press enter, and click **OK**. This offsets the working plane -0.10 units in the working plane x-direction.

15. Select **Main Menu> Preprocessor> Modeling> Create> Volumes> Cylinder> By Dimensions**. The Create Cylinder by Dimensions dialog box appears.

16. Enter the following values:

   Outer radius (RAD1): 1
   Optional inner radius (RAD2): 0.35

Z-coordinates (Z1, Z2): 0.75, 1
Ending angle (THETA2): 180

17. Click **OK** to create the cylinder and close the Create Cylinder by Dimensions dialog box.

18. Select **Utility Menu> WorkPlane> Display Working Plane** (toggle off).

19. Select **Main Menu> Preprocessor> Modeling> Operate> Booleans> Glue> Volumes**.

20. Pick all (in the picker).

21. Select **Main Menu> Preprocessor> Numbering Ctrls> Compress Numbers**.

22. Select All for Item to be compressed and click **OK**.

23. Select **Utility Menu> Plot> Volumes**.

2.9.4.6. Mesh Geometry

1. Select **Main Menu> Preprocessor> Meshing> Meshtool**.

2. Under Element Attributes, choose Global and click **Set**.

3. Set the Material number to 1 and click **OK**.

4. Be sure smart sizing is off and click **Mesh**.

5. Pick volumes 4 and 5 (the two annular plates) and click **OK** in the picking menu.

6. Select **Utility Menu> Plot> Volumes**.

7. In the MeshTool dialog box, choose Global and click **Set** under Element Attributes.

8. Set the Material number to 2 and click **OK**.

9. Click **Mesh**.

10. Pick volumes 1, 2, and 3 and click **OK** in the picking menu.

11. Close the MeshTool dialog box.

12. Select **Utility Menu> PlotCtrls> Numbering**.

13. Choose Material numbers for Elem/Attrib numbering and click**OK**.

14. Select **Utility Menu> Plot> Elements**.

15. Select **Main Menu> Preprocessor> Sections> Pretension> Pretensn Mesh> With Options> Divide at Valu> Elements in Volu**.

16. Pick volume 1 and click **OK** in the picker.

17. Enter the following information in the dialog box and click **OK**:

NAME: Example
KCN: Global Cartesian
KDIR: Z-axis
VALUE: 0.5
ECOMP: elems

18. Select **Utility Menu> Select> Comp/Assembly> Create Component**.

19. Enter Line for the Component name (Cname).

20. Choose Lines for the Entity and click **OK**.

21. Select **Utility Menu> PlotCtrls> View Settings> Magnification**.

22. Choose User Specified.

23. Enter 1.1 for the User specified distance and click **OK**.

24. Select **Utility Menu> Plot> Components> Selected Components**.

2.9.4.7. Solution: Apply Pretension

1. Select **Main Menu> Solution> Analysis Types> Sol'n Controls**.

2. Click on the Sol'n Options tab.

3. Choose Precondition CG under Equation Solvers and click **OK**.

4. Select **Utility Menu> Select> Entities**.

5. Choose Areas, By Location, and Y-coordinates and click **OK**.

6. Select **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> Symmetry B.C.> On Areas**.

7. Click **Pick All**.

8. Select **Utility Menu> Select> Entities**.

9. Make sure Areas are still selected and click **Sele All**.

10. Click **OK**.

11. Select **Main Menu> Solution> Define Loads> Apply> Structural> Displacement> On Keypoints**.

12. Pick the middle keypoint on the bottom of the bolt (KeyP No. = 1) and click **OK** in the picker.

13. Choose UX and UZ for DOFs to be constrained (Lab2) and click **Apply** to accept your choices and return to the picker.

14. Pick the middle keypoint on the top of the bolt (KeyP No. = 12) and click **OK** in the picker.

15. Choose UX for DOFs to be constrained (Lab2) and click **OK**.

16. Select **Main Menu> Solution> Define Loads> Apply> Structural> Pretensn Sectn**.

17. Choose 1 Example under Pretension Sections.

18. Enter 100 for Force (under Pretension Load) and click **OK**.

19. Select **Utility Menu> File> Change Title**.

20. Change the title to "Sample Application of PSMESH - Preload Only" and click **OK**.

21. Select **Main Menu> Solution> Solve> Current LS**.

22. Review the information in the /STATUS Command window and click **OK** to begin the solution.

23. Click **Close** when the Solution is Done message appears.

2.9.4.8. Postprocessing: Pretension Results

1. Select **Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu**. The Contour Nodal Solution Data dialog box appears.

2. Select Stress from the scroll box on the left and Z-direction (SZ) from the scroll box on the right and click **OK**.

2.9.4.9. Solution: Apply Thermal Gradient

1. Select **Main Menu> Solution> Analysis Type> Restart**. Close any warning messages that appear.

2. Select **Main Menu> Solution> Define Loads> Settings> Uniform Temp**.

3. Enter 150 for the uniform temperature and click **OK**.

4. Select **Utility Menu> File> Change Title**.

5. Change the title to "Sample Application of PSMESH - Uniform 150 deg" and click **OK**.

6. Select **Main Menu> Solution> Solve> Current LS**.

2.9.4.10. Postprocessing: Pretension and Thermal Results

1. Select **Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu**. The Contour Nodal Solution Data dialog box appears.

2. Select Stress from the scroll box on the left and Z-direction (SZ) from the scroll box on the right and click **OK**.

2.9.4.11. Exit ANSYS

1. Choose QUIT from the ANSYS Toolbar.

2. Choose Quit - No Save!

3. Click on OK.

Chapter 3: Solution

3.1. What Is Solution?

In the solution phase of the analysis, the computer takes over and solves the simultaneous set of equations that the finite element method generates. The results of the solution are:

- Nodal degree of freedom values, which form the primary solution

- Derived values, which form the element solution

The element solution is usually calculated at the elements' integration points. The ANSYS program writes the results to the database as well as to the results file (**.RST**, **.RTH**, **.RMG**, or **.RFL** files).

Several methods of solving the system of simultaneous equations are available in the ANSYS program: sparse direct solution, frontal direct solution, Jacobi Conjugate Gradient (JCG) solution, Incomplete Cholesky Conjugate Gradient (ICCG) solution, Preconditioned Conjugate Gradient (PCG) solution, and an automatic iterative solver option (ITER). The sparse direct solver is the default solver for all analyses, except for electromagnetic analyses with CIRCU124 elements present, analyses that include both p-elements and constraint equations, spectrum analyses, and substructuring analyses (which each use the frontal direct solver by default). In addition to these solvers, the Parallel Performance for ANSYS add-on product includes several multiprocessor solvers: Algebraic Multigrid (AMG) solution and the distributed solvers (Distributed Domain Solver - DDS, Distributed PCG - DPCG, and Distributed JCG - DJCG).

You can select a solver using one of the following:
> **Command(s): EQSLV**
> **GUI: Main Menu> Preprocessor> Loads> Analysis Type> Analysis Options**
> **Main Menu> Solution> Load Step Options> Sol'n Control (: Sol'n Options Tab)**
> **Main Menu> Solution> Analysis Options**
> **Main Menu> Solution> Unabridged Menu> Analysis Options**

Refer to Section 3.2: Selecting a Solver for more information.

> *Note* — The AMG and distributed solvers are part of Parallel Performance for ANSYS, which is a separately-licensed product. For detailed information on multiprocessor solvers, see Chapter 13, "Improving ANSYS Performance and Parallel Performance for ANSYS" in the *ANSYS Advanced Analysis Techniques Guide*.

3.2. Selecting a Solver

The following table provides general guidelines you may find useful in selecting which solver to use for a given problem.

Table 3.1 Solver Selection Guidelines

Solver	Typical Applications	Model Size	Memory Use	Disk Use
Frontal Solver (direct elimination solver)	When robustness is required (nonlinear analysis) or when memory is limited.	Under 50,000 DOFs	Low	High

Solver	Typical Applications	Model Size	Memory Use	Disk Use
Sparse Direct Solver (direct elimination solver)	When robustness and solution speed are required (nonlinear analysis); for linear analysis where iterative solvers are slow to converge (especially for ill-conditioned matrices, such as poorly shaped elements).	10,000 to 500,000 DOFs (more for shell & beam models).	Medium	High
JCG Solver (iterative solver)	When solution speed is crucial in "single-field" problems (thermal, magnetics, acoustics, and multiphysics)	50,000 to 1,000,000+ DOFs	Medium	Low
ICCG Solver (iterative solver)	When solution speed is crucial in multiphysics applications. Handles models that are harder to converge in other iterative solvers.	50,000 to 1,000,000+ DOFs	High	Low
PCG Solver (iterative solver)	When solution speed is crucial (linear analysis of large models). Especially well suited for large models with solid elements.	50,000 to 1,000,000+ DOFs	High	Low
AMG Solver (iterative solver)	Same as the PCG solver. In addition, it is scalable up to 8 processors on shared memory parallel machines.	50,000 to 1,000,000+ DOFs	Shared Memory	
DDS Solver (distributed solver)	This solver is scalable up to a few dozen processors on multiple systems across a network.	1,000,000 - 10,000,000 DOFs	Distributed	
DPCG Solver (distributed solver)	Same as PCG.	50,000 to 10,000,000+ DOFs	Distributed	

> *Note* — The AMG and distributed solvers are part of Parallel Performance for ANSYS, which is a separately-licensed product. For detailed information on multiprocessor solvers, see Chapter 13, "Improving ANSYS Performance and Parallel Performance for ANSYS" in the *ANSYS Advanced Analysis Techniques Guide.*

3.3. Using the Frontal Solver

The frontal solver does *not* assemble the complete global matrix in continuous operation. Instead, ANSYS performs the assembly and solution steps simultaneously as the solver processes each element. The method works as follows:

- After the individual element matrices are calculated, the solver reads in the degrees of freedom (DOF) for the first element.

- The program eliminates any degrees of freedom that can be expressed in terms of the other DOF by writing an equation to the .TRI file. This process repeats for all elements until all degrees of freedom have been eliminated and a complete triangularized matrix is left on the .TRI file.

- The program then calculates the nodal DOF solution by back substitution, and uses the individual element matrices to calculate the element solution.

A term frequently mentioned in the context of a frontal solver is *wavefront*. The wavefront is the number of degrees of freedom retained by the solver during triangularization because they cannot yet be eliminated. As the solver processes each element and its degrees of freedom, the wavefront swells and shrinks, and finally becomes zero when all degrees of freedom have been processed. The highest value of the wavefront is called the *maximum wavefront*, and the averaged, root-mean-square value is called the *RMS wavefront*.

The RMS wavefront of a model directly affects solution time: the smaller its value, the less CPU time required for solution. Before solution, therefore, you may wish to reorder the elements for a smaller wavefront. The ANSYS

program does this automatically at the beginning of solution unless you have reordered the model yourself or have requested no reordering. (The *ANSYS Modeling and Meshing Guide* provides details about element reordering.) The maximum wavefront of a model directly affects memory requirements, in particular the amount of memory allocated for scratch data. See Chapter 19, "Memory Management and Configuration" for details about memory management.

3.4. Using the Sparse Direct Solver

The sparse direct solver is based on a direct elimination of equations as opposed to iterative solvers, where the solution is obtained through indirect means (that is, through iterative solution). Since the sparse direct solver is based on direct elimination, poorly conditioned matrices do not pose any difficulty in producing a solution.

3.5. Using the Jacobi Conjugate Gradient (JCG) Solver

The JCG solver also starts with element matrix formulation, but the remaining steps are different. Instead of tri-angularizing the global matrix, the JCG solver assembles the full global matrix. The solver then calculates the DOF solution by iterating to convergence (starting with an assumed zero value for all DOF). The JCG solver is best suited for 3-D scalar field analyses that involve large, sparse matrices, for example a 3-D magnetic field analysis.

For some cases, the tolerance default value (set via the **EQSLV**,JCG command) of 1.0E-8 may be too restrictive, and may increase running time needlessly. The value 1.0E-5 may be acceptable in many situations.

The JCG solver is available only for static analyses, full harmonic analyses, or full transient analyses. (You specify these analysis types using the commands **ANTYPE**,STATIC, **HROPT**,FULL, or **TRNOPT**,FULL respectively.)

With all conjugate gradient solvers, the user must be particularly careful to check that the model is appropriately constrained. No minimum pivot is calculated and the solver will continue to iterate if there is any rigid body motion possible.

3.6. Using the Incomplete Cholesky Conjugate Gradient (ICCG) Solver

The ICCG solver operates similarly to the JCG solver with the following exceptions:

- The ICCG solver is more robust than the JCG solver for matrices that are not well-conditioned. Performance will vary with matrix conditioning, but in general ICCG performance compares to that of the JCG solver.

- The ICCG solver uses a more sophisticated preconditioner than the JCG solver. Therefore, the ICCG solver requires approximately twice as much memory as the JCG solver.

The ICCG solver is available only for static analyses, full harmonic analyses [**HROPT**,FULL], or full transient analyses [**TRNOPT**,FULL]. (You specify the analysis type using the **ANTYPE** command.) The ICCG solver is useful for structural and multiphysics applications, and for symmetric, unsymmetric, complex, definite, and indefinite matrices. The solver runs faster than the frontal solver for most problem sizes.

3.7. Using the Preconditioned Conjugate Gradient (PCG) Solver

The PCG solver operates similarly to the JCG solver, with these exceptions:

- The PCG solver is usually about 4 to 10 times faster than the JCG solver for structural solid elements and about 10 times faster for shell elements. Savings increase with the problem size.

- The PCG solver uses the **.EMAT** file but not the **.FULL** file.

- The JCG solver uses the diagonal of the global assembled matrix as a preconditioner. The PCG solver uses a more sophisticated preconditioner.

- The PCG solver usually requires approximately twice as much memory as the JCG solver because it retains two matrices in memory:

    – The preconditioner, which is almost the same size as the stiffness matrix

    – The symmetric, nonzero part of the stiffness matrix

A memory savings of up to 70% can result when using **MSAVE**,ON. The **MSAVE** command will cause an element-by-element approach (rather than globally assembling the stiffness matrix) for the parts of the structure using SOLID92, SOLID95, SOLID186, and/or SOLID187 elements that have linear material properties. This feature applies only to static analyses or modal analyses using the PowerDynamics method. (You specify these analysis types using the commands **ANTYPE**,STATIC, or **ANTYPE**,MODAL; **EQSLV**,PCG respectively.) When using SOLID186 and/or SOLID187, only small strain (**NLGEOM**,OFF) analyses are allowed. The solution time may be affected depending on the processor speed and manufacturer of your computer, as well as the chosen element options (for example, 2 x 2 x 2 integration for SOLID95) .

You can use the **/RUNST** command (**Main Menu> Run-Time Stats**), to determine the space needed or the wavefront size. Special memory allocation is desirable. (See Chapter 19, "Memory Management and Configuration" for further information.)

The PCG solver usually requires less than 1/4 the disk file space used by the frontal solver. Savings increase with the problem size. The PCG solver usually runs faster than the frontal solver for large models (wavefront greater than about 1000).

The PCG solver is best suited for structural analysis. It is valid for elements with symmetric, sparse, definite or indefinite matrices. This solver is available only for static or steady-state analyses and transient analyses, or for subspace eigenvalue analyses (PowerDynamics).

The PCG solver primarily solves for displacements/rotations (in structural analysis), temperatures (in thermal analysis), etc. The accuracy of other derived variables (such as strains, stresses, flux, etc.) is dependent upon accurate prediction of primary variables.

Direct solvers (such as the frontal and sparse direct solvers) produce very accurate solutions. Indirect iterative solvers, such as the PCG solver, depend upon a convergence criterion specified by the user. Therefore, relaxation of the default tolerance may significantly affect the accuracy, especially of derived quantities.

The PCG solver is not recommended for models with SHELL150 elements. For these types of problems, use the sparse solver. Also, the PCG solver does not support SOLID62 or MATRIX50 elements.

Note —

- With all conjugate gradient solvers, the **RIGID** command is not applicable and the user must be particularly careful to check that the model is appropriately constrained. No minimum pivot is calculated and the solver will continue to iterate even if there is any rigid body motion possible.

- In a modal analysis, the number of modes should be limited to 12 or less.

- When the PCG solver encounters an indefinite matrix, the solver will invoke an algorithm that handles indefinite matrices. If the indefinite PCG algorithm also fails (this happens when the equation system is ill-conditioned; for example, losing contact at a substep or plastic hinge developments), an outer Newton-Raphson loop will be triggered to perform a bisection. Normally the stiffness matrix will be better conditioned after bisection and the PCG solver can eventually solve all the nonlinear steps.

3.8. Using the Algebraic Multigrid (AMG) Solver

The Algebraic Multigrid (AMG) solver, which is based on the multi-level method, is an iterative solver that you can use in single- and multiprocessor environments. To use this solver, you must have a license for the Parallel Performance for ANSYS advanced task (add-on). For detailed information on this solver, see Section 13.2.1: About the Algebraic Multigrid (AMG) Solver in the *ANSYS Advanced Analysis Techniques Guide*.

Algebraic Multigrid iterative equation solver is available for STATIC analyses and TRANS (full method only) analyses. It is very efficient for single-field structural analyses (where the solution DOFs are combinations of UX, UY, UZ, ROTX, ROTY, and ROTZ). For applications such as single-field thermal analyses (where the solution DOF is TEMP), the AMG solver is less efficient. Recommended for ill-conditioned problems in which the PCG and ICCG solvers would have difficulty converging; applicable in both single- and multiprocessor environments. In terms of CPU time when used in a single-processor environment, the AMG solver performs better than the PCG and ICCG solvers for ill-conditioned problems, and it delivers about the same level of performance for ordinary problems. In a multiprocessor environment, the AMG solver scales better than the PCG and ICCG solvers on shared memory parallel machines. The AMG solver also handles indefinite matrix problems for nonlinear analyses.

3.9. Using the Distributed Solvers (DDS, DPCG, DJCG)

The distributed solvers (DDS, DPCG, DJCG) decompose large models into smaller domains, then send those domains to multiple processors. To use these solvers, you must have a license for the Parallel Performance for ANSYS add-on.

The distributed solvers work by dividing the meshed model into domains without user intervention. During the solution phase, the solution is launched on the host system and on all designated systems simultaneously. Each processor solves its own set of domains, communicating with the other processors. The host system then builds the total DOF solution, and the analysis continues on the host system only.

The domains noted here are domains which are used by a solver's preconditioners only. The larger domains built based on these "preconditioner" domains are called CPU domains. The number of CPU domains is equal to the number of processors used.

Choose the appropriate distributed solver based on your particular analysis. In general, use the DDS solver for large static or full transient analyses. The DJCG solver is best suited for large 3-D scalar field analyses, such as a 3-D thermal or magnetic field analysis. The DPCG solver is best suited for structural analyses. It is valid for elements with symmetric, sparse, or indefinite matrices.

These solvers can dramatically improve solving time by using multiple processors.

For detailed information on these solvers, see Section 13.2.2: About the Distributed Solvers (DDS, DPCG, DJCG) in the *ANSYS Advanced Analysis Techniques Guide*.

3.10. Using the Automatic Iterative (Fast) Solver Option

The Automatic Iterative Solver option [**EQSLV**,ITER] chooses an appropriate iterative solver (PCG, JCG, etc.) based on the physics of the problem being solved. When you use the Automatic Iterative Solver option you must input an accuracy level. The accuracy level is specified as an integer between 1 and 5 and is used for selecting the Iterative Solver tolerance for convergence checking. An accuracy level of 1 corresponds to the fastest setting (less number of iterations) and an accuracy level to 5 corresponds to the slowest setting (accurate, more number of iterations). ANSYS selects the tolerance based on the chosen accuracy level. For example:

- For linear static or linear full transient structural analysis, an accuracy level of 1 corresponds to a tolerance of 1.0e-4 and an accuracy level of 5 corresponds to a tolerance of 1.0e-8.

- For steady-state linear or nonlinear thermal analysis, an accuracy level of 1 corresponds to a tolerance of 1.0e-5 and accuracy level of 5 corresponds to a tolerance of 1.0e-9.

- For transient linear or nonlinear thermal analysis, an accuracy level of 1 corresponds to a tolerance of 1.0e-6 and an accuracy level of 5 corresponds to a tolerance of 1.0e-10.

This solver option is available only for linear static and linear full transient structural analysis and steady-state/transient linear or nonlinear thermal analysis.

Since the solver and tolerance are selected based on the physics and conditions of the problem being solved, it is recommended that this command be issued immediately before solving the problem (once the problem has been completely defined).

When the automatic iterative solver option is chosen and appropriate conditions have been met, neither the **Jobname.EMAT** nor **Jobname.EROT** files will be created for structural analysis and thermal analysis. This option is not recommended for thermal analysis involving phase change. When this option is chosen but the appropriate conditions have *not* been met, ANSYS uses the Frontal solver for the solution and issues a message displaying the solver and tolerance used in the solution.

3.11. Using Special Solution Controls for Certain Types of Structural Analyses

When you are performing certain types of structural analyses, you can take advantage of these special solution tools:

- Abridged **Solution** menus, which are available for static, transient (all solution methods), modal, and buckling analyses. See Section 3.11.1: Using Abridged Solution Menus.

- The **Solution Controls** dialog box, which is available for static and transient (full solution method only) analyses. See Section 3.11.2: Using the Solution Controls Dialog Box.

3.11.1. Using Abridged Solution Menus

If you are using the GUI to perform a structural static, transient, modal, or buckling analysis, you have the choice of using abridged or unabridged **Solution** menus:

- Unabridged **Solution** menus list all solution options, regardless of whether it is recommended, or even possible, for you to use them in the current analysis. (If it is not possible for you to use an option in the current analysis, the option is listed but is grayed out.)

- Abridged **Solution** menus are simpler. They list only those options that apply to the type of analysis that you are performing. For example, if you are performing a static analysis, the **Modal Cyclic Sym** option does not appear on the abridged **Solution** menu. Only those options that are valid and/or recommended for the current analysis type appear.

If you are performing a structural analysis, the abridged **Solution** menu appears by default when you enter the solution processor (**Main Menu> Solution**).

If your analysis is either static or full transient, you can use the options on the menu to complete the solution phase of your analysis. However, if you select a different analysis type, the default abridged **Solution** menu that you see above will be replaced by a different **Solution** menu. The new menu will be appropriate for the analysis type you select.

All variants of the abridged **Solution** menu contain an **Unabridged Menu** option. This option is always available for you to select in case you prefer using the unabridged menu.

If you do one analysis and then choose to do a new analysis within the same ANSYS session, ANSYS will (by default) present you with the same type of **Solution** menu that you used for the first analysis. For example, if you choose to use the unabridged **Solution** menu to perform a static analysis and then select a new buckling analysis, ANSYS presents you with the unabridged **Solution** menu that is appropriate for buckling analyses. However, you can toggle between the unabridged and abridged **Solution** menus at any time during the solution phase of the analysis by selecting the appropriate menu option (**Main Menu> Solution> Unabridged Menu** or **Main Menu> Solution> Abridged Menu**).

3.11.2. Using the Solution Controls Dialog Box

If you are performing a structural static or full transient analysis, you can use a streamlined solution interface (called the **Solution Controls** dialog box) for setting many of your analysis options. The **Solution Controls** dialog box consists of five tabbed "pages," each of which contains a set of related solution controls. The dialog box is useful for specifying the settings for each load step of a multiple load step analysis.

As long as you are performing a structural static or full transient analysis, your **Solution** menu will contain the **Sol'n Control** option. When you click the **Sol'n Control** menu item, the **Solution Controls** dialog box appears. This dialog box provides you with a single interface for setting analysis and load step options. See Figure 3.1: "Solution Controls Dialog Box" for an illustration.

Figure 3.1 Solution Controls Dialog Box
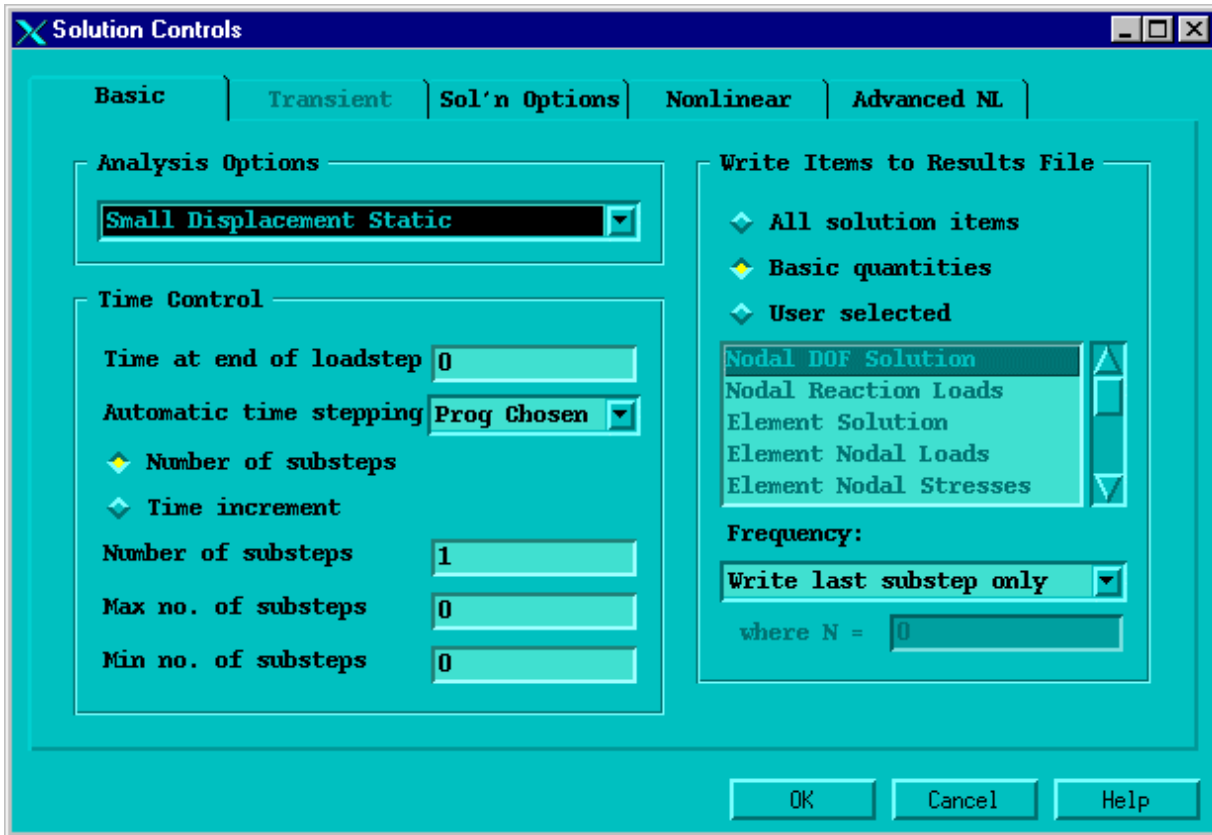
The **Basic** tab, which is shown above, is active when you access the dialog box. The complete list of tabs, in order from left to right, is as follows:

- **Basic**
- **Transient**
- **Sol'n Options**
- **Nonlinear**
- **Advanced NL**

Each set of controls is logically grouped on a tab; the most basic controls appear on the first tab, with each subsequent tab providing more advanced controls. 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.

Each of the controls on the **Solution Controls** dialog box corresponds to an ANSYS command. The table below illustrates the relationships between the tabs and the command functionality that you can access from each.

Table 3.2 Relationships Between Tabs of the Solution Controls Dialog Box and Commands

Solution Controls Dialog Box Tab	What Does This Tab Let You Do?	What Commands Are Related to This Tab?
Basic	Specify the type of analysis that you want to perform. Control various time settings. Specify the solution data that you want ANSYS to write to the database.	**ANTYPE**, **NLGEOM**, **TIME**, **AUTOTS**, **NSUBST**, **DELTIM**, **OUTRES**
Transient	Specify transient options, such as transient effects and ramped vs. stepped loading. Specify damping options. Choose time integration method. Define integration parameters.	**TIMINT**, **KBC**, **ALPHAD**, **BETAD**, **TRNOPT**, **TINTP**
Sol'n Options	Specify the type of equation solver that you want to use. Specify parameters for performing a multiframe restart. Specify configuration details for distributed solvers	**EQSLV**, **RESCONTROL**, **DSOPT**, **DSPROC**
Nonlinear	Control nonlinear options, such as line search and solution predictor. Specify the maximum number of iterations that are allowed per substep. Indicate whether you want to include creep calculation in the analysis. Control bisections. Set convergence criteria.	**LNSRCH**, **PRED**, **NEQIT**, **RATE**, **CUTCONTROL**, **CNVTOL**
Advanced NL	Specify analysis termination criteria. Control activation and termination of the arc-length method.	**NCNV**, **ARCLEN**, **ARCTRM**

Once you are satisfied with the settings on the **Basic** tab, you do not need to progress through the remaining tabs unless you want to change some of the advanced 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.

> *Note* — Whether you make changes to only one or to multiple tabbed pages, your changes are applied to the ANSYS database only when you click **OK** to close the dialog box.

3.11.3. Accessing More Information

Discussions of the **Solution Controls** dialog box are included throughout the ANSYS manual set as applicable.

For additional information, refer to the following:

- Online help for the **Solution Controls** dialog box
- Chapter 2, "Structural Static Analysis" in the *ANSYS Structural Analysis Guide*
- Chapter 5, "Transient Dynamic Analysis" in the *ANSYS Structural Analysis Guide*
- Chapter 8, "Nonlinear Structural Analysis" in the *ANSYS Structural Analysis Guide*

3.12. Using the PGR File to Store Data for Postprocessing

In many analyses, a large amount of preliminary graphics information is created in order to obtain specific solution data. This information is often discarded when the final solution criteria is reached, even though you may request it later, during POST1 operations. You can have ANSYS save this information for rapid POST1 access by designating a PGR file. The PGR file is a dedicated ANSYS data storage format that saves this "precooked" data for rapid access in POST1.

You can designate the items to be included in the PGR file during solution and gain up to a 10X performance benefit when you access the information during POST1. You can also create the PGR file in POST1, append new types of data to the PGR file, or create a new PGR file from any existing results file.

You use a dedicated postprocessing tool, The Results Viewer, to access the information you store in the PGR file. The ANSYS results Viewer is a compact toolbar for viewing your analysis results. Although it is designed to display the information in your PGR file, you can use it to access any data from a valid results file (***.RST, *.RFL, *.RTH, *.RMG**, etc.). For more information on the Results Viewer, see Section 5.4.3.1: The Results Viewer Layout later on in this manual.

3.12.1. PGR File Capability

You use the PGR file to rapidly access complex display data during postprocessing. This data is often converted from machine language information to display data (graphical representations) during the Solution phase of your analysis. Although this information is written to the results file as solution parameters, the process of recon-verting it for viewing in POST1 can be time consuming. The PGR file preserves modeling and display data as a graphical object, allowing the data to be accessed and displayed in POST1 markedly faster. The PGR uses the existing ANSYS command structure to define, generate and access the data that will be saved and retrieved. See Section 3.12.3: PGR Commands later in this chapter for links to the various PGR commands.

Your PGR file will always contain the nodal solution data defined in the **POUTRES** command. You can also specify the following items for inclusion in your PGR file:

- Stress
- Structural nonlinear data
- Contact data (3-D only)
- Total Strain
- Elastic Strain
- Thermal Strain
- Creep Strain
- Thermal Gradient
- Thermal Flux
- Electric Field
- Electric Flux Density
- Magnetic Field Intensity
- Magnetic Flux Density
- Magnetic Forces
- Pressure Gradients

- Body Temperatures
- Densities for Topological Optimization

Because the PGR file is constructed without resynchronizing the **.db** file, the following items will not provide valid data to the PGR file:

- Sourc36 Elements
- Super Elements
- Circuit Elements

3.12.2. Selecting Information for the PGR File

The information you choose in Solution will be stored in the PGR file. Although you can create or append a PGR file in POST1 (by accessing the .rst file), choosing the information during Solution will provide the greatest speed/accessibility benefits.

You designate the PGR solution data by accessing the unabridged Solution menu (**Main Menu> Solution> Unabridged**). From the unabridged solution menu, you select **Output Cntrls**, and then **PGR File**. The following PGR File Options dialog box appears:

Figure 3.2 PGR File Options
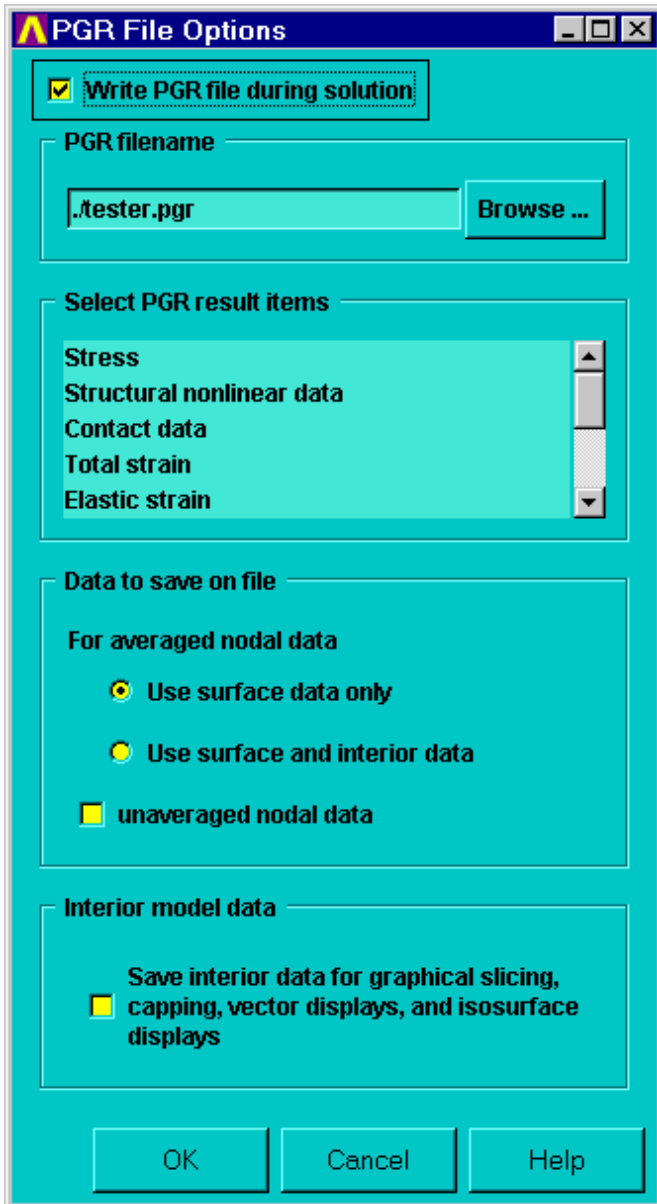
From this panel, you can designate the location and the name of your PGR file, the type of data you wish to include in the file, the type of averaging scheme(s) for the data, and whether or not to include data from the interior portions of your model. You can change any of the items you wish to include in your PGR file up until the first solution. After you solve (even just one load step) you cannot make changes. See Section 5.4.2: Appending to an Existing PGR File in POST1 for information on how to append your file after solution.

The options available from the PGR File Options panel correspond to the **PGWRITE** and **POUTRES** commands, and to some extent, the **AVRES** command.

You can access the following operations from the PGR File Options Panel :

Write PGR file during solution - Checking this box creates the PGR file. If you are restarting a solution, you will overwrite the existing PGR file.

PGR filename - From this location, you can provide a specific name for your PGR file and designate an alternate directory location to create it in. The default file name is **Jobname.pgr**, and the default location is the working directory of the current analysis.

Select PGR result items - You use this list box to select the items you want to include in your PGR file. You can hold down the Ctrl key to select multiple items individually, drag your mouse across multiple select items, or use the Shift key to select the boundaries of inclusive lists. The following items can be included in a PGR file (the available items will change according to the context of your analysis):

- Nodal Solution Data
- Stress
- Structural nonlinear data
- Contact data (3-D only)
- Total Strain
- Elastic Strain
- Thermal Strain
- Creep Strain
- Thermal Gradient
- Thermal Flux
- Electric Field
- Electric Flux Density
- Magnetic Field Intensity
- Magnetic Flux Density
- Magnetic Forces
- Pressure Gradients
- Body Temperatures
- Densities for Topological Optimization

Data to save on file - From this location, you can designate whether to store averaged data or averaged plus unaveraged nodal data. You can also specify whether to use the surface data or the surface data in conjunction with the interior data. Averaged data is used with the **PLNSOL** and **PRNSOL** commands. Unaveraged data is used with the **PLESOL** and **PRESOL** commands.

The averaging scheme used for the "Surface and Interior data" selection will yield stress contours that are similar to those obtained in the Full Model Graphics mode or in PowerGraphics with the **AVRES**,,FULL command option. The data obtained with the "Surface data only selection will be the same as the data obtained using PowerGraphics with the default **AVRES** command option (using only the exterior element faces). Interior data can be obtained only when nodal data averaging is enabled. This function cannot be changed if you plan to append your PGR file.

Interior model data - This selection actually saves the interior results data for subsequent displays using slicing, capping, vector display, or isosurface display techniques (see the **/TYPE**, **/CTYPE**, and **PLVECT** commands). The data that is saved when this item is selected can be displayed on the model or ported to data tables and listings. This function cannot be changed if you plan to append your PGR file.

Stresses can only be displayed in the coordinate system that was active when the PGR file was written. If you wish to use the results viewer to view stresses in other coordinate system displays, you must reload your results file (***.RST, *.RFL, *.RTH, *.RMG**, etc.) in POST1, in that coordinate system.

3.12.3. PGR Commands

The ANSYS PGR file uses the following commands to create and access the PGR data:

Solution Commands
    PGWRITE, **POUTRES**, and **AVRES**.

Postprocessing Commands
    POUTRES, **PGSAVE**, **PGRAPH**, **PGRSET**, **PLESOL**, **PLNSOL**, **PLTRAC**, and **PLVECT**.

3.13. Obtaining the Solution

To initiate the solution, use one of the following:
        Command(s): SOLVE
        GUI: Main Menu> Solution> Current LS or Run FLOTRAN

Because the solution phase generally requires more computer resources that the other phases of an analysis, it is better suited to batch (background) mode than interactive mode.

The solver writes output to the output file (**Jobname.OUT**) and the results file. If you run the solution interactively, the output "file" is actually your screen (window). By using one of the following before issuing **SOLVE**, you can divert the output to a file instead of the screen:
        Command(s): /OUTPUT
        GUI: Utility Menu> File> Switch Output to> File or Output Window

Data written to the output file consist of the following:

  · Load summary information

  · Mass and moments of inertia of the model

  · Solution summary information

  · A final closing banner that gives total CPU time and elapsed time.

  · Data requested by the **OUTPR** output control command or its GUI counterpart

In interactive mode, much of the output is suppressed. The results file (**.RST**, **.RTH**, **.RMG**, or **.RFL**) contains all results data in binary form, which you can then review in the postprocessors.

Another useful file produced during solution is **Jobname.STAT**, which gives the status of the solution. You can use this file to monitor an analysis while it is running. It is particularly useful in iterative analyses such as nonlinear and transient analyses.

The **SOLVE** command calculates the solution for the load step data currently in the database.

3.14. Solving Multiple Load Steps

There are three ways to define and solve multiple load steps:

  · Multiple **SOLVE** method

  · Load step file method

- Array parameter method.

3.14.1. Using the Multiple SOLVE Method

This method is the most straightforward. It involves issuing the **SOLVE** command after each load step is defined. The main disadvantage, for interactive use, is that you have to wait for the solution to be completed before defining the next load step. A typical command stream for the multiple **SOLVE** method is shown below:

```
/SOLU
...
! Load step 1:
D,...
SF,...
0
SOLVE           ! Solution for load step 1
! Load step 2
F,...
SF,...
...
SOLVE            ! Solution for load step 2
Etc.
```

3.14.2. Using the Load Step File Method

The load step file is a convenient method to use when you want to solve problems while you are away from your terminal or PC (for example, overnight). It involves writing each load step to a load step file (via the **LSWRITE** command or its GUI equivalent) and, with one command, reading in each file and obtaining the solution. See Chapter 2, "Loading" for details about creating load step files.

To solve multiple load steps, issue the **LSSOLVE** command (**Main Menu> Solution> From LS Files**). **LSSOLVE** is actually a macro that reads in the load step files sequentially and initiates the solution for each load step. A sample command input for the load step file method is shown below:

```
/SOLU            ! Enter SOLUTION
...
!  Load Step 1:
D,...                    ! Loads
SF,...
...
NSUBST,...                ! Load step options
KBC,...
OUTRES,...
OUTPR,...
...
LSWRITE                  ! Writes load step file: Jobname.S01
!  Load Step 2:
D,...                    ! Loads
SF,...
...
NSUBST,...                ! Load step options
KBC,...
OUTRES,...
OUTPR,...
...
LSWRITE                  ! Writes load step file: Jobname.S02
...
0
LSSOLVE,1,2        ! Initiates solution for load step files 1 and 2
```

See the *ANSYS Commands Reference* for a discussion of the **NSUBST**, **KBC**, **OUTRES** ,**OUTPR**, **LSWRITE**, and **LSSOLVE** commands.

3.14.3. Using the Array Parameter Method

This method, mainly intended for transient or nonlinear static (steady-state) analyses, requires knowledge of array parameters and do-loops, which are part of APDL (ANSYS Parametric Design Language). See the *ANSYS APDL Programmer's Guide* for information about APDL. The array parameter method involves building tables of *load* versus *time* using array parameters and is best explained by the following example.

Figure 3.3 Examples of Time-Varying Loads
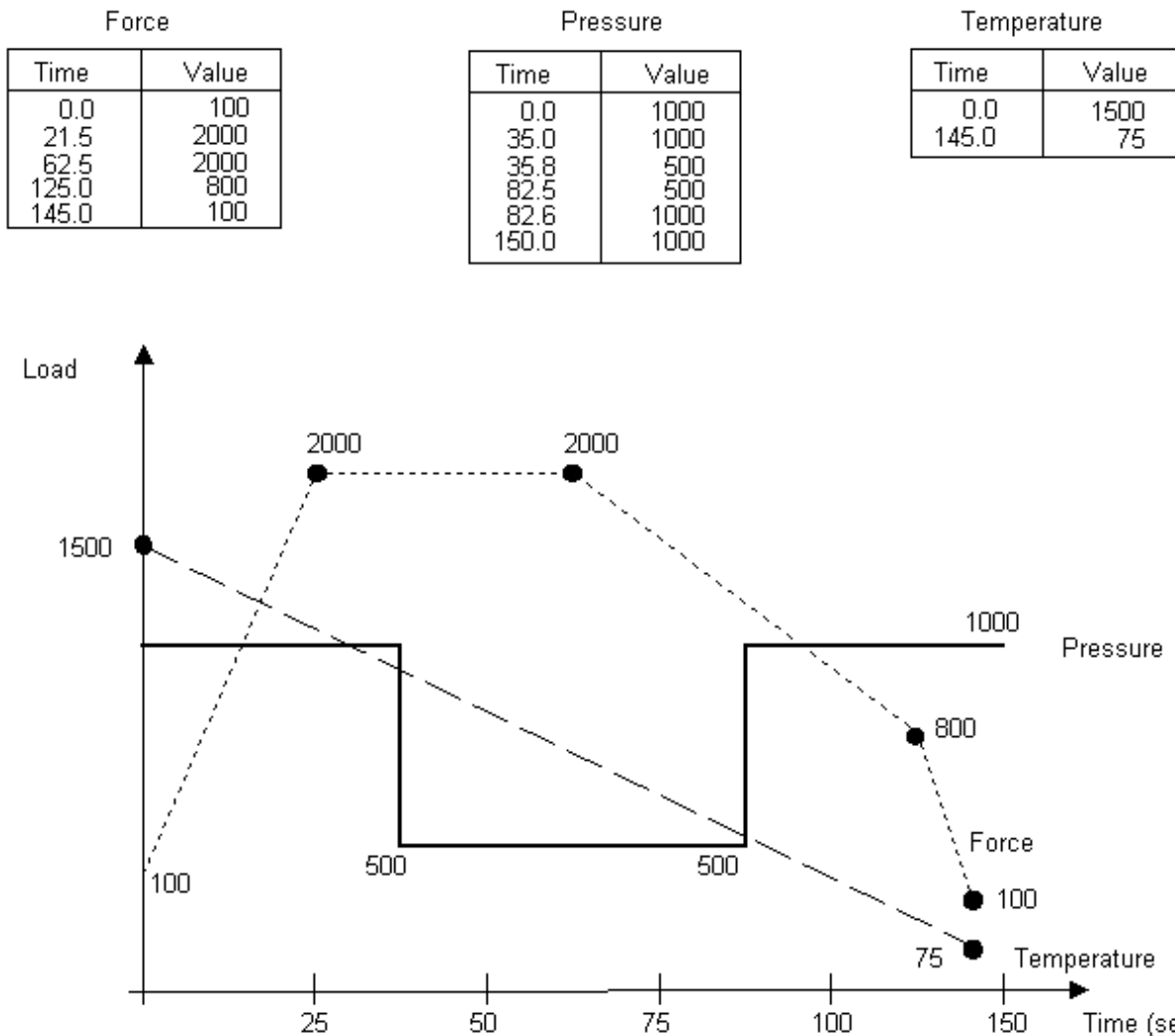
Suppose that you have a set of time-varying loads such as the ones shown above. There are three load functions, so you need to define three array parameters. All three array parameters must be of type TABLE. The force function has five points, so it needs a 5 x 1 array; the pressure function needs a 6 x 1 array; and the temperature function needs a 2 x 1 array. Notice that all three arrays are one-dimensional. The load values are entered in column 1 and the time values are entered in column *zero*. (The *zero*th column and *zero*th row, which normally contain index numbers, must be changed and filled with a monotonically increasing set of numbers if you define the array parameter as a TABLE.)

To define the three array parameters, you first need to declare their type and dimensions. To do so, use either of the following:

> **Command(s): *DIM**
> **GUI: Utility Menu> Parameters> Array Parameters> Define/Edit**

ANSYS Basic Analysis Procedures Guide . ANSYS Release 8.1 . 001972 . © SAS IP, Inc.

For example:

```
*DIM,FORCE,TABLE,5,1
*DIM,PRESSURE,TABLE,6,1
*DIM,TEMP,TABLE,2,1
```

You can now use either the array parameter editor (**Utility Menu> Parameters> Array Parameters> Define/Edit**) or a set of "=" commands to fill these arrays. The latter method is shown below.

```
FORCE(1,1)=100,2000,2000,800,100  ! Force values in column 1
FORCE(1,0)=0,21.5,50.9,98.7,112   ! Corresponding time values in column 0
FORCE(0,1)=1 ! Zeroth row
PRESSURE(1,1)=1000,1000,500,500,1000,1000
PRESSURE(1,0)=0,35,35.8,74.4,76,112
PRESSURE(0,1)=1
TEMP(1,1)=800,75
TEMP(1,0)=0,112
TEMP(0,1)=1
```

You have now defined the load histories. To apply these loads and obtain the solution, you need to construct a do-loop (using the commands ***DO** and ***ENDDO**) such as the one shown below:

```
TM_START=1E-6                ! Starting time (must be > 0)
TM_END=112                   ! Ending time of the transient
TM_INCR=1.5                  ! Time increment
*DO,TM,TM_START,TM_END,TM_INCR   ! Do for TM from TM_START to TM_END in
                                 !  steps of TM_INCR
  TIME,TM                    ! Time value
  F,272,FY,FORCE(TM)         ! Time-varying force (at node 272, FY)
  NSEL,...                   ! Select nodes on pressure surface
  SF,ALL,PRES,PRESSURE(TM)   ! Time-varying pressure
  NSEL,ALL                   ! Activate all nodes
  NSEL,...                   ! Select nodes for temperature specification
  BF,ALL,TEMP,TEMP(TM)       ! Time-varying temperature
  NSEL,ALL                   ! Activate all nodes
  SOLVE                      ! Initiate solution calculations
*ENDDO
```

See the *ANSYS Commands Reference* for discussions of the ***DO**, **TIME**, **F**, **NSEL**, **SF**, **BF**, and ***ENDDO** commands.

You can change the time increment (TM_INCR parameter) very easily with this method. With other methods, changing the time increment for such complex load histories would be quite cumbersome.

3.15. Terminating a Running Job

You can terminate a running ANSYS job, if necessary, with the help of system functions such as a system break, issuing a kill signal, or deleting the entry in the batch queue. For nonlinear analyses, however, this is not the preferred method, because a job terminated in this manner cannot be restarted.

To terminate a nonlinear analysis "cleanly" on a multitasking operating system, create an abort file, named **Jobname.ABT** (or, on some case-sensitive systems, **jobname.abt**), containing the word *nonlinear* on the first line, starting in column 1. At the start of an equilibrium iteration, if the ANSYS program finds such a file in the working directory, the analysis will be stopped and can be restarted at a later time.

> *Note* — If commands are being read using a file specified via the **/INPUT** command (**Main Menu> Pre-processor> Material Props> Material Library**, or **Utility Menu> File> Read Input from**), the abort file will terminate the solution, but the program will continue to read commands from the specified input file. Thus, any postprocessing commands included in the input file will execute.

3.16. Restarting an Analysis

Sometimes you may need to restart an analysis after the initial run has been completed. For instance, you may want to add more load steps to the analysis. These may be additional loading conditions in a linear static analysis or additional portions of a time-history loading curve in a transient analysis. Or, you may need to recover from a convergence failure in a nonlinear analysis.

ANSYS allows two different types of restarts: the singleframe restart and the multiframe restart, which can be used for static or full transient structural analyses. The singleframe restart only allows you to resume a job at the point it stopped. The multiframe restart can resume a job at any point in the analysis for which information is saved. This capability allows you to do multiple analyses of a model and gives you more options for recovering from an abort.

The model must meet the following conditions to restart an analysis:

- The analysis type must be either static (steady-state), harmonic (2-D magnetic only), or transient (full method only). No other analysis can be restarted.

- At least one iteration must have been completed in the initial run.

- The initial run should not have stopped due to a "killed" job, system break, or system crash.

- You performed your initial analysis and generated the restart file under the same ANSYS version number.

3.16.1. Singleframe Restart

A traditional restart requires that certain files from the initial run of the job are present, and requires that you make any changes to the input before the **SOLVE** command.

3.16.1.1. Singleframe Restart Requirements

The following files must be available from the initial run:

- **Jobname.DB** - the database file saved immediately *after* the initial **SOLVE**. If you save the database at any point later in the analysis, boundary conditions and other variables may be changed from their initial values, which would prevent the restart from running properly. (For non-converged solutions, the database file is saved automatically; see the note below.)

- **Jobname.EMAT** - element matrices (if created).

- **Jobname.ESAV** or **.OSAV** - Element saved data (**.ESAV**) or old element saved data (**.OSAV**). **Jobname.OSAV** is required only if the **.ESAV** file is missing, incomplete, or otherwise corrupted because of a diverging solution; because the displacement limit was exceeded; or because of a negative pivot (see Table 3.3: "Restart Information for Nonlinear Analyses"). It is written if $KSTOP$ is set to 1 (default) or 2 on the **NCNV** command, or if automatic time stepping is active. If the **.OSAV** file is required, you must rename it as **Jobname.ESAV** before restarting the analysis.

- Results file - This file is not required, but if it is available, results from the restart run will be appended to it with the proper, sequential load step and substep numbers. If the initial run terminated because the number of results sets on the results file were exceeded, you will need to rename the initial results file to a different name before restarting. To do so, issue the **/ASSIGN** command (**Utility Menu> File> ANSYS File Options**).

Note — If the program terminates due to nonconvergence, time limits, the abort file (**Jobname.ABT**), or other program-detected failure, the database is automatically saved, and the solution output (**Jobname.OUT**) will list the files and other information required for restarting. See also Table 3.3: "Restart Information for Nonlinear Analyses" for a list of termination causes and the element saved data file needed to restart.

If the files **.RDB**, **.LDHI**, or **.R***nnn* were accidentally created from a previous run, you must delete them before doing a singleframe restart.

In interactive mode, an existing database file is first written to a backup file (**Jobname.DBB**). In batch mode, an existing database file is replaced by the current database information with no backup.

Table 3.3 Restart Information for Nonlinear Analyses

Cause of Termination	Element Saved Data File	Required Corrective Action
Normal (i.e., no errors)	**Jobname.ESAV**	Add more load steps at the end of your job.
Nonconvergence	**Jobname.OSAV**	Define a smaller time step, change the adaptive descent option, or take other action to enhance convergence. Rename **Jobname.OSAV** as **Jobname.ESAV** before restarting.
Nonconvergence due to insufficient equilibrium iterations	**Jobname.ESAV**	If the solution was converging, allow more equilibrium equations (**NEQIT** command).
Cumulative iteration limit exceeded (**NCNV** command)	**Jobname.ESAV**	Increase *ITLIM* on **NCNV** command.
Time limit exceeded (**NCNV**)	**Jobname.ESAV**	None (simply restart the analysis). (If you were running the analysis interactively and you want to restart it from within the same ANSYS session, you must reset the time limits before attempting the restart.)
Displacement limit exceeded (**NCNV**)	**Jobname.OSAV**	(Same as for nonconvergence.)
Negative pivot	**Jobname.OSAV**	(Same as for nonconvergence.)
Jobname.ABT • if solution was converging • if solution was diverging	**Jobname.EMAV**, **Jobname.OSAV**	Make whatever changes are necessary to address the behavior that caused you to voluntarily terminate the analysis.
"Full" results file (more than 1000 substeps). Time steps output.	**Jobname.ESAV**	Could indicate a problem - check settings on **CNVTOL**, **DELTIM**, and **NSUBST**, or KEYOPT(7) for contact elements. Or, specify larger number of results allowed on results file [**/CONFIG**,NRES] *before* solution or reduce the number of results to be output. Also rename results file (**/ASSIGN**).
"Killed" job (system break), system crash, or system time-limit exceeded	Not applicable	No restart is possible.

3.16.1.2. Singleframe Restart Procedure

The procedure to do a restart analysis is as follows:

1.   Enter the ANSYS program and specify the same jobname that was used in the initial run with **/FILNAME** (**Utility Menu> File> Change Jobname**).

2. Enter the SOLUTION processor using **/SOLU** (**Main Menu> Solution**), then resume the database file using **RESUME** (**Utility Menu> File> Resume Jobname.db**).

3. Indicate that this is a restart analysis by issuing **ANTYPE,,REST** (**Main Menu> Solution> Restart**).

4. Specify revised or additional loads as needed. Modified ramped loads start from their previous values. Newly applied ramped loads are ramped from zero; newly applied body loads start from initial values. Deleted loads which are reapplied are treated as new, not modified, loads. Surface and body loads to be deleted should be ramped to zero or to the initial value, so that the **Jobname.ESAV** and **Jobname.OSAV** files are consistent with the database.

    Be sure to take whatever corrective action is necessary if you are restarting from a convergence failure.

5. If you used the Frontal solver, specify whether the triangularized matrix (**Jobname.TRI**) is to be reused, using one of the following:
    > **Command(s): KUSE**
    > **GUI: Main Menu> Preprocessor> Loads> Other> Reuse Tri Matrix**
    > **Main Menu> Solution> Other> Reuse Tri Matrix**

    By default, the ANSYS program calculates a new triangularized matrix for the first load step of a restart run. By issuing the **KUSE**,1 command, you can force the program to reuse the existing matrix, thereby saving a significant amount of computer time. However, you can reuse **Jobname.TRI** only under certain conditions, in particular if the specified DOF constraints have not changed and it is a linear analysis. See the *ANSYS, Inc. Theory Reference* for details.

    By issuing **KUSE**,-1, you can cause ANSYS to redo the element matrices. This can be useful for debugging analyses and for handling error cases.

    Sometimes, you may have to analyze the same model for different constraint conditions, for instance a quarter-symmetry model with symmetry-symmetry (SS), symmetry-antisymmetry (SA), antisymmetry-symmetry (AS), and antisymmetry-antisymmetry (AA) conditions. In such a situation, keep the following points in mind:

    •   All four cases (SS, SA, AS, AA) require a new triangularized matrix.

    •   You can keep copies of the **Jobname.TRI** files for the various cases and use them as appropriate.

    •   You can use substructuring (with the constrained nodes as master DOF) to minimize computing time. (See Chapter 8, "Substructuring" in the *ANSYS Advanced Analysis Techniques Guide*.)

6. Initiate the restart solution by issuing the **SOLVE** command. (See Section 3.13: Obtaining the Solution for details.)

7. Repeat steps 4, 5, and 6 for additional load steps, if any. Or use the load step file method to create and solve multiple load steps. Use the following commands:
    > **Command(s): LSWRITE**
    > **GUI: Main Menu> Preprocessor> Loads> Write LS File**
    > **Main Menu> Solution> Write LS File**
    > **Command(s): LSSOLVE**
    > **GUI: Main Menu> Solution> From LS Files**

8. Postprocess as desired, then exit the ANSYS program.

A sample restart input listing is shown below.

```
!  Restart run:
/FILNAME,...        ! Jobname
RESUME
```

ANSYS Basic Analysis Procedures Guide . ANSYS Release 8.1 . 001972 . © SAS IP, Inc.

```
/SOLU
ANTYPE,,REST         ! Specify restart of previous analysis
!
! Specify new loads, new load step options, etc.
! Take appropriate corrective action for nonlinear analyses.
!
SOLVE                ! Initiate restart solution
SAVE                 ! Optional SAVE for possible subsequent singleframe restart
FINISH
!
! Postprocess as desired
!
/EXIT,NOSAV
```

3.16.1.3. Restarting a Nonlinear Analysis From an Incompatible Database

Sometimes, postprocessing is performed prior to a restart. If you issue **SET** and **SAVE** commands during this postprocessing, the boundary conditions in your database might be altered and become inconsistent with those needed for a restart. By default, the program saves your file automatically when you exit. At the end of solution, the boundary conditions for the last load step are stored in the database memory. (The database contains only one set of boundary conditions.)

A **SET** command in POST1 (other than **SET**,LAST) reads the boundary conditions for the specified results into the database, and overwrites the database stored in memory. If you subsequently save your file or exit, ANSYS overwrites the boundary conditions in the database file with the D's and F's from the current results file. However, to perform a restart which ramps boundary conditions from the last solved substep, you need the boundary conditions for the last successfully solved load substep.

3.16.1.3.1. Re-establishing Boundary Conditions

To re-establish the correct boundary conditions for the restart, first run a "dummy" load step. The procedure is as follows:

1. Rename **Jobname.OSAV** as **Jobname.ESAV**.

2. Enter the ANSYS program and specify the same jobname that was used in the initial run with **/FILNAME** (**Utility Menu> File> Change Jobname**).

3. Enter the SOLUTION processor using **/SOLU** (**Main Menu> Solution**), then resume the database file using **RESUME** (**Utility Menu> File> Resume Jobname.db**).

4. Indicate that this is a restart analysis by issuing **ANTYPE**,,REST (**Main Menu> Solution> Restart**).

5. Respecify boundary conditions from the last substep that was successfully solved. One substep is sufficient since the solution will converge immediately.

6. Issue **SOLVE** (**Main Menu> Solution> Current LS** or **Main Menu> Solution> Run FLOTRAN**).

7. Apply final loads and load step options as desired. You will need to adjust the number of substeps (or time step size) if this load step is a "continuation" of the previous (before the dummy) load step. Time step numbering may be altered from your initial intent. Use a small time increment in step 6 if you need to preserve the time step numbering (such as for a transient analysis).

8. Continue the procedure as outlined in Section 3.16: Restarting an Analysis.

3.16.2. Multiframe Restart

If you are doing a nonlinear static or full transient structural analysis, ANSYS will, by default, set up the parameters for a multiframe restart. Multiframe restart allows you to save analysis information at many substeps during a run, then restart the run at one of those substeps. Before running your initial analysis, you should use the

RESCONTROL command to set up the frequency at which restart files are saved within each load step of the run.

When you need to restart a job, use the **ANTYPE** command to specify the restart point and type of restart. You can continue the job from the restart point (making any corrections necessary), or you can terminate a load step at the restart point (rescaling all of the loading) then continue with the next load step.

If you want to disable the multiframe restart feature and use the singleframe restart, issue the command **RESCONTROL**,DEFINE,NONE. Upon doing a singleframe restart (**ANTYPE**,,REST), make sure that any **.LDHI**, **.RDB**, and **.R***nnn* files are deleted from the current directory.

The sample input listing below shows how to set up the restart file parameters in an analysis then restart the analysis, continuing from a specified load step and substep.

```
!  Restart run:
/prep7
et,1,42,,,              !Define nodes and elements
mp,ex,1,10
mp,alpx,1,0.1
mp,alpy,1,0.1
mp,alpx,1,0.1
mp,nuxy,1,0.2
n,1
n,2,1
n,3,1,1
n,4,,1
n,5,2
n,6,2,1
e,1,2,3,4
e,2,5,6,3
finish
/solu
rescontrol,,all,1,5      !For all load steps, write the restart
                         !file .Rnnn at every substep, but allow
                         !a maximum of 5 restart files per load step
nlgeom,on               !Large strain analysis with temperature loadings
nsubst,2
d,1,all
d,2,uy
outres,all,all
solve
bfe,1,temp,1,1
bfe,2,temp,1,5
solve
rescontrol,file_summary  !List information contained in all the
                         !.Rnnn files for this job
finish
/post1
set,last
presol
finish
/solu
antyp,,rest,1,3          !Restart the analysis at load step 1,
                         !substep 3
solve
rescontrol,file_summary
finish
/post1
set,last
prnsol
presol
finish
```

The sample input listing below shows how to terminate a load step at a particular substep, then continue with the next load step.

```
/solu
antype,,rest,1,3,endstep  !End load step 1 at substep 3
                          !when time (load factor) = 0.6125
                          !ldstep = 1, substep = 3, load
solve                     !execute ENDSTEP, loads are
                          !rescaled to time = 0.6125
rescontrol,file_summary
bfe,1,temp,1,2            !apply higher loads,
bfe,2,temp,1,6
solve                     !execute solve to advance load
                          !factor from previous
                          !time = 0.6125 to time = 1.6125

/post1
set,last
presol
finish
```

The sample input listing below shows how to restart an analysis with old and new parameters.

```
/title, Multiframe Restart with Tabular Load.
/prep7
et,1,42        ! Build model
n,1,0.0,0.0
n,2,0.0,0.5
n,3,0.0,1.0
n,4,1.0,0.0
n,5,1.0,0.5
n,6,1.0,1.0
e, 1,4,5,2
e, 2,5,6,3
mp,ex,1,1000.0
mp,nuxy,1,0.3
mp,alpx,1,1.e-4

d,1,all
d,2,ux,0.0
d,3,ux,0.0
d,4,uy,0.0

*dim,ftbl,table,4,1,,time    ! Make tabular point load
ftbl(1,0)=1,2,3,4
ftbl(1,1)=2.5,5.0,7.5,10.0
nsel,all
f,all,fx,%ftbl%              ! Apply it to all nodes
flist

/solu
rescont,,all,all            ! Save all substeps for possible restart
nlgeo,on
time,4
DELTIM,1
outres,all,all
solve                       ! Solve with point loads and the *.RDB file is saved
                            ! at the moment. The parameterized tabular point load
                            ! FTBL is also saved into *.RDB

*dim,temtbl,table,4,1,,time  ! Define table TEMTBL and use it for body load: temperature
temtbl(1,0)=1,2,3,4
temtbl(1,1)=250,500.0,750,1000.
! bf,all,TEMP,%temtbl%       ! May use this to apply the body load table
! bflist
parsave,all,moreload         ! Save all the APDL parameters and tables to file: moreload
                             ! NOTE: *.RDB does not have information of table TEMTBL.
fini

/clear, nostart
/solu
ANTYPE,,RESTART,1,3,endstep  ! Do restart ENDSTEP because we want to apply TEMTBL at
                             ! TIME = 3.5 (LDSTEP=1,SUBSTEP=3) because we want to
                             ! Apply the temperature load from TIME=3.5 onwards.
                             ! Here, RESTART has resumed *.RDB database where the
```

```
                          ! Table FTBL is saved.

solve                     ! Activate ENDSTEP

parresu,,moreload         ! For further load step, we want to apply table TEMTBL
                          ! as body force. NOTE: table TEMTBL is not in *.RDB. Therefore,
                          ! we have to use PARRESU command. APDL file "moreload" is
                          ! saved earlier.

*status                   ! List all the ADPL information available at this point
bf,all,TEMP,%temtbl%      ! Apply temperature table load TEMTBL
bflist
time,4                    ! Solve up to TIME = 4.0 because the load step ENDSTEP only
                          ! carries up to TIME = 3.5
solve
fini
/post1
set,last
prdis
prrsol
fini
```

The sample below demonstrates a restart after changing boundary conditions.

```
/prep7
et,1,21
r,1,1,1,1,1,1,1
n,1
e,1
fini

/solu
antyp,trans
timint,off
time,.1
nsub,2
kbc,0
d,1,ux,100                ! to apply initial velocity (IC command is preferred)
solve


timint,on
ddele,1,ux                ! this requires special handling by multi-frame restart
                              ! if a reaction force exists at this dof, replace it with an equal
                              ! force using the endstop option
time,.2
nsub,5
rescontrol,define,all,1   ! request possible restart from any substep
outres,nsol,1
solve
fini

/solu
antyp,,restart,2,3        ! this command resumes the .rdb database created at the start of solution
                          ! (restart from substep 3)
ddele,1,ux                ! re-specify boundary condition deleted during solution
solve
fini

/post26
nsol,2,1,ux
prvar,2                   ! results show constant velocity through restart
fini
/exit
```

Note — If you are using the **Solution Controls** dialog box to do a static or full transient analysis, you can specify basic multiframe restart options on the dialog's **Sol'n Options** tab. These options include the maximum number of restart files that you want ANSYS to write for a load step, as well as how frequently you want the files to be written. For an overview of the **Solution Controls** dialog box, see Section 3.11: Using Special Solution Controls for Certain Types of Structural Analyses. For details about how to set options on the **Solution Controls** dialog box, access the dialog box (**Main Menu> Solution> Sol'n Control**), select the tab that you are interested in, and click the **Help** button.

3.16.2.1. Multiframe Restart Requirements

The following files are necessary to do a multiframe restart:

- **Jobname.RDB** - This is an ANSYS database file saved automatically at the first iteration of the first load step, first substep of a job. This file provides a complete description of the solution with all initial conditions, and will remain unchanged regardless of how many restarts are done for a particular job. When running a job, you should input all information needed for the solution - including parameters (APDL), components, and mandatory solution setup information - before you issue the first **SOLVE**. If you do not specify parameters before issuing the first **SOLVE** command, the parameters will not be saved in the **.RDB** file. In this case, you must use **PARSAV** before you begin the solution and **PARRES** during the restart to save and restore the parameters. If the information stored in the **.RDB** file is not sufficient to perform the restart, you must input the additional information in the restart session before issuing the **SOLVE** command.

- **Jobname.LDHI** - This is the load history file for the specified job. This file is an ASCII file similar to files created by **LSWRITE** and stores all loading and boundary conditions for each load step. The loading and boundary conditions are stored for the FE mesh. Loading and boundary conditions applied to the solid model are transferred to the FE mesh before storing in the **Jobname.LDHI**. When doing a multiframe restart, ANSYS reads the loading and boundary conditions for the restart load step from this file (similar to an **LSREAD** command). In general, you need the loading and boundary conditions for two contiguous load steps because of the ramped load conditions for a restart. You cannot modify this file because any modifications may cause an unexpected restart condition. This file is modified at the end of each load step or when an **ANTYPE,,REST,**$LDSTEP$**,**$SUBSTEP$**,ENDSTEP** command is encountered. For tabular loads or boundary conditions, you should ensure that the APDL parameter tables are available at restart.

- **Jobname.R**nnn - This file contains element saved records similar to the **.ESAV** or **.OSAV** files. This file also contains all solution commands and status for a particular substep of a load step. All of the **.R**nnn files are saved at the converged state of a substep so that all element saved records are valid. If a substep does not converge, no **.R**nnn file will be written for that substep. Instead, a **.R**nnn file from a previously converged substep is written.

Multiframe restart has the following limitations:

- It does not support **KUSE**. A new stiffness matrix and the related **.TRI** file will be regenerated.

- The **.R**nnn file does not save the **EKILL** and **EALIVE** commands. If **EKILL** or **EALIVE** commands are required in the restarted session, you must reissue these commands.

- The **.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 restart session. This situation often occurs when parameters are used (APDL). 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 during the restart session in this case.

- You cannot restart a job at the equation solver level (for example, the PCG iteration level). The job can only be restarted at a substep level (either transient or Newton-Raphson loop).

- Multiframe restart does not support the ENDSTEP option of **ANTYPE** when the arc-length method is employed.

- All loading and boundary conditions are stored in the **Jobname.LDHI** file; therefore, upon restart, removing or deleting solid modeling loading and boundary conditions will not result in the removal of these conditions from the finite element model. You must remove these conditions directly from nodes and elements.

3.16.2.2. Multiframe Restart Procedure

Use the following procedure to restart an analysis:

1. Enter the ANSYS program and specify the same jobname that was used in the initial run. To do so, issue the **/FILNAME** command (**Utility Menu> File> Change Jobname**). Enter the SOLUTION processor using **/SOLU** (**Main Menu> Solution**).

2. Determine the load step and substep at which to restart by issuing **RESCONTROL**, FILE_SUMMARY. This command will print the substep and load step information for all **.R***nnn* files in the current directory.

3. Resume the database file and indicate that this is a restart analysis by issuing **ANTYPE**,,REST,*LDSTEP*,*SUBSTEP*,*Action* (**Main Menu> Solution> Restart**).

4. Specify revised or additional loads as needed. Be sure to take whatever corrective action is necessary if you are restarting from a convergence failure.

5. Initiate the restart solution by issuing the **SOLVE** command. (See Section 3.13: Obtaining the Solution for details.) You must issue the **SOLVE** command when taking any restart action, including ENDSTEP or RSTCREATE.

6. Postprocess as desired, then exit the ANSYS program.

If the files **Jobname.LDHI** and **Jobname.RDB** exist, the **ANTYPE**,,REST command will cause ANSYS to do the following:

- Resume the database **Jobname.RDB**

- Rebuild the loading and boundary conditions from the **Jobname.LDHI** file

- Rebuild the ANSYS solution commands and status from the **.R***nnn* file.

At this point, you can enter other commands to overwrite input restored by the **ANTYPE** command.

> *Note* — The loading and boundary conditions restored from the **Jobname.LDHI** are for the FE mesh. The solid model loading and boundary conditions are not stored on the **Jobname.LDHI**.

After the job is restarted, the files are affected in the following ways:

- The **.RDB** file is unchanged.

- All information for load steps and substeps past the restart point is deleted from the **.LDHI** file. Information for each new load step is then appended to the file.

- All of the **.R***nnn* files that have load steps and substeps earlier than the restart point will be kept unchanged. Those files containing load steps and substeps beyond the restart point will be deleted before the restart solution begins in order to prevent file conflicts.

- the **.RST** file is updated according to the restart. All results from load steps and substeps later than the restart point will be deleted from the file to prevent conflicts, and new information from the solution will be appended to the end of the **.RST** file.

When a job is started from the beginning again (first substep, first load step), all of the restart files (**.RDB**, **.LDHI**, **.R***nnn*) in the current directory for the current jobname will be deleted before the new solution begins.

ANTYPE,,REST,*LDSTEP*,*SUBSTEP*,RSTCREATE can be used to create a results file for a particular load step and substep of an analysis. It must be used in conjunction with **OUTRES** to write the results. A RSTCREATE session will not update or delete any of the restart files, allowing you to use RSTCREATE for any number of saved points in a session.

The sample input listing below shows how to create a results file for a particular substep in an analysis.

```
!  Restart run:
/solu
antype,,rest,1,3,rstcreate !Create a .RST file from load
    !step 1, substep 3
outres,all,all  !Store everything into the .RST file
outpr,all,all  !Optional for printed output
solve    !Execute the .RST file creation
finish
/post1
set,,1,3  !Get results from load step 1,
    !substep 3
prnsol
finish
```

3.17. Exercising Partial Solution Steps

When you initiate a solution, the ANSYS program goes through a predefined series of steps to calculate the solution; it formulates element matrices, triangularizes matrices, and so on. Another SOLUTION command, **PSOLVE**, (**Main Menu> Solution> Partial Solu**) allows you to exercise each such step individually, completing just a portion of the solution sequence each time. For example, you can stop at the element matrix formulation step and go down a different path to perform inertia relief calculations. Or, you can stop at the Guyan reduction step (matrix reduction) and go on to calculate reduced eigenvalues.

Some possible uses of the **PSOLVE** approach are listed below.

- You can use it as a restart tool for singleframe restarts. For instance, you can start from the **.EMAT** file and perform a different analysis.

- You can use it to perform a prestressed modal analysis of a large deflection static solution.

- You can use the results of an intermediate solution step (such as the triangularized matrix) as input to another software package or user-written program.

- If you are interested just in inertia relief calculations or some such intermediate result, the **PSOLVE** approach is useful. See the *ANSYS Structural Analysis Guide* for more information.

The following examples show how you can use partial solution steps with direct and iterative solvers, respectively:

```
! This outline describes how to use the PSOLVE commands in combination
! with a direct solver (the Frontal solver in this case).
!
/prep7
!
! define model geometry
! define model materials
!
fini
/solu
!
! apply boundary conditions
! define psolve commands
!
psolve,elform     ! Note the psolve commands in a single load step
```

```
psolve,elprep
psolve,triang
!
fini            ! Finish required to end psolve load step
/solu
kuse,1          ! Needed for stiffness created by the psolve commands used
outres,all,all
solve
fini            ! End of direct solver outline
!
! This outline describes how to use the PSOLVE commands in combination
! with an iterative solver (the JCG solver in this case).
! Although documented to work, using the PSOLVE commands with an
! iterative solver is not likely to decrease solution-processing time.
!
/prep7
!
! Define model geometry
! Define model materials
!
fini
/solu
!
! Apply boundary conditions
! Define psolve commands
!
eqslv,jcg       ! This first load step can be any load step which creates a
solve           ! jobname.full file.  If a jobname.full exists, this load step
fini            ! is unnecessary.
/solu
psolve,cgsol    ! Need jobname.full in place before psolve,cgsol
outres,all,all
fini            ! End of iterative solver outline
```

3.18. Estimating Run Time and File Sizes

For "small to moderately sized" ANSYS models with little added analysis complexity, you will initiate the solution as indicated previously in this chapter. However, for larger models, or for analyses with complicated nonlinear options, you may feel more comfortable knowing what to expect before initiating solution. For example, how long will your analysis take to solve and how much disk space will you need before you run it? How much memory will your analysis need? While there is no *exact* method of predicting these quantities, the ANSYS program can *estimate* them in its RUNSTAT module.

The RUNSTAT module is a processor, or routine, of its own. You can enter it by issuing the **/RUNST** (**Main Menu> Run-Time Stats**) command at the Begin level.

The RUNSTAT module estimates run times and other statistics based on information in the database. Therefore, you must define the model geometry (nodes, elements, etc.), loads and load options, and analysis options before you enter RUNSTAT. It is best to use RUNSTAT immediately before solving.

3.18.1. Estimating Run Time

To estimate the run time, the ANSYS program needs your computer's performance information: MIPS (millions of instructions per second), MFLOPs (millions of floating point operations per second), etc. To obtain this inform-ation, use the **RSPEED** command (**Main Menu> Run-Time Stats> System Settings**).

If you do not know such details about your computer, you can execute the macro **SETSPEED**, which issues the appropriate **RSPEED** command for you. For more information about the **SETSPEED** macro, see Section 3.7: Es-timating ANSYS Run Time in the *ANSYS Operations Guide*.

The other piece of information needed to estimate the total run time is the number of iterations (or load steps in a linear, static analysis) for the analysis. To get this information, use either method shown below:

   Command(s): RITER
   GUI: Main Menu> Run-Time Stats> Iter Setting

To obtain an estimate of run time, use one of the following:
   Command(s): RTIMST
   GUI: Main Menu> Run-Time Stats> Individual Stats

Based on information supplied by the **RSPEED** and **RITER** commands and the model information in the database, the **RTIMST** command gives you a run time estimate.

3.18.2. Estimating File Size

The **RFILSZ** command estimates the size of the **.ESAV**, **.EMAT**, **.EROT**, **.TRI**, **.FULL**, **.RST**, **.RTH**, **.RMG**, and **.RFL** files. The GUI equivalent for **RFILSZ** is the same as for **RTIMST**. The estimate of the results files is based on one set of results (one substep). You will need to multiply it by the estimated number of results written for the actual results file sizes.

3.18.3. Estimating Memory Requirements

The **RWFRNT** command (**Main Menu> Run-Time Stats> Individual Stats**) gives wavefront estimates and memory requirements for solution. You can then request that amount of memory using the ANSYS work space entry option. (See Chapter 19, "Memory Management and Configuration" for more information about memory management.) The **RWFRNT** command automatically reorders the elements if no reordering has previously been done.

Other RUNSTAT commands are **RSTAT**, which gives model statistics (node and element information); **RMEMRY**, which gives memory statistics; and **RALL** (**Main Menu> Run-Time Stats> All Statistics**), which is a convenience command that executes the **RSTAT**, **RWFRNT**, **RTIMST**, and **RMEMRY** commands. The GUI equivalent for all these commands except for **RALL** is **Main Menu> Run-Time Stats> Individual Stats**.

3.19. Singularities

A singularity exists in an analysis whenever an indeterminate or non-unique solution is possible. A negative or zero equation solver pivot value will yield such a solution. In some instances, it may be desirable to continue the analysis, even though a negative or zero pivot value is encountered. You can use the **PIVCHECK** command to specify whether or not to stop the analysis when this occurs.

The default value for the **PIVCHECK** command is ON. With **PIVCHECK** set to ON, a linear static analysis (in batch mode only) stops when a negative or zero pivot value is encountered. The message "NEGATIVE PIVOT VALUE" or "PIVOTS SET TO ZERO" is displayed. If **PIVCHECK** is set to OFF, the pivots are not checked. Set **PIVCHECK** to OFF if you want your batch mode linear static analysis to continue in spite of a zero or negative pivot value. The **PIVCHECK** setting has no effect for nonlinear analyses, since a negative or zero pivot value can occur for a valid analysis. When **PIVCHECK** is set to OFF, ANSYS automatically increases any pivot value smaller than machine "zero" to a value between 10 and 100 times that machine's "zero" value. Machine "zero" is a tiny number the machine uses to define "zero" within some tolerance. This value varies for different computers (approx1e-15).

The following conditions may cause singularities in the solution process:

- Insufficient constraints.

- Nonlinear elements in a model (such as gaps, sliders, hinges, cables, etc.). A portion of the structure may have collapsed or may have "broken loose."

- Negative values of material properties, such as DENS or C, specified in a transient thermal analysis.

- Unconstrained joints. The element arrangements may cause singularities. For example, two horizontal spar elements will have an unconstrained degree of freedom in the vertical direction at the joint. A linear analysis would ignore a vertical load applied at that point. Also, consider a shell element with no in-plane rotational stiffness connected perpendicularly to a beam or pipe element. There is no in-plane rotational stiffness at the joint. A linear analysis would ignore an in-plane moment applied at that joint.

- Buckling. When stress stiffening effects are negative (compressive) the structure weakens under load. If the structure weakens enough to effectively reduce the stiffness to zero or less, a singularity exists and the structure has buckled. The "NEGATIVE PIVOT VALUE - " message will be printed.

- Zero Stiffness Matrix (on row or column). Both linear and nonlinear analyses will ignore an applied load if the stiffness is exactly zero.

3.20. Stopping Solution After Matrix Assembly

You can terminate the solution process after the assembled global matrix file (**.FULL** file) has been written by using **WRFULL**. By doing so, the equation solution process and the process of writing data to the results file are skipped. This feature can then be used in conjunction with **HBMAT** in /AUX2 to dump any of the assembled global matrices into a new file that is written in Harwell-Boeing format.

Note — The **WRFULL** command is only valid for linear static, full harmonic, and full transient analyses when the sparse direct solver is selected. **WRFULL** is also valid for buckling and modal analyses when any mode extraction method is selected. This command is not valid for nonlinear analyses or analyses containing p-elements.

Chapter 4: An Overview of Postprocessing

4.1. What Is Postprocessing?

After building the model and obtaining the solution, you will want answers to some critical questions: Will the design really work when put to use? How high are the stresses in this region? How does the temperature of this part vary with time? What is the heat loss across this face of my model? How does the magnetic flux flow through this device? How does the placement of this object affect fluid flow? The postprocessors in the ANSYS program can help you answer these questions and others.

Postprocessing means reviewing the results of an analysis. It is probably the most important step in the analysis, because you are trying to understand how the applied loads affect your design, how good your finite element mesh is, and so on.

Two postprocessors are available to review your results: POST1, the general postprocessor, and POST26, the time-history postprocessor. POST1 allows you to review the results over the entire model at specific load steps and substeps (or at specific time-points or frequencies). In a static structural analysis, for example, you can display the stress distribution for load step 3. Or, in a transient thermal analysis, you can display the temperature distribution at time = 100 seconds. The contour plot in Figure 4.1: "A Typical POST1 Contour Display" is a typical POST1 plot.

Figure 4.1 A Typical POST1 Contour Display
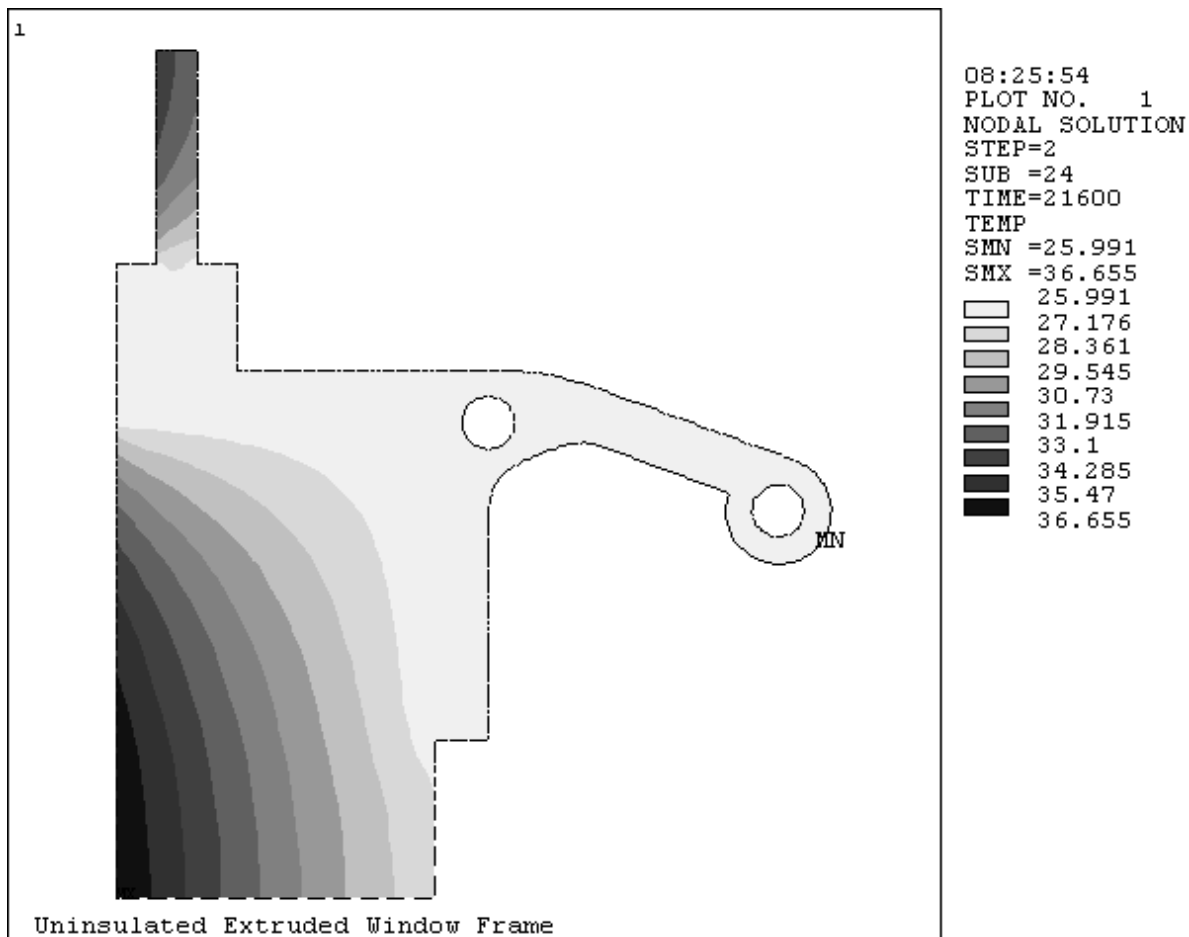
POST26 allows you to review the variation of a particular result item at specific points in the model with respect to time, frequency, or some other result item. In a transient magnetic analysis, for instance, you can graph the eddy current in a particular element versus time. Or, in a nonlinear structural analysis, you can graph the force at a particular node versus its deflection. Figure 4.2: "A Typical POST26 Graph" is shown below.

Figure 4.2 A Typical POST26 Graph
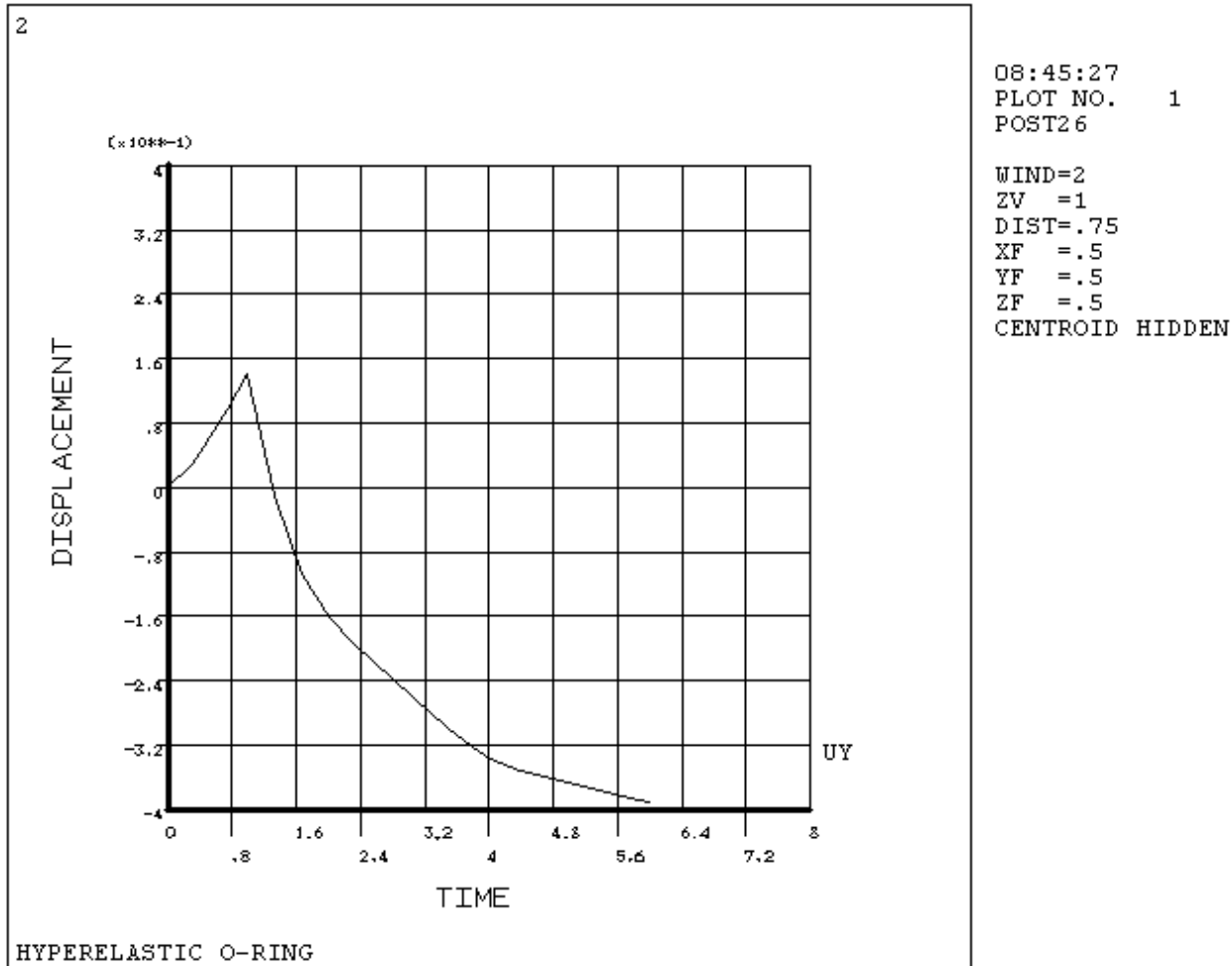
It is important to remember that the postprocessors in ANSYS are just *tools* for reviewing analysis results. You still need to use your engineering judgment to *interpret* the results. For example, a contour display may show that the highest stress in the model is 37,800 psi. It is now up to you to determine whether this level of stress is acceptable for your design.

4.2. The Results Files

You can use **OUTRES** to direct the ANSYS solver to append selected results of an analysis to the *results file* at specified intervals during solution. The name of the results file depends on the analysis discipline:

- **Jobname.RST** for a structural analysis
- **Jobname.RTH** for a thermal analysis
- **Jobname.RMG** for a magnetic field analysis
- **Jobname.RFL** for a FLOTRAN analysis

For a FLOTRAN analysis, the file extension is **.RFL**. For other fluid analyses, the file extension is **.RST** or **.RTH**, depending on whether structural degrees of freedom are present. (Using different file identifiers for different disciplines helps you in coupled-field analyses where the results from one analysis are used as loads for another. The *ANSYS Coupled-Field Analysis Guide* presents a complete description of coupled-field analyses.)

4.3. Types of Data Available for Postprocessing

The solution phase calculates two types of results data:

- *Primary data* consist of the degree-of-freedom solution calculated at each node: displacements in a structural analysis, temperatures in a thermal analysis, magnetic potentials in a magnetic analysis, and so on (see Table 4.1: "Primary and Derived Data for Different Disciplines"). These are also known as nodal solution data.

- *Derived data* are those results calculated from the primary data, such as stresses and strains in a structural analysis, thermal gradients and fluxes in a thermal analysis, magnetic fluxes in a magnetic analysis, and the like. They are typically calculated for each element and may be reported at any of the following locations: at all nodes of each element, at all integration points of each element, or at the centroid of each element. Derived data are also known as element solution data, except when they are averaged at the nodes. In such cases, they become nodal solution data.

Table 4.1 Primary and Derived Data for Different Disciplines

Discipline	Primary Data	Derived Data
Structural	Displacement	Stress, strain, reaction, etc.
Thermal	Temperature	Thermal flux, thermal gradient, etc.
Magnetic	Magnetic Potential	Magnetic flux, current density, etc.
Electric	Electric Scalar Potential	Electric field, flux density, etc.
Fluid	Velocity, Pressure	Pressure gradient, heat flux, etc.

Chapter 5: The General Postprocessor (POST1)

5.1. Overview

Use POST1, the general postprocessor, to review analysis results over the entire model, or selected portions of the model, for a specifically defined combination of loads at a single time (or frequency). POST1 has many capabilities, ranging from simple graphics displays and tabular listings to more complex data manipulations such as load case combinations.

To enter the ANSYS general postprocessor, issue the **/POST1** command (**Main Menu> General Postproc**).

5.2. Reading Results Data into the Database

The first step in POST1 is to read data from the results file into the database. To do so, model data *model data* (nodes, elements, etc.) must exist in the database. If the database does not already contain model data, issue the **RESUME** command (**Utility Menu> File> Resume Jobname.db**) to read the database file, **Jobname.DB**. The database should contain the *same* model for which the solution was calculated, including the element types, nodes, elements, element real constants, material properties, and nodal coordinate systems.

> **Caution:** The database should contain the same set of selected nodes and elements that were selected for the solution. Otherwise, a data mismatch may occur. For more information about data mismatches, see Section 5.2.2.3: Appending Data to the Database.

After model data are in the database, load the results data from the results file by issuing one of the following commands: **SET**, **SUBSET**, or **APPEND**.

5.2.1. Reading in Results Data

The **SET** command (**Main Menu> General Postproc> Read Results> *datatype***) reads results data over the entire model from the results file into the database for a particular loading condition, replacing any data previously stored in the database. The boundary condition information (constraints and force loads) is also read in, *but only if either element nodal loads or reaction loads are available*; see the **OUTRES** command for more information. If they are not available, no boundary conditions will be available for listing or plotting. Only constraints and forces are read in; surface and body loads are not updated and remain at their last specified value. However, if the surface and body loads have been specified using tabular boundary conditions, they will reflect the values corresponding to this results set. Loading conditions are identified either by load step and substep or by time (or frequency). The arguments specified with the command or path identify the data to be read into the database. For example, **SET**,2,5 reads in results for load step 2, substep 5. Similarly, **SET**,,,,,3.89 reads in results at time = 3.89 (or frequency = 3.89 depending on the type of analysis that was run). If you specify a time for which no results are available, the program performs linear interpolation to calculate results at that time.

The default maximum number of substeps in the results file (**Jobname.RST**) is 1000. When the number of substeps exceeds this limit, you need to issue **SET**,*Lstep*,LAST to bring in the 1000th load step. Use **/CONFIG** to increase the limit.

> **Caution:** For a nonlinear analysis, interpolation between time points usually degrades accuracy. Therefore, take care to postprocess at a time value for which a solution is available.

Some convenience labels are also available on **SET**:

- **SET**,FIRST reads in the first substep. The GUI equivalent is **Main Menu> General Postproc> Read Results> First Set**.

- **SET**,NEXT reads in the next substep. The GUI equivalent is **Main Menu> General Postproc> Read Results> Next Set**.

- **SET**,LAST reads in the last substep. The GUI equivalent is **Main Menu> General Postproc> Read Results> Last Set**.

- The *NSET* field on the **SET** command (GUI equivalent is **Main Menu> General Postproc> Read Results> By Set Number**) retrieves data that corresponds to its unique data set number, rather than its load step and substep number. This is extremely useful with FLOTRAN results, because you can have multiple sets of results data with identical load step and substep numbers. Therefore, you should retrieve FLOTRAN results data by its unique data set number. The LIST option on the **SET** command (or **Main Menu> General Postproc> List Results** in the GUI) lists the data set number along with its corresponding load step and substep numbers. You can enter this data set number on the *NSET* field of the next **SET** command to request the proper set of results.

- The *ANGLE* field on **SET** specifies the circumferential location for harmonic elements (structural - PLANE25, PLANE83, and SHELL61; thermal - PLANE75 and PLANE78).

5.2.2. Other Options for Retrieving Results Data

Other GUI paths or commands also enable you to retrieve results data.

5.2.2.1. Defining Data to be Retrieved

The **INRES** command (**Main Menu> General Postproc> Data & File Opts**) in POST1 is a companion to the **OUTRES** command in the PREP7 and SOLUTION processors. Where the **OUTRES** command controls data written *to* the database and the results file, the **INRES** command defines the type of data to be retrieved *from* the results file for placement into the database through commands such as **SET**, **SUBSET**, and **APPEND**. Although not required for postprocessing of data, the **INRES** command limits the amount of data retrieved and written to the database. As a result, postprocessing your data may take less time.

5.2.2.2. Reading Selected Results Information

To read a data set from the results file into the database for the *selected* portions of the model only, use the **SUBSET** command (**Main Menu> General Postproc> Read Results> `By characteristic`**). Data that has not been specified for retrieval from the results file by the **INRES** command will be listed as having a zero value.

The **SUBSET** command behaves like the **SET** command except that it retrieves data for the selected portions of the model only. It is very convenient to use the **SUBSET** command to look at the results data for a *portion* of the model. For example, if you are interested only in surface results, you can easily select the exterior nodes and elements, and then use **SUBSET** to retrieve results data for just those selected items.

5.2.2.3. Appending Data to the Database

Each time you use **SET**, **SUBSET**, or their GUI equivalents, ANSYS writes a new set of data over the data currently in the database. The **APPEND** command (**Main Menu> General Postproc> Read Results> `By characteristic`**) reads a data set from the results file and merges it with the existing data in the database, for the *selected model only*. The existing database is not zeroed (or overwritten in total), allowing the requested results data to be merged into the database.

You can use any of the commands **SET**, **SUBSET**, or **APPEND** to read data from the results file into the database. The only difference between the commands or paths is *how much* or *what type* of data you wish to retrieve. When

appending data, be very careful not to generate a data mismatch inadvertently. For example, consider the following set of commands:

```
/POST1
INRES,NSOL            ! Flag data from nodal DOF solution
NSEL,S,NODE,,1,5      ! Select nodes 1 to 5
SUBSET,1              ! Write data from load step 1 to database
```

At this point results data for nodes 1 to 5 from load step 1 are in the database.

```
NSEL,S,NODE,,6,10     ! Select nodes 6 to 10
APPEND,2              ! Merge data from load step 2 into database
NSEL,S,NODE,,1,10     ! Select nodes 1 to 10
PRNSOL,DOF            ! Print nodal DOF solution results
```

The database now contains data for *both* load steps 1 and 2. This is a data mismatch. When you issue the **PRNSOL** command (**Main Menu> General Postproc> List Results> Nodal Solution**), the program informs you that you will have data from the second load step, when actually data from two different load steps now exist in the database. The load step listed by the program is merely the one corresponding to the most *recent* load step stored. Of course, appending data to the database is very helpful if you wish to compare results from different load steps, but if you purposely intend to mix data, it is extremely important to keep track of the *source* of the data appended.

To avoid data mismatches when you are solving a subset of a model that was solved previously using a different set of elements, do either of the following:

  · Do not reselect any of the elements that were unselected for the solution currently being postprocessed.

  · Remove the earlier solution from the ANSYS database. You can do so by exiting from ANSYS between solutions or by saving the database between solutions.

For more information, see the *ANSYS Commands Reference* for descriptions of the **INRES**, **NSEL**, **APPEND**, **PRNSOL**, and **SUBSET** commands.

If you wish to clear the database of any previous data, use one of the following methods:
>   **Command(s): LCZERO**
>   **GUI: Main Menu> General Postproc> Load Case> Zero Load Case**

Either method sets all current values in the database to zero, therefore giving you a fresh start for further data storage. If you set the database to zero before appending data to it, the result is the same as using the **SUBSET** command or the equivalent GUI path, assuming that the arguments on **SUBSET** and **APPEND** are equivalent.

  Note — All of the options available for the **SET** command are also available for the **SUBSET** and **APPEND** commands.

By default, the **SET**, **SUBSET**, and **APPEND** commands look for one of these results files: **Jobname.RST**, **Jobname.RTH**, **Jobname.RMG**, or **Jobname.RFL**. You can specify a different file name by issuing the **FILE** command (**Main Menu> General Postproc> Data & File Opts**) before issuing **SET**, **SUBSET**, or **APPEND**.

5.2.3. Creating an Element Table

In the ANSYS program, the *element table* serves two functions. First, it is a tool for performing arithmetic operations among results data. Second, it allows access to certain element results data that are not otherwise directly accessible, such as derived data for structural line elements. (Although the **SET**, **SUBSET**, and **APPEND** commands read all requested results items into the database, not all data are directly accessible with commands such as **PLNSOL**, **PLESOL**, etc.).

Think of the element table as a spreadsheet, where each row represents an element, and each column represents a particular data item for the elements. For example, one column might contain the average SX stress for the elements, while another might contain the element volumes, while yet a third might contain the Y coordinate of the centroid for each element.

To create or erase the element table, use one of the following:

Command(s): ETABLE
GUI: Main Menu> General Postproc> Element Table> Define Table
Main Menu> General Postproc> Element Table> Erase Table

5.2.3.1. Filling the Element Table for Variables Identified By Name

To identify an element table column, you assign a label to it using the *Lab* field (GUI) or the *Lab* argument on the **ETABLE** command. This label will be used as the identifier for all subsequent POST1 commands involving this variable. The data to go into the columns is identified by an *Item* name and a *Comp* (component) name, the other two arguments on the **ETABLE** command. For example, for the SX stresses mentioned above, SX could be the *Lab*, S would be the *Item*, and X would be the *Comp* argument.

Some items, such as the element volumes, do not require *Comp*; in such cases, *Item* is VOLU and *Comp* is left blank. Identifying data items by an *Item*, and *Comp* if necessary, is called the "Component Name" method of filling the element table. The data which are accessible with the component name method are data generally calculated for most element types or groups of element types.

The **ETABLE** command documentation lists, in general, all the *Item* and *Comp* combinations. However see the "Element Output Definitions" table in each element description in the *ANSYS Elements Reference* to see which combinations are valid. Table 5.1: "3-D BEAM4 Element Output Definitions" is an example of such a table for BEAM4. You can use any name in the **Name** column of the table that contains a colon (:) to fill the element table via the Component Name method. The portion of the name *before* the colon should be input for the *Item* argument of the **ETABLE** command. The portion (if any) *after* the colon should be input for the *Comp* argument. The **O** and **R** columns indicate the availability of the items in the file **Jobname.OUT** (**O**) or in the results file (**R**): a **Y** indicates that the item is *always* available, a number refers to a table footnote which describes when the item is *conditionally* available, and a - indicates that the item is *not* available.

Table 5.1 3-D BEAM4 Element Output Definitions

Name	Definition	O	R
EL	Element number	Y	Y
NODES	Element node number (I and J)	Y	Y
MAT	Material number for the element	Y	Y
VOLU:	Element volume	-	Y
CENT:(X, Y, Z)	Global location of the element centroid XC, YC, ZC	-	Y
TEMP	Temperatures at integration points T1, T2, T3, T4, T5, T6, T7, T18	Y	Y
PRES	Pressure P1 at nodes I,J; OFFST1 at I,J; P2 at I,J; OFFST2 at I,J; P3 at I,J; OFFST3 at I,J; P4 at I; P5 at J	Y	Y
SDI R	Axial direct stress	1	1
SBYT	Bending stress on the element +Y side of the beam	1	1
SBYB	Bending stress on the element -Y side of the beam	1	1
SBZT	Bending stress on the element +Z side of the beam	1	1
SBZB	Bending stress on the element -Z side of the beam	1	1

Name	Definition	O	R
SMAX	Maximum stress (direct stress + bending stress)	1	1
SMIN	Minimum stress (direct stress - bending stress)	1	1
EPELDIR	Axial elastic strain at the end	1	1
EPELBYT	Bending elastic strain on the element +Y side of the beam	1	1
EPELBYB	Bending elastic strain on the element -Y side of the beam	1	1
EPELBZT	Bending elastic strain on the element +Z side of the beam	1	1
EPELBZB	Bending elastic strain on the element -Z side of the beam	1	1
EPTHDIR	Axial thermal strain at the end	1	1
EPTHBYT	Bending thermal strain on the element +Y side of the beam	1	1
EPTHBYB	Bending thermal strain on the element -Y side of the beam	1	1
EPTHBZT	Bending thermal strain on the element +Z side of the beam	1	1
EPTHBZB	Bending thermal strain on the element -Z side of the beam	1	1
EPINAXL	Initial axial strain in the element	1	1
MFOR:(X,Y,Z)	Member forces in the element coordinate system X, Y, Z directions	2	Y
MMOM:(X,Y,Z)	Member moments in the element coordinate system X, Y, Z directions	2	Y

1. The item repeats for end I, intermediate locations (see KEYOPT(9)), and end J.
2. If KEYOPT(6) = 1.

5.2.3.2. Filling the Element Table for Variables Identified By Sequence Number

You can load data that is not averaged, or that is not naturally single-valued for each element, into the element table. This type of data includes integration point data, *all* derived data for structural *line* elements (such as spars, beams, and pipes) and contact elements, *all* derived data for thermal line elements, layer data for layered elements, etc. These data are listed in the "Item and Sequence Numbers for the **ETABLE** and **ESOL** Commands" table with each element type description in the *ANSYS Commands Reference*. Table 5.2: "BEAM4 (KEYOPT(9) = 0) Item and Sequence Numbers for the ETABLE and ESOL Commands" is an example of such a table for BEAM4.

The data in the tables is broken down into item groups (such as LS, LEPEL, SMISC, etc.). Each item within the item group has an identifying "sequence" number listed. You load these data into the element table by giving the item group (such as LS, LEPEL, SMISC, etc.) as the *Item* argument on the **ETABLE** command, and the sequence number as the *Comp* argument. This is referred to as the "Sequence Number" method of filling the element table. For example, the maximum stress at node J for BEAM4 is *Item* = NMISC and *Comp* = 3, while the initial axial strain (EPINAXL) for the element (E) is *Item* = LEPTH and *Comp* = 11.

Table 5.2 BEAM4 (KEYOPT(9) = 0) Item and Sequence Numbers for the ETABLE and ESOL Commands

KEYOPT(9) = 0				
Label	Item	E	I	J
SDIR	LS	-	1	6
SBYT	LS	-	2	7
SBYB	LS	-	3	8
SBZT	LS	-	4	9
SBZB	LS	-	5	10

KEYOPT(9) = 0				
Label	**Item**	**E**	**I**	**J**
EPELDIR	LEPEL	-	1	6
EPELBYT	LEPEL	-	2	7
EPELBYB	LEPEL	-	3	8
EPELBZT	LEPEL	-	4	9
EPELBZB	LEPEL	-	5	10
SMAX	NMISC	-	1	3
SMIN	NMISC	-	2	4
EPTHDIR	LEPTH	-	1	6
EPTHBYT	LEPTH	-	2	7
EPTHBYB	LEPTH	-	3	8
EPTHBZT	LEPTH	-	4	9
EPTHBZB	LEPTH	-	5	10
EPINAXL	LEPTH	11	-	-
MFORX	SMISC	-	1	7
MFORY	SMISC	-	2	8
MFORZ	SMISC	-	3	9
MMOMX	SMISC	-	4	10
MMOMY	SMISC	-	5	11
MMOMZ	SMISC	-	6	12
P1	SMISC	-	13	14
OFFST1	SMISC	-	15	16
P2	SMISC	-	17	18
OFFST2	SMISC	-	19	20
P3	SMISC	-	21	22
OFFST3	SMISC	-	23	24
P4	SMISC	-	25	-
P5	SMISC	-	-	26

		Pseudo Node							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
TEMP	LBFE	1	2	3	4	5	6	7	8

For some line elements, such as BEAM4, KEYOPT settings govern the amount of data calculated. This can change the sequence number of a particular data item. Therefore, in these cases a table for each KEYOPT setting is provided. Table 5.3: "BEAM4 (KEYOPT(9) = 3) Item and Sequence Numbers for the ETABLE and ESOL Commands" shows the same information for BEAM4 as Table 5.2: "BEAM4 (KEYOPT(9) = 0) Item and Sequence Numbers for the ETABLE and ESOL Commands", but with sequence numbers as they look when KEYOPT(9) = 3 (3 intermediate calculation points). For example, with KEYOPT(9) = 0, the member moment (MMOM) in the y direction at end J of the element is sequence number 11 (item SMISC) in Table 5.2: "BEAM4 (KEYOPT(9) = 0) Item and Sequence Numbers for the ETABLE and ESOL Commands", while with KEYOPT(9) = 3 (Table 5.3: "BEAM4 (KEYOPT(9) = 3) Item and Sequence Numbers for the ETABLE and ESOL Commands") the sequence number is 29.

Table 5.3 BEAM4 (KEYOPT(9) = 3) Item and Sequence Numbers for the ETABLE and ESOL Commands

Label	Item	E	I	IL1	IL2	IL3	J
KEYOPT(9) = 3							
SDIR	LS	-	1	6	11	16	21
SBYT	LS	-	2	7	12	17	22
SBYB	LS	-	3	8	13	18	23
SBZT	LS	-	4	9	14	19	24
SBZB	LS	-	5	10	15	20	25
EPELDIR	LEPEL	-	1	6	11	16	21
EPELBYT	LEPEL	-	2	7	12	17	22
EPELBYB	LEPEL	-	3	8	13	18	23
EPELBZT	LEPEL	-	4	9	14	19	24
EPELBZB	LEPEL	-	5	10	15	20	25
SMAX	NMISC	-	1	3	5	7	9
SMIN	NMISC	-	2	4	6	8	10
EPTHDIR	LEPTH	-	1	6	11	16	21
EPTHBYT	LEPTH	-	2	7	12	17	22
EPTHBYB	LEPTH	-	3	8	13	18	23
EPTHBZT	LEPTH	-	4	9	14	19	24
EPTHBZB	LEPTH	-	5	10	15	20	25
EPINAXL	LEPTH	26	-	-	-	-	-
MFORX	SMISC	-	1	7	13	19	25
MFORY	SMISC	-	2	8	14	20	26
MFORZ	SMISC	-	3	9	15	21	27
MMOMX	SMISC	-	4	10	16	22	28
MMOMY	SMISC	-	5	11	17	23	29
MMOMZ	SMISC	-	6	12	18	24	30
P1	SMISC	-	31	-	-	-	32
OFFST1	SMISC	-	33	-	-	-	34
P2	SMISC	-	35	-	-	-	36
OFFST2	SMISC	-	37	-	-	-	38
P3	SMISC	-	39	-	-	-	40
OFFST3	SMISC	-	41	-	-	-	42
P4	SMISC	-	43	-	-	-	-
P5	SMISC	-	-	-	-	-	44

		Pseudo Node							
		1	2	3	4	5	6	7	8
TEMP	LBFE	1	2	3	4	5	6	7	8

5.2.3.3. Notes About Defining Element Tables

- The **ETABLE** command works only on the *selected* elements. That is, only data for the elements you have selected are moved to the element table. By changing the selected elements between **ETABLE** commands, you can selectively fill rows of the element table.

- The same Sequence Number combination may mean different data for different element types. For example, the combination SMISC,1 means MFOR(X) for BEAM4 (member force in the element X direction), P1 for SOLID45 (pressure on face 1), and MECHPOWER for TRANS126 (mechanical power). *Therefore, if your model has a combination of element types, be sure to select elements of one type (using* **ESEL** *or* **Utility Menu> Select> Entities***) before using the* **ETABLE** *command.*

- The ANSYS program *does not* automatically refill (update) the element table when you read in a different set of results (such as for a different load step) or when you alter the results in the database (such as by a load case combination). For example, suppose your model consists of our sample elements, and you issue the following commands in POST1:

```
SET,1              ! Read in results for load step 1
ETABLE,ABC,lS,6    ! Move SDIR at end J (KEYOPT(9)=0) to the element table
                   !  under heading "ABC"
SET,2              ! Read in results for load step 2
```

At this point, the "ABC" column in the element table *still contains data for load step 1*. To refill (update) the column with load step 2 values, you should issue the command **ETABLE**,*REFL*, or specify the refill option via the GUI.

- You can use the element table as a "worksheet" to do calculations among results data. This feature is described in Section 5.5: Additional POST1 Postprocessing.

- To save the element table, issue **SAVE**,*Fname*,*Ext* in POST1 or issue **/EXIT**,ALL when exiting the ANSYS program. (If you are using the GUI, follow the prompts in the dialog boxes that appear when you choose **Utility Menu> File> Save as** or **Utility Menu> File> Exit**.) This saves the table along with the rest of the database onto the database file.

- To erase the entire element table from memory, issue **ETABLE**,ERASE (**Main Menu> General Postproc> Element Table> Erase Table**). (Or issue **ETABLE**,*Lab*,*ERASE* to erase just the *Lab* column of the element table). The element table will automatically be erased from memory if you issue a **RESET** command (**Main Menu> General Postproc> Reset**).

5.2.4. Special Considerations for Principal Stresses

Principal stresses for SHELL61 elements are not readily available for review in POST1. By default, the principal stresses are available for all line elements except in either of the following cases:

- You have requested an interpolated time point or angle specification on the **SET** command.

- You have performed load case operations.

In the above cases (including *all* cases for SHELL61), you *must* choose **Main Menu> General Postproc> Load Case> Line Elem Stress** or issue the command **LCOPER**,LPRIN in order to calculate the principal stresses. *Then* you may access this data through **ETABLE**, or any appropriate printing or plotting command.

5.2.5. Reading in FLOTRAN Results

To read results data from the FLOTRAN "residual" file into the database, issue the **FLREAD** command (**Main Menu> General Postproc> Read Results> FLOTRAN 2.1A**). FLOTRAN results (**Jobname.RFL**) are read with the normal postprocessing functions or commands, such as **SET** (**Utility Menu> List> Results> Load Step Summary**).

5.2.6. Resetting the Database

The **RESET** command (**Main Menu> General Postproc> Reset**), allows you to re-initialize the POST1 command defaults portion of the database without leaving POST1. The command has the same effect as leaving and re-entering the ANSYS program.

5.3. Reviewing Results in POST1

Once the desired results data are stored in the database, you can review them through graphics displays and tabular listings. In addition, you can map the results data onto a path (for details, see Section 5.3.5: Mapping Results onto a Path).

5.3.1. Displaying Results Graphically

Graphics displays are perhaps the most effective way to review results. You can display the following types of graphics in POST1:

- Contour displays
- Deformed shape displays
- Vector displays
- Path plots
- Reaction force displays
- Particle flow traces.

5.3.1.1. Contour Displays

Contour displays show how a result item (such as stress, temperature, magnetic flux density, etc.) varies over the model. Four commands are available for contour displays:

Command(s):
   PLNSOL

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

Command(s):
   PLESOL

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

Command(s):
   PLETAB

GUI:
   Main Menu> General Postproc> Plot Results> Contour Plot> Elem Table

Command(s):
   PLLS

GUI:
   Main Menu> General Postproc> Plot Results> Line Elem Res

The **PLNSOL** command produces contour lines that are continuous across the entire model. Use either for primary as well as derived solution data. Derived solution data, which are typically discontinuous from element to element, are averaged at the nodes so that continuous contour lines can be displayed.

Sample contour displays of primary data (TEMP) and derived data (TGX) are shown below.

```
PLNSOL,TEMP              ! Primary data: degree of freedom TEMP
```

Figure 5.1 Contouring Primary Data with PLNSOL
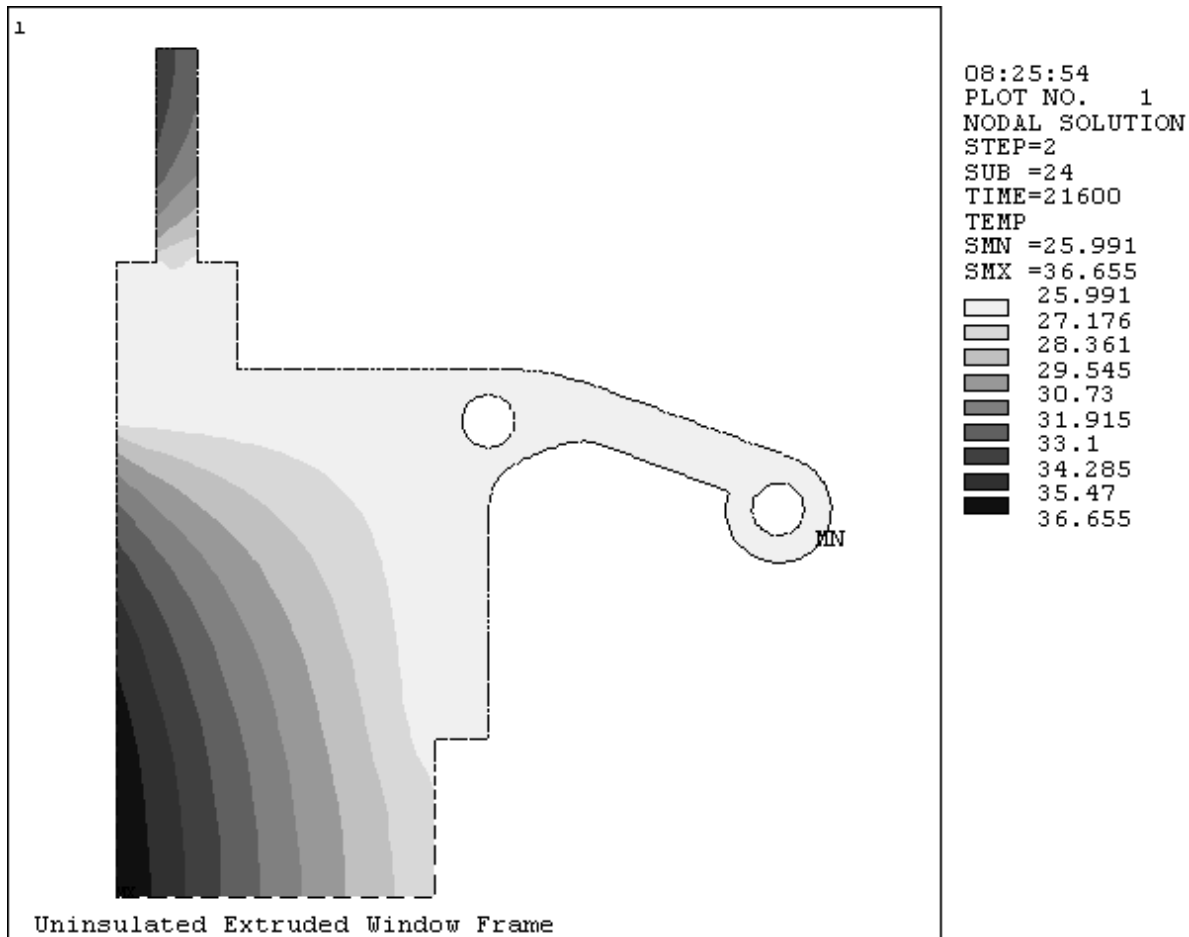
If PowerGraphics is enabled, you can control averaging of derived data with the following:

> **Command(s): AVRES**
> **GUI: Main Menu> General Postproc> Options for Outp**
> **Utility Menu> List> Results> Options**

Any of the above allows you to specify whether or not results will be averaged at element boundaries where material and/or real constant discontinuities exist. For more information, see Chapter 10, "PowerGraphics".

> **Caution:** If PowerGraphics is disabled, you cannot use the **AVRES** command to control averaging, and the averaging operation is performed at all nodes of the selected elements without regard to the attributes of the elements connected to them. This can be inappropriate in areas of material or geometric discontinuities. When contouring derived data (which are averaged at the nodes), be sure to select elements of the same material, same thickness (for shells), same element coordinate system orientation, etc.

```
PLNSOL,TG,X              ! Derived data: thermal gradient TGX
```

See the **PLNSOL** command description for further information.

Figure 5.2 Contouring Derived Data with PLNSOL
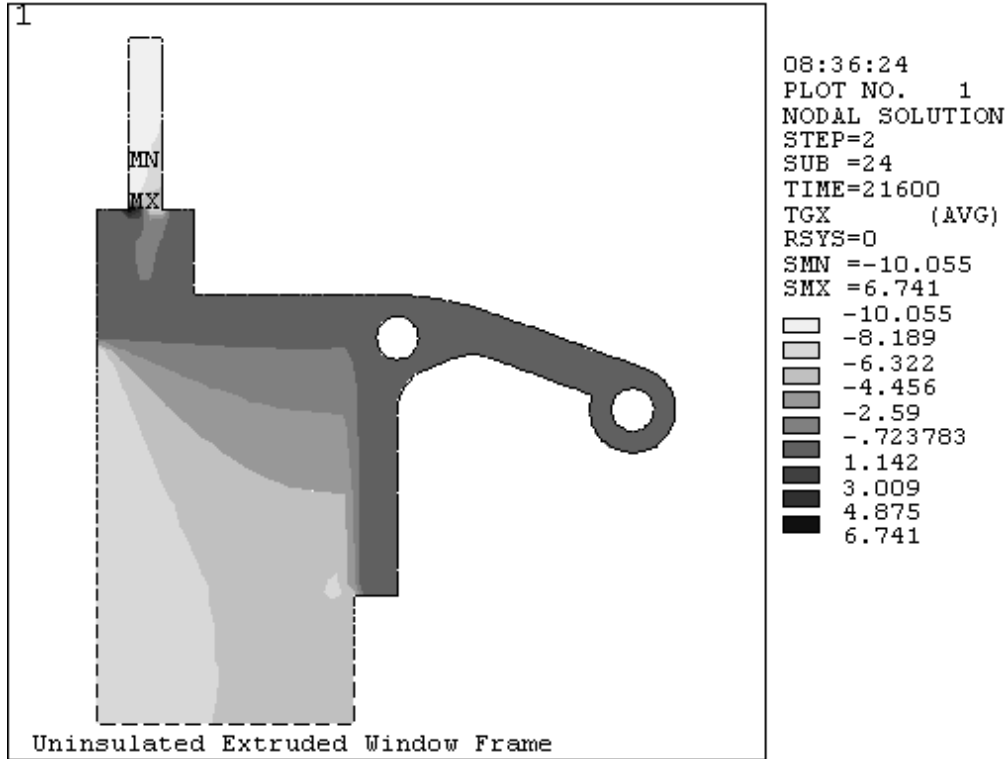
The **PLESOL** command produce contour lines that are discontinuous across element boundaries. Use this type of display mainly for derived solution data. For example:

```
PLESOL,TG,X
```

Figure 5.3 A Sample PLESOL Plot Showing Discontinuous Contours
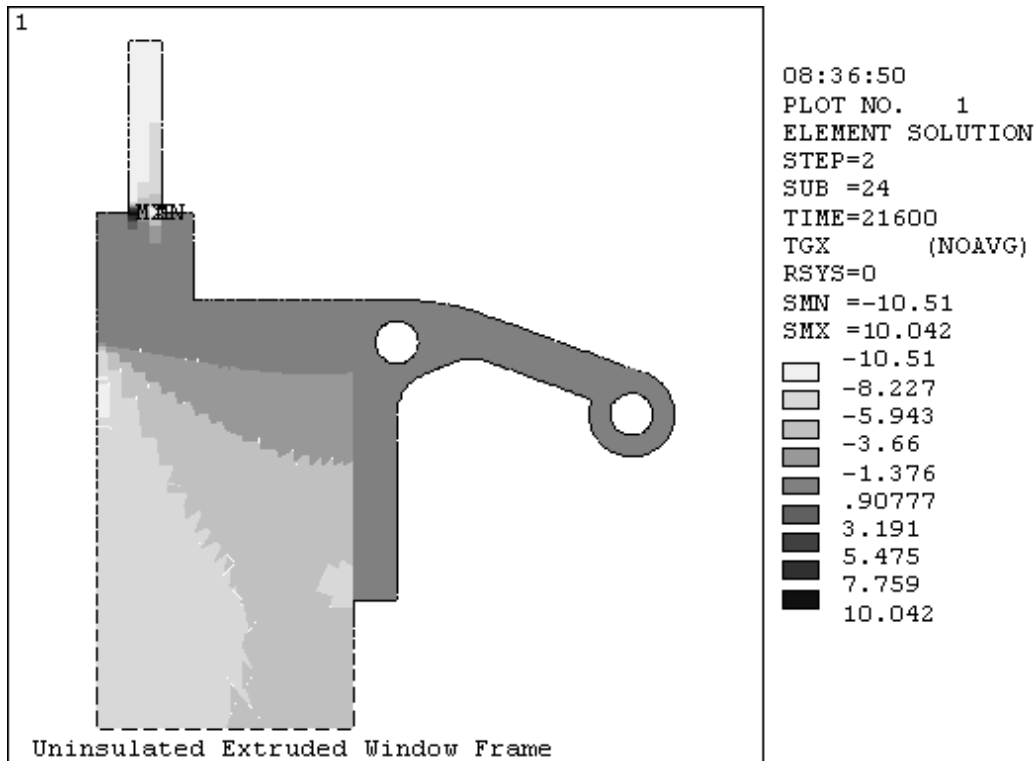
The **PLETAB** command contours data stored in the element table. The *Avglab* field on the **PLETAB** command gives you the option of averaging the data at nodes (for continuous contours) or not averaging (the default, for having discontinuous contours). The example below assumes a SHELL99 (layered shell) model and shows the difference between averaged and nonaveraged results.

```
ETABLE,SHEARXZ,SMISC,9      ! Interlaminar shear (ILSXZ) at bottom of layer 2
PLETAB,SHEARXZ,AVG          ! Averaged contour plot of SHEARXZ
```

Figure 5.4 Averaged PLETAB Contours
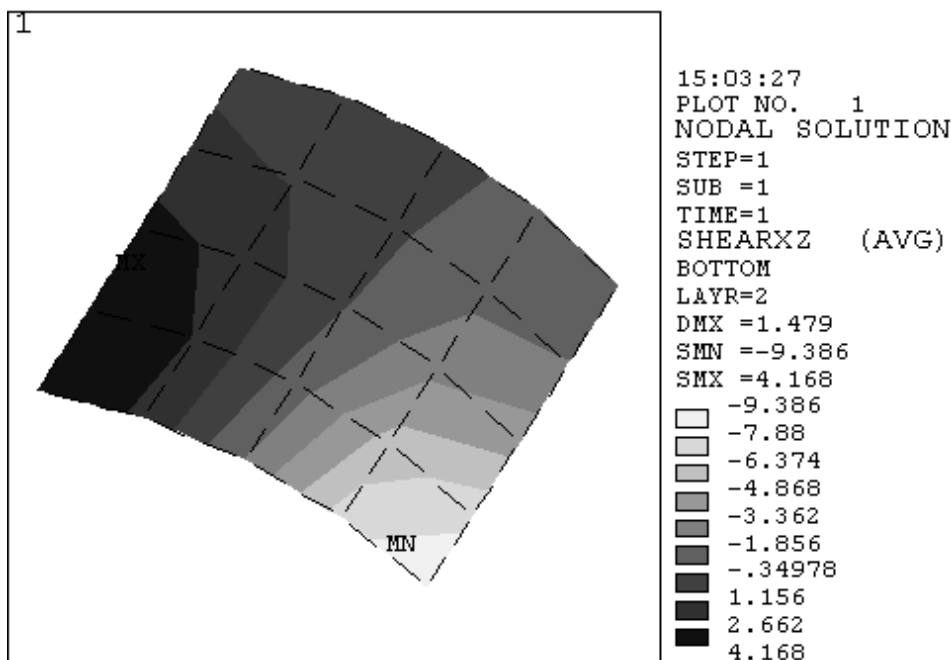
```
PLETAB,SHEARXZ,NOAVG        ! Unaveraged (default) contour plot of SHEARXZ
```

Figure 5.5 Unaveraged PLETAB Contours
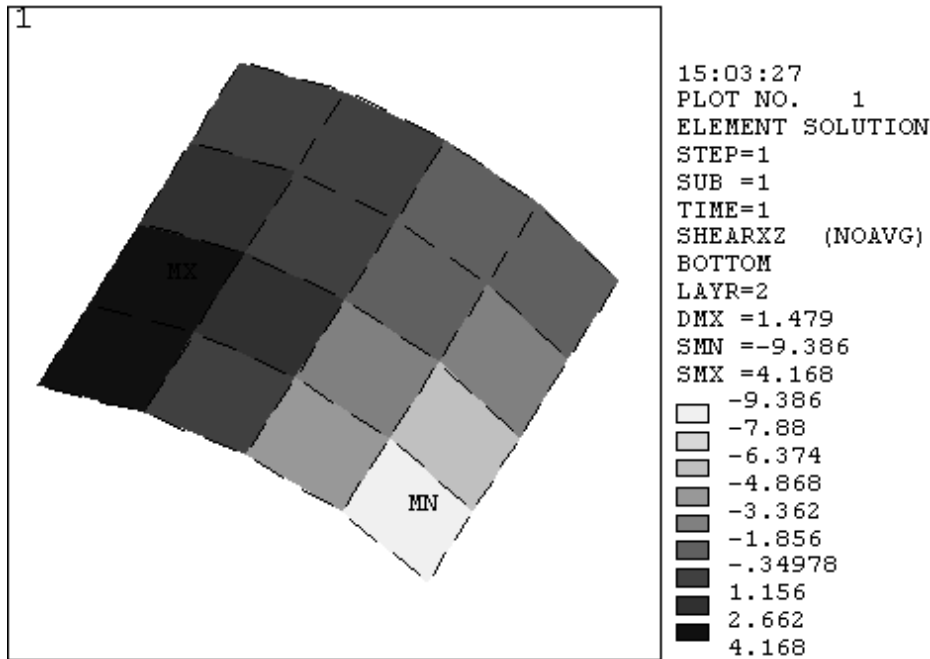
The **PLLS** command displays line element results in the form of contours. This command also requires data to be stored in the element table. This type of display is commonly used for shear and moment diagrams in beam analyses. The example below assumes a BEAM3 (2-D beam) model with KEYOPT(9) = 1:

```
ETABLE,IMOMENT,SMISC,6     ! Bending moment (MMOMZ) at end I, named IMOMENT
ETABLE,JMOMENT,SMISC,18    ! MMOMZ at end J, named JMOMENT
PLLS,IMOMENT,JMOMENT       ! Display results for IMOMENT, JMOMENT
```

Figure 5.6 Moment Diagram Using PLLS
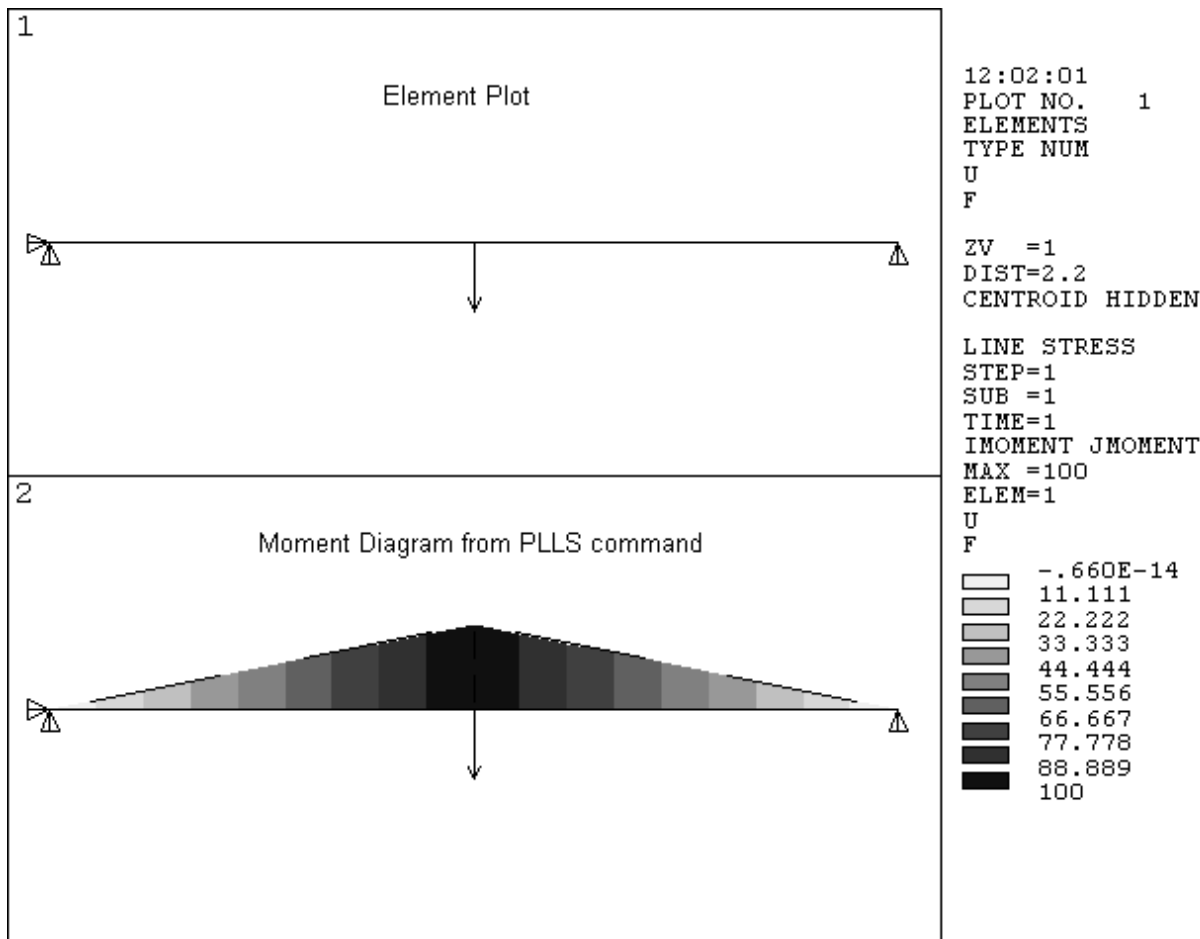
PLLS simply draws a straight line between values at the I and J nodes of an element without any regard to how the result item varies along the element length. You can use a negative scaling factor to invert the plot.

Notes

- You can produce *isosurface* contour displays by first setting Key on the **/CTYPE** command (**Utility Menu> PlotCtrls> Style> Contours> Contour Style**) to 1. See Chapter 6, "The Time-History Postprocessor (POST26)" for more information about isosurfaces.

- *Averaged principal stresses*: By default, principal stresses at each node are calculated from averaged component stresses. You can reverse this, so that principal stresses are first calculated per element, then averaged at the nodes. To do so, use the following:
  Command(s): AVPRIN
  GUI: Main Menu> General Postproc> Options for Outp
  Utility Menu> List> Results> Options

This method is not normally used, but can be useful in special circumstances. Averaging operations should *not* be done at nodal interfaces of differing materials.

- *Vector sum data*: These follow the same practice as the principal stresses. By default, the vector sum magnitude (square root of the sum of the squares) at each node is calculated from averaged components. By using the **AVPRIN** command, you can reverse this, so that the vector sum magnitudes are first calculated per element, then averaged at the nodes.

- *Shell elements or layered shell elements*: By default, results for shell or layered elements are assumed to be at the top surface of the shell or layer. To display results at the top, middle or bottom surface, use the **SHELL** command (**Main Menu> General Postproc> Options for Outp**). For layered elements, use the **LAYER** command (**Main Menu> General Postproc> Options for Outp**) to indicate layer number.

- *von Mises equivalent strains (EQV):* The effective Poisson's ratio used in computing these quantities may be changed using the **AVPRIN** command.

> **Command(s): AVPRIN**
> **GUI: Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu**
> **Main Menu> General Postproc> Plot Results> Contour Plot> Element Solu**
> **Utility Menu> Plot> Results> Contour Plot> Elem Solution**

  Typically, you would set the effective Poisson's ratio to the input Poisson's ratio for elastic equivalent strain (item and component EPEL,EQV) and to 0.5 for inelastic strains (item and component EPPL,EQV or EP-CR,EQV). For total strains (item and component EPTOT,EQV), you would typically use an effective Poisson's ratio between the input Poisson's ratio and 0.5. As an alternative, you can save the equivalent elastic strains using **ETABLE** with the effective Poisson's ratio equal to the input Poisson's ratio and save the equivalent plastic strains in another table using 0.5 as the effective Poisson's ratio, then combine the two table entries using **SADD** to obtain the total equivalent strain.

- *Effect of /EFACET*: You may see different plots with different **/EFACET** settings when viewing continuous contour plots (**PLNSOL**). If you set **/EFACET**,1, the contour values for the intermediate locations are interpolated based on the average of the adjacent *averaged* corner node values. However, if you set **/EFACET**,2, the midside node values are first calculated within each element, based on the average of the adjacent *unaveraged* corner node values. The midside node values are then averaged together for a **PLNSOL** contour plot. If you issue **/EFACET**,4, ANSYS uses shape functions (except for higher order p-elements) to calculate results values at three subgrid points along each element edge. The subgrid values are first calculated within each element and are then averaged together for **PLNSOL** plots. Therefore, the contour values at the midside locations will differ with different **/EFACET** settings.

  In most cases, **PLESOL** contours will be the same regardless of **/EFACET** settings. However, you will see differences in **PLESOL** contour plots if you change **/EFACET** settings in conjunction with any **RSYS** setting other than $KCN = 0$. When a coordinate system other than global Cartesian is chosen ($KCN = 1, 2$, etc.), the results are first averaged in the global Cartesian coordinate system, and then the averaged results are transformed to the specified results coordinate system.

5.3.1.2. Deformed Shape Displays

You can use these in a structural analysis to see how the structure has deformed under the applied loads. To generate a deformed shape display, use one of the following:

> **Command(s): PLDISP**
> **GUI: Utility Menu> Plot> Results> Deformed Shape**
> **Main Menu> General Postproc> Plot Results> Deformed Shape**

For example, you might issue the following **PLDISP** command:

```
PLDISP,1                 ! Deformed shape superimposed over undeformed shape
```

Figure 5.7 A Sample PLDISP Plot


```
1                                    14:45:21
                                     PLOT NO.    1
                                     DISPLACEMENT
                                     STEP=3
                                     SUB =43
                                     TIME=6
                                     RSYS=0
                                     DMX =1.122

                                    *DSCA=1
                                     ZV  =1
                                     DIST=5.5
                                    *XF  =52.765
                                    *YF  =30
                                     CENTROID HIDDEN




HYPERELASTIC O-RING
```
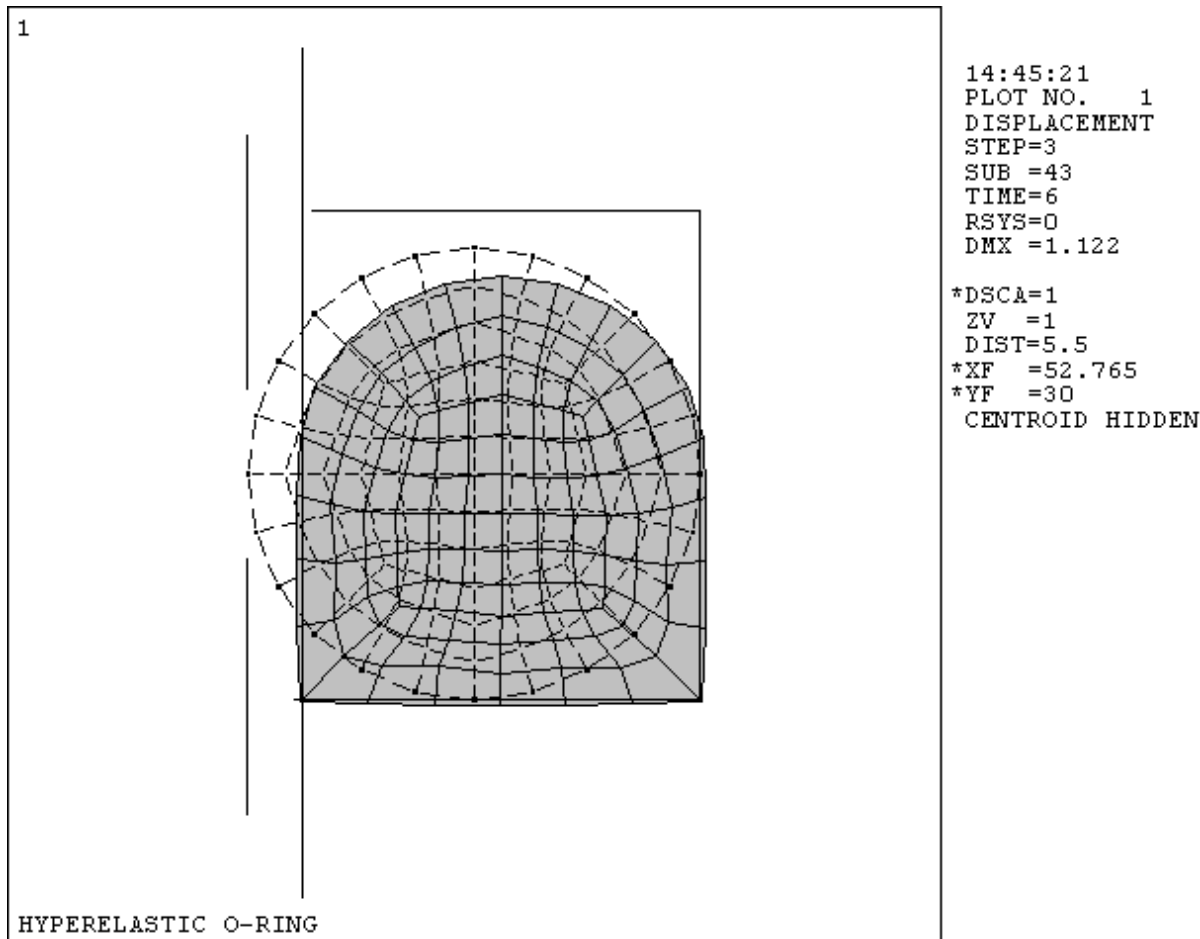
You can change the displacement scaling by issuing the **/DSCALE** command (**Utility Menu> PlotCtrls> Style> Displacement Scaling**).

Be aware that when you enter POST1, all load symbols are automatically turned off. These load symbols remain off if you subsequently re-enter the PREP7 or SOLUTION processors. If you turn the load symbols on in POST1, the resulting display will show the loads on the deformed shape.

5.3.1.3. Vector Displays

Vector displays use arrows to show the variation of both the magnitude and direction of a *vector* quantity in the model. Examples of vector quantities are displacement (U), rotation (ROT), magnetic vector potential (A), magnetic flux density (B), thermal flux (TF), thermal gradient (TG), fluid velocity (V), principal stresses (S), etc.

To produce a vector display, use one of the following:
>**Command(s): PLVECT**
>**GUI: Main Menu> General Postproc> Plot Results> Vector Plot> Predefined**
>**Main Menu> General Postproc> Plot Results> Vector Plot> User-Defined**

To scale the arrow lengths, use one of the following:
>**Command(s): /VSCALE**
>**GUI: Utility Menu> PlotCtrls> Style> Vector Arrow Scaling**

```
    PLVECT,B                ! Vector display of magnetic flux density
```

Figure 5.8 PLVECT Vector Plot of Magnetic Field Intensity
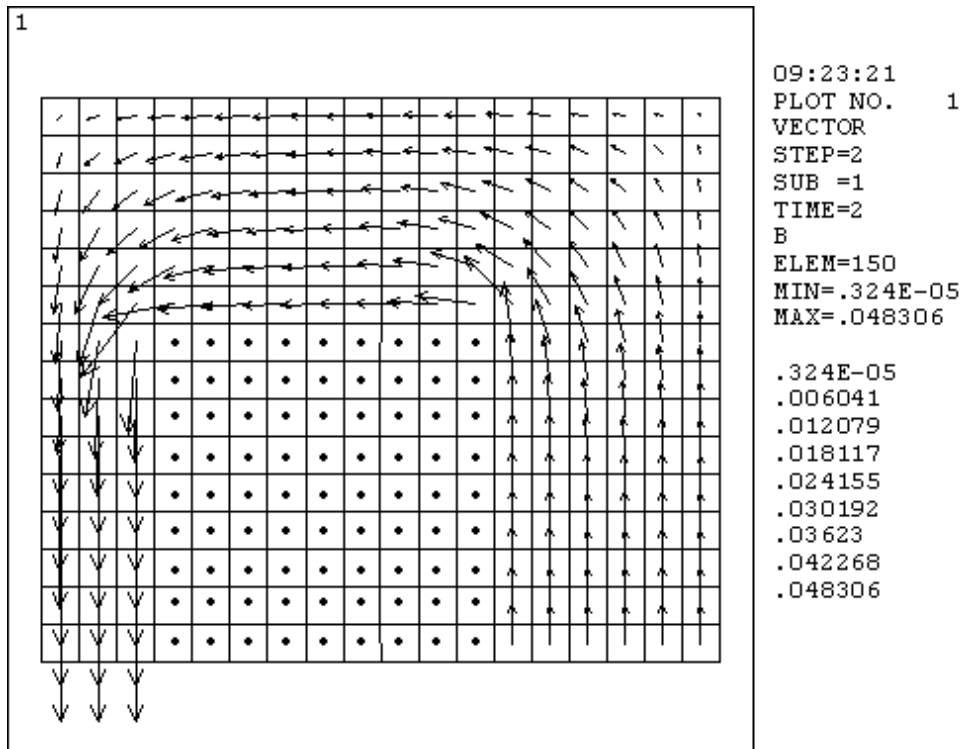
You can also create your own vector quantity by specifying two or three components on the **PLVECT** command.

5.3.1.4. Path Plots

These are graphs that show the variation of a quantity along a predefined path through the model. To produce a path plot, you need to perform these tasks:

1. Define path attributes using the **PATH** command (**Main Menu> General Postproc> Path Operations> Define Path> Path Status> Defined Paths**).

2. Define the points of the path using the **PPATH** command (**Main Menu> General Postproc> Path Operation> Define Path> Modify Path**).

3. Map the desired quantity on to the path using the **PDEF** command (**Main Menu> General Postproc> Path Operations> Map onto Path**)

4. Use the **PLPATH** and **PLPAGM** commands (**Main Menu> General Postproc> Path Operations> Plot Path Items**) to display the results.

More details on this appear later in Section 5.3.5: Mapping Results onto a Path.

5.3.1.5. Reaction Force Displays

These are similar to boundary condition displays and are activated using the labels RFOR or RMOM on the **/PBC** command. Any subsequent display (produced by commands such as **NPLOT**, **EPLOT**, or **PLDISP**) will include reaction force symbols at points where DOF constraints were specified. The sum of nodal forces for a DOF belonging to a constraint equation does not include the force passing through that equation. See the *ANSYS, Inc. Theory Reference*.

Like reactions, you can also display *nodal* forces using labels NFOR or NMOM on the **/PBC** command (**Utility Menu> PlotCtrls> Symbols**). These are forces exerted by an element on its node. The sum of these forces at each node is usually zero except at constrained nodes or at nodes where loads were applied.

By default, the force (or moment) values that are printed and plotted represent the *total* forces (sum of the static, damping, and inertial components). The **FORCE** command (**Main Menu> General Postproc> Options for Outp**) allows you to separate the total force into individual components.

5.3.1.6. Particle Flow and Charged Particle Traces

A particle flow trace is a special form of graphics display that shows how a particle travels in a flowing fluid. A charged particle trace is a graphics display that shows how a charged particle travels in an electric or magnetic field. See Chapter 12, "Creating Geometric Results Displays" for more information on graphic displays and see Chapter 15, "Animation" for information on particle trace animation. See the *ANSYS, Inc. Theory Reference* for simplifying assumptions on electromagnetic particle tracing.

A particle flow or charged particle trace requires two functions:

1. The **TRPOIN** command (**Main Menu> General Postproc> Plot Results> Flow Trace> Defi Trace Pt**). Either defines a point on the path trajectory (starting point, ending point, or anywhere in between).

2. The **PLTRAC** command (**Main Menu> General Postproc> Plot Results> Flow Trace> Plot Flow Tra**). Either produces the flow trace on an element display. Up to 50 points can be defined and plotted simultaneously.

A sample **PLTRAC** plot is shown below.

Figure 5.9 A Sample Particle Flow Trace
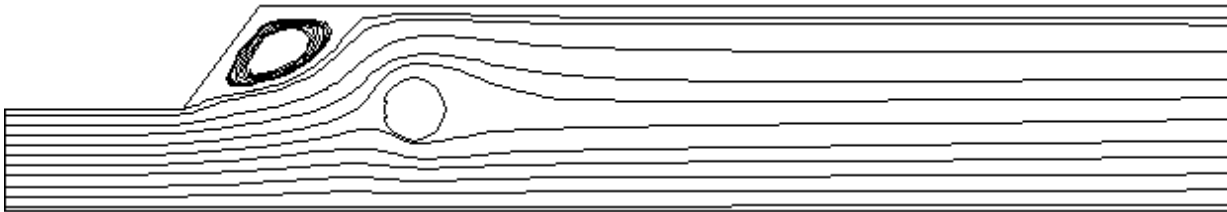
The *Item* and *Comp* fields on **PLTRAC** allow you to see the variation of a specified item (such as velocity, pressure, and temperature for a particle flow trace or electric potential for a charged particle trace). The variation of the item is displayed along the path trajectory as a color-contoured ribbon.

Figure 5.10 A Sample Charge Particle Trace in Electric and/or Magnetic Fields

Tracing a particle moving in a pure magnetic field might look like this:
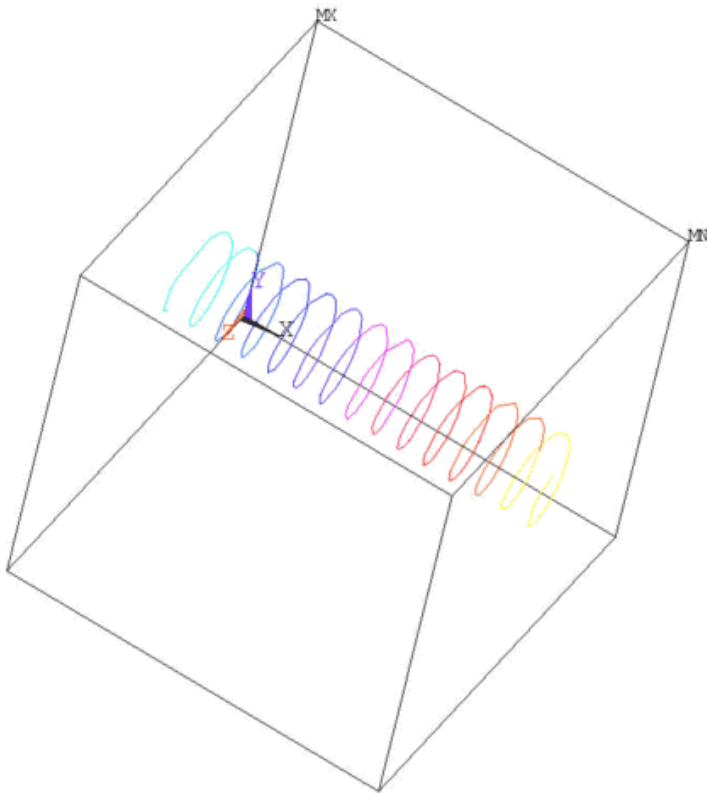
The path of that particle moving through a pure electric field might look like this:
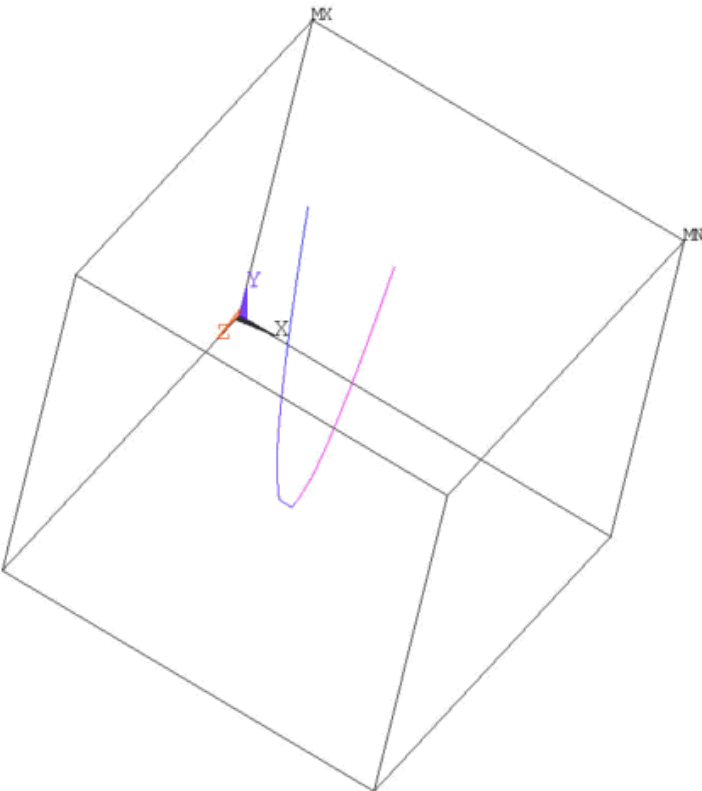
Plotting that same particle in the presence of both the electric and magnetic fields (with E normal to B) would then look like this:
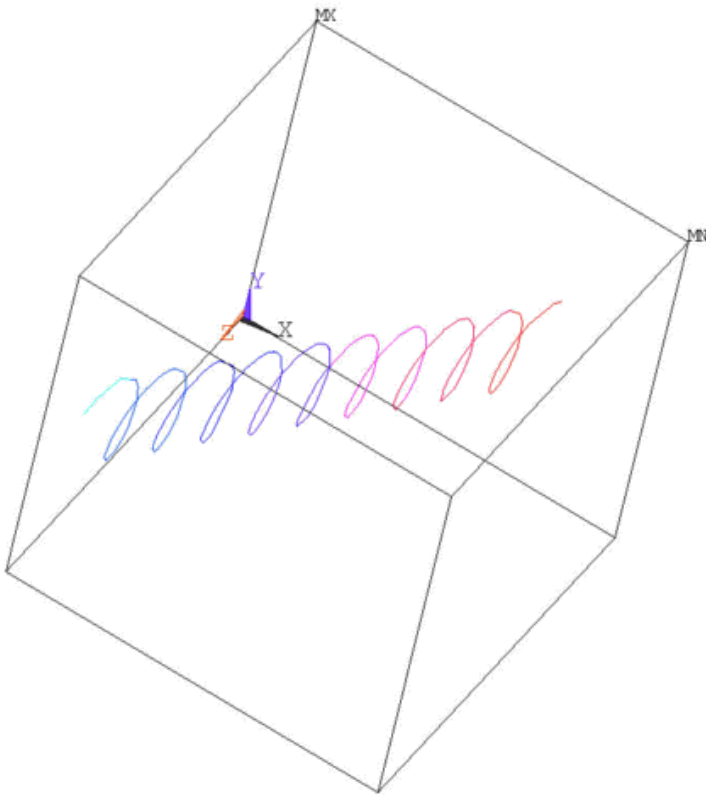
Other commands are:

- **TRPLIS** command (**Main Menu> General Postproc> Plot Results> Flow Trace> List Trace Pt**) - lists trace points.

- **TRPDEL** command (**Main Menu> General Postproc> Plot Results> Flow Trace> Dele Trace Pt**) - deletes trace points.

- **TRTIME** command (**Main Menu> General Postproc> Plot Results> Flow Trace> Time Interval**) - defines the flow trace time interval.

- **ANFLOW** command (**Utility Menu> PlotCtrls> Animate> Particle Flow**) - generates an animated sequence of particle flow.

Notes

- Three array parameters are created at the time of the particle trace: TRACPOIN, TRACDATA and TRACLABL. These array parameters can be used to put the particle velocity and the elapsed time into path form. The procedure to put the arrays into a path named PATHNAME is as follows:

```
*get,npts,PARM,TRACPOIN,DIM,x
PATH,PATHNAME,npts,9,1
PAPUT,TRACPOIN,POINTS
PAPUT,TRACDATA,TABLES
PAPUT,TRACLABL,LABELS
PRPATH,S,T_TRACE,VX_TRACE,VY_TRACE,VZ_TRACE,VS_TRACE
```

- Particle flow traces occasionally stop for no apparent reason. This can occur in stagnant flow regions, near wall flow regions, or when a particle is tracking along an element edge. To resolve the problem, adjust the initial particle point slightly in the cross stream direction.

- For charged particle traces, the variables Chrg and Mass input by the **TRPOIN** command (**Main Menu> General Postproc> Plot Results> Flow Trace> Defi Trace Pt**) have units of Coulombs and kilograms, respectively, in the MKS system.

- The particle tracing algorithm could lead to an infinite loop. For example, a charged particle trace could lead to an infinite circular loop. To avoid infinite loops, the **PLTRAC** command argument *MXLOOP* sets a limiting value.

- Charge particle tracing could be performed after an electrostatic analysis (using only electric field), or after a magnetostatic analysis using only magnetic field or coupled magnetic and electric fields. The latter case could be done using the electric field as a body load applied either with **BFE**,EF command or with **LDREAD**,EF command.

5.3.1.7. Cracking and Crushing Plots

If you have SOLID65 elements in your model, you can use the **PLCRACK** command (**Main Menu> General Postproc> Plot Results> Crack/Crush**) to determine which elements have cracked and/or crushed. Small circles will be shown where the concrete has cracked, and small octagons will be shown where the concrete has crushed (see Figure 5.11: "Concrete Beam with Cracks"). The cracking and crushing symbols are visible when a non-hidden, vector type of display is used. To specify such a device, issue the command **/DEVICE**,VECTOR,ON (**Utility Menu> PlotCtrls> Device Options**).

Figure 5.11 Concrete Beam with Cracks

REINFORCED CONCRETE BEAM SHOWING FAILURE PLANES
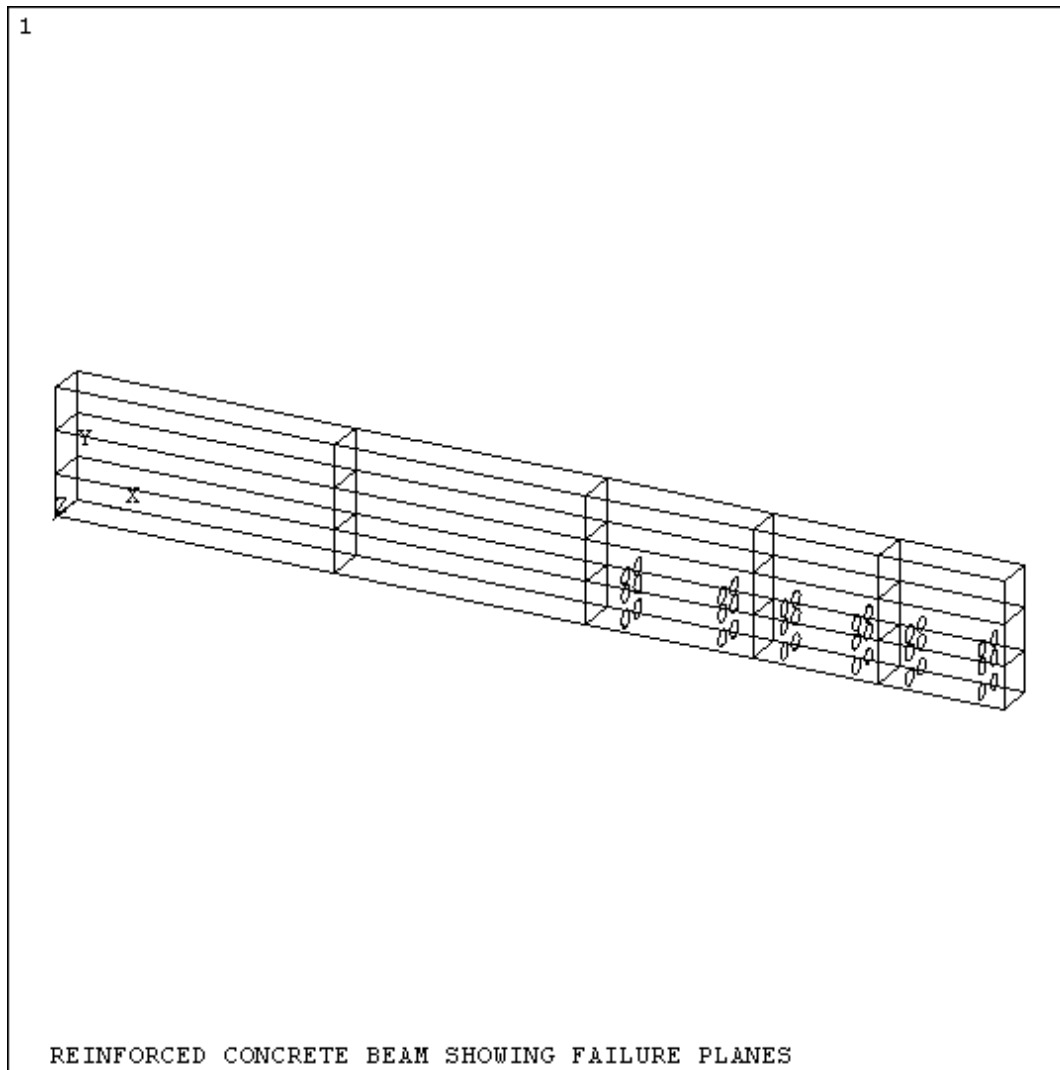
5.3.2. Surface Operations

You can map any nodal results data onto a user defined surface in POST1. You can then perform mathematical operations on these surface results to calculate meaningful quantities, including total force or average stress across a cross section, net charge inside a closed volume, fluid mass flow rate, heat flow across a cross section, and more. You can also plot contours of the mapped results.

Surface operations are available both interactively (from the GUI), and via batch (command line operations). Each of the commands is referenced below; each process is found in the **Main Menu> General Postproc> Surface Operations** area of the GUI. A full complement of surface commands are provided to perform surface operations.

Table 5.4 Surface Operations

These POST1 commands are used to define an arbitrary surface and to develop results information for that surface.

SUCALC	Create new result data by operating on two existing result datasets on a given surface.
SUCR	Create a surface.
SUDEL	Delete geometry information as well as any mapped results for specified surface or for all selected surfaces.

These POST1 commands are used to define an arbitrary surface and to develop results information for that surface.

SUEVAL	Perform operations on a mapped item and store result in a scalar parameter.
SUGET	Move surface geometry and mapped results to an array parameter.
SUMAP	Map results onto selected surface(s).
SUPL	Plot specified result data on all selected surfaces or on the specified surface.
SUPR	Print surface information.
SURESU	Resume surface definitions from a specified file.
SUSAVE	Save surface definitions and result items to a file.
SUSEL	Select a subset of surfaces
SUVECT	Perform Operations between two mapped result vectors.

> *Note* — You can define surfaces only in models containing 3-D solid elements. Shells, beams and 2-D element types are not supported. Surface creation will operate on selected, valid 3-D solid elements only and ignore other element types if they are present in your model.

The basic steps for surface operations are as follows:

- Define the surfaces using the **SUCR** command.

- Map results data on the selected surfaces using the **SUSEL** and **SUMAP** commands.

- Operate on the results using the **SUEVAL**, **SUCALC** and **SUVECT** commands.

Once your data is mapped on the surface, you can review the results using the graphical display and tabular listing capabilities found in the **SUPL** and **SUPR** commands.

Additional capabilities include archiving the surface data you create to a file or an array parameters and recalling stored surface data. The following topics relate primarily to surface definition and usage.

5.3.2.1. Defining the Surface

You define your surface using the **SUCR** command. This command creates your named surface (containing no more than eight characters), according to a specified category (plane or sphere), at a defined refinement level.

The surfaces you create fall into two categories:

- A cross section you create based on the current working plane

- A closed surface represented by a sphere at the current working plane origin, with a user-specified radius.

For $SurfType$ = CPLANE, $nRefine$ refers to the number of points that define the surface. If $SurfType$ = CPLANE, and $nRefine$ = 0, the points reside at the section cuts through the element by the cutting plane. Increasing $nRefine$ to 1 will subdivide each surface facet into 4 subfacets, thus increasing the number of points at which the results can be interpolated. $nRefine$ can vary between 0 and 3. Increasing $nRefine$ can have significant impact on memory and speed of surface operations.

/EFACET operations will add to this refinement, and values greater than 1 can amplify the effect of $nRefine$. An **/EFACET** setting greater than 1 divides the elements into subelements, and $nRefine$ then refines the facets of the subelements.

For $SurfType$ = SPHERE, $nRefine$ is the number of divisions along a 90° arc of the sphere (default is 90, Min = 10, Max = 90).

Each time you create a surface, the following predefined geometric items are computed and stored.

- • GCX, GCY, GCZ - global Cartesian coordinates at each point on the surface.
- • NORMX, NORMY, NORMZ - components of the unit normal at each point on the surface.
- • DA - contributory area of each point.

These items are used to perform mathematical operations with surface data (for instance, DA is required to calculate surface integrals). Once you create a surface, these quantities (using the predefined labels) are available for all subsequent math operations.

Issue **SUPL**,*SurfName* to display your defined surface. A maximum of 100 surfaces can exist within one model, and all operations (mapping results, math operations, etc.) will be carried out on all selected surfaces. You can use the **SUSEL** command to change the selected surface set.

See the **SUCR** command for more information of creating surfaces.

5.3.2.2. Mapping Results Data Onto a Surface

Once you define a surface, you use the **SUMAP** command to map your data onto that surface. Nodal results data in the active results coordinate system is interpolated onto the surface and operated on as a *result set*. Your result sets can be made up of primary data (nodal DOF solution), derived data (stress, flux, gradients, etc.), FLOTRAN nodal results, and other results values.

You define your mapped data in the **SUMAP** command by supplying a name for the result set, and then specifying the type of data and the directional properties.

You can make the results coordinate system match the active coordinate system (used to define the path) by issuing the following pair of commands:

```
*GET,ACTSYS,ACTIVE,,CSYS
RSYS,ACTSYS
```

The first command creates a user-defined parameter (ACTSYS) that holds the value defining the currently active coordinate system. The second command sets the results coordinate system to the coordinate system specified by ACTSYS.

Results mapped on to a surface do not account for discontinuities (e.g., material discontinuities) but are based on the currently selected set of elements. Selecting the proper set of elements is critical to valid surface operations, and improper selection will either result in failed mapping, or produce invalid results.

To clear result sets from the selected surfaces (except GCX, GCY, GCZ, NORMX, NORMY, NORMZ, DA), issue **SUMAP**,*RSetname*,CLEAR. To form additional labeled result sets by operating on existing surface result sets, use the **SUEVAL**, **SUVECT** or **SUCALC** commands.

5.3.2.3. Reviewing Surface Results

You can use the **SUPL** command to visually display your surface results, or use the **SUPR** command to get a tabular listing.

SUPL of a single result set item is displayed as a contour plot on the selected surfaces. You can also obtain a vector plot (such as for fluid velocity vector) by using a special result set naming convention. If *SetName* is a "vector prefix" (i.e., if SetNameX, SetNameY, and SetNameZ exist), ANSYS will plot these vectors on the surface as arrows.

Example for vector plot:

```
    SUCREATE,SURFACE1,CPLANE     ! create a surface called "SURFACE1"
SUMAP,VELX,V,X          ! map x,y,z velocities with VEL as prefix
SUMAP,VELY,V,Y
SUMAP,VELZ,V,Z
SUPLOT,SURFACE1,VEL      ! this will result in a vector plot of velocities
```

Display of facet outlines on the surface plots is controlled by **/EDGE** command similar to other postprocessing plots.

5.3.2.4. Performing Operations on Mapped Surface Result Sets

Three commands are available for mathematical operations among surface result sets:

- The **SUCALC** command lets you add, multiply, divide, exponentiate and perform trigonometric operations on all selected surfaces.

- The **SUVECT** command calculates the cross or dot product of two result vectors on all selected surfaces.

- The **SUEVAL** command calculates surface integral, area weighted average, or sum of a result set on all selected surfaces. The result of this operation is a APDL scalar parameter.

5.3.2.5. Archiving and Retrieving Surface Data to a File

You can store your surface data in a file, so that when you leave POST1, it can be retrieved later. You use the **SUSAVE** command to archive your data. Once you have saved the information for your surface, you use the **SURESU** command to retrieve it.

You can opt to archive all defined surfaces, all selected surfaces or only a specified surface. When you retrieve surface data, it becomes the currently active surface data. Any existing surface data is cleared.

The following input listings provides examples of archiving and retrieving operations.

```
/post1
! define spherical surface at WP origin, with a radius of 0.75 and 10 divisions per 90 degree arc
sucreate,surf1,sphere,0.75,10
wpoff,,,-2                              ! offset working plane
! define a plane surface based on the intersection of working plane
! with the currently selected elements
sucreate,surf2,cplane

susel,s,surf1          ! select surface 'surf1'
sumap,psurf1,pres   ! map pressure on surf1. Result set name "psurf1"
susel,all             ! select all surfaces
sumap,velx,v,x      ! map VX on both surfaces. Result set name "velx"
sumap,vely,v,y      ! map VY on both surfaces. Result set name "vely"
sumap,velz,v,z      ! map VZ on both surfaces. Result set name "velz"

supr                ! global status of current surface data
supl,surf1,sxsurf1  !  contour plot result set sxsurf1
supl,all,velx,1       ! contour plot result set velx on all surfaces. Plot in context of all elements in
model
supl,surf2,vel     ! vector plot of resultant velocity vector on surface "surf2"

suvect, vdotn,vel,dot,normal   ! dot product of velocity vector and surface normal
                                        ! result is stored in result set "vdotn"
sueval, flowrate, INTG, vdotn      ! integrate "vdotn" over area to get apdl parameter "flow rate"
susave,all,file,surf               ! Store defined surfaces in a file
finish
```

5.3.2.6. Archiving and Retrieving Surface Data to an Array Parameter

Writing surface data to an array allows you to perform APDL operations on your result sets. You use the **SUGET** command to write either the interpolated results data only (default), or the results data and the geometry data to a parameter that you define. The parameter is automatically dimensioned and filled with data.

5.3.2.7. Deleting a Surface

Use the **SUDEL** command to delete one or more surfaces, along with the mapped results on those surfaces. You can choose to delete all surfaces, or choose to delete individual surfaces by name. Use the **SUPR** command to review the current list of surface names.

5.3.3. Integrating Surface Results

The **INTSRF** command (**Main Menu> General Postproc> Nodal Calcs> Surface Integrl**) allows you to integrate nodal results on a selected surface. You must first select the nodes on the surface where the nodal results are to be integrated.

You may use **INTSRF** to calculate lift and drag. If the surface is a fluid-solid interface, select only the fluid elements for integration. Then, select the nodes by using the *EXT* option on the **NSEL** command (**Utility Menu> Select> Entities**).

To use **INTSRF** to calculate lift and drag, you must specify a results coordinate system with the X-axis and Y-axis aligned in the direction of the incoming flow field and the direction of gravity, respectively. Then, the drag force is the force in the X-direction and the lift is the force in the Y-direction. You use **INTSRF**,PRES and **INTSRF**,TAUW to obtain the lift and drag forces, respectively. You can use **INTSRF**,FLOW to obtain both the lift and drag forces, separately. The outcome is written to the output (**Jobname.OUT**).

Integration results are in the active coordinate system (see the **RSYS** command). The type of results coordinate system must match the type used in the analysis. However, you may translate and rotate forces and moments as needed. You use the ***GET** command (**Utility Menu> Parameters> Get Scalar Data**) to retrieve the results.

5.3.4. Listing Results in Tabular Form

An effective way of documenting analysis results (for reports, presentations, etc.) is to produce tabular listings in POST1. Listing options are available for nodal and element solution data, reaction data, element table data, and more.

Sample Listing of PRESOL,ELEM

```
     PRINT ELEM ELEMENT SOLUTION PER ELEMENT

     ***** POST1 ELEMENT SOLUTION LISTING *****

      LOAD STEP     1  SUBSTEP=     1
      TIME=    1.0000          LOAD CASE=  0

     EL=     1  NODES=     1     3  MAT=  1
     TEMP =     0.00     0.00     0.00     0.00
     LOCATION    SDIR        SBYT         SBYB
      1 (I)    0.00000E+00  130.00      -130.00
      2 (J)    0.00000E+00  104.00      -104.00
     LOCATION    SMAX         SMIN
      1 (I)     130.00       -130.00
      2 (J)     104.00       -104.00
     LOCATION   EPELDIR      EPELBYT      EPELBYB
      1 (I)     0.000000     0.000004    -0.000004
      2 (J)     0.000000     0.000003    -0.000003
```

```
LOCATION  EPTHDIR     EPTHBYT     EPTHBYB
 1 (I)    0.000000    0.000000    0.000000
 2 (J)    0.000000    0.000000    0.000000
EPINAXL =    0.000000

EL=    2  NODES=    3     4  MAT=  1
TEMP =    0.00    0.00    0.00    0.00
LOCATION   SDIR        SBYT        SBYB
 1 (I)   0.00000E+00  104.00     -104.00
 2 (J)   0.00000E+00  78.000     -78.000
LOCATION   SMAX        SMIN
 1 (I)    104.00     -104.00
 2 (J)    78.000     -78.000
LOCATION  EPELDIR     EPELBYT     EPELBYB
 1 (I)    0.000000    0.000003   -0.000003
 2 (J)    0.000000    0.000003   -0.000003
LOCATION  EPTHDIR     EPTHBYT     EPTHBYB
 1 (I)    0.000000    0.000000    0.000000
 2 (J)    0.000000    0.000000    0.000000
EPINAXL =    0.000000
```

5.3.4.1. Listing Nodal and Element Solution Data

To list specified nodal solution data (primary as well as derived), use either of the following:
> **Command(s): PRNSOL**
> **GUI: Main Menu> General Postproc> List Results> Nodal Solution**

To list specified results for selected elements, use one of these methods
> **Command(s): PRESOL**
> **GUI: Main Menu> General Postproc> List Results> Element Solution**

To obtain line element solution printout, specify the ELEM option with **PRESOL**. The program will list all applicable element results for the selected elements.

Sample Listing of PRNSOL,S

```
PRINT S     NODAL SOLUTION PER NODE

***** POST1 NODAL STRESS LISTING *****

 LOAD STEP=      5  SUBSTEP=      2
  TIME=    1.0000       LOAD CASE=    0

 THE FOLLOWING X,Y,Z VALUES ARE IN GLOBAL COORDINATES

  NODE     SX          SY          SZ          SXY         SYZ         SXZ
     1   148.01      -294.54      .00000E+00  -56.256      .00000E+00  .00000E+00
     2   144.89      -294.83      .00000E+00   56.841      .00000E+00  .00000E+00
     3   241.84       73.743      .00000E+00  -46.365      .00000E+00  .00000E+00
     4   401.98      -18.212      .00000E+00  -34.299      .00000E+00  .00000E+00
     5   468.15      -27.171      .00000E+00   .48669E-01  .00000E+00  .00000E+00
     6   401.46      -18.183      .00000E+00   34.393      .00000E+00  .00000E+00
     7   239.90       73.614      .00000E+00   46.704      .00000E+00  .00000E+00
     8  -84.741      -39.533      .00000E+00   39.089      .00000E+00  .00000E+00
     9   3.2868      -227.26      .00000E+00   68.563      .00000E+00  .00000E+00
    10  -33.232      -99.614      .00000E+00   59.686      .00000E+00  .00000E+00
    11  -520.81      -251.12      .00000E+00   .65232E-01  .00000E+00  .00000E+00
    12  -160.58      -11.236      .00000E+00   40.463      .00000E+00  .00000E+00
    13  -378.55       55.443      .00000E+00   57.741      .00000E+00  .00000E+00
    14  -85.022      -39.635      .00000E+00  -39.143      .00000E+00  .00000E+00
    15  -378.87       55.460      .00000E+00  -57.637      .00000E+00  .00000E+00
    16  -160.91      -11.141      .00000E+00  -40.452      .00000E+00  .00000E+00
    17  -33.188      -99.790      .00000E+00  -59.722      .00000E+00  .00000E+00
    18   3.1090      -227.24      .00000E+00  -68.279      .00000E+00  .00000E+00
    19   41.811       51.777      .00000E+00  -66.760      .00000E+00  .00000E+00
    20  -81.004       9.3348      .00000E+00  -63.803      .00000E+00  .00000E+00
    21   117.64      -5.8500      .00000E+00  -56.351      .00000E+00  .00000E+00
```

```
   22   -128.21      30.986     .00000E+00 -68.019     .00000E+00  .00000E+00
   23    154.69     -73.136     .00000E+00  .71142E-01 .00000E+00  .00000E+00
   24   -127.64     -185.11     .00000E+00  .79422E-01 .00000E+00  .00000E+00
   25    117.22     -5.7904     .00000E+00  56.517     .00000E+00  .00000E+00

   26   -128.20      31.023     .00000E+00  68.191     .00000E+00  .00000E+00
   27    41.558      51.533     .00000E+00  66.997     .00000E+00  .00000E+00
   28   -80.975      9.1077     .00000E+00  63.877     .00000E+00  .00000E+00

 MINIMUM VALUES
 NODE        11         2           1          18           1           1
 VALUE   -520.81     -294.83     .00000E+00 -68.279     .00000E+00  .00000E+00

 MAXIMUM VALUES
 NODE         5         3           1           9           1           1
 VALUE    468.15      73.743     .00000E+00  68.563     .00000E+00  .00000E+00
```

5.3.4.2. Listing Reaction Loads and Applied Loads

You have several options in POST1 for listing reaction loads and applied loads. The **PRRSOL** command (**Main Menu> General Postproc> List Results> Reaction Solu**) lists reactions at constrained nodes in the selected set. The **FORCE** command dictates which component of the reaction data is listed: total (default), static, damping, or inertia. **PRNLD** (**Main Menu> General Postproc> List Results> Nodal Loads**) lists the summed element nodal loads for the selected nodes, except for any zero values.

Listing reaction loads and applied loads is a good way to check equilibrium. It is always good practice to check a model's equilibrium after solution. That is, the sum of the applied loads in a given direction should equal the sum of the reactions in that direction. (If the sum of the reaction loads is not what you expect, check your loading to see if it was applied properly.)

The presence of coupling or constraint equations can induce either an actual or apparent loss of equilibrium. Actual loss of load balance can occur for poorly specified couplings or constraint equations (a usually undesirable effect). Coupled sets created by **CPINTF** and constraint equations created by **CEINTF** or **CERIG** will in nearly all cases maintain actual equilibrium. Also, the sum of nodal forces for a DOF belonging to a constraint equation does not include the force passing through that equation, which affects both the individual nodal force and the nodal force totals. Other cases where you may see an apparent loss of equilibrium are: (a) 4-node shell elements where all 4 nodes do no lie in an exact flat plane, (b) elements with an elastic foundation specified, and (c) un-converged nonlinear solutions. See the *ANSYS, Inc. Theory Reference*.

Another useful command is **FSUM**. **FSUM** calculates and lists the force and moment summation for the selected set of nodes.

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

Sample FSUM Output

```
 *** NOTE ***
 Summations based on final geometry and will not agree with solution
  reactions.

 ***** SUMMATION OF TOTAL FORCES AND MOMENTS IN GLOBAL COORDINATES *****
  FX  =   .1147202
  FY  =   .7857315
  FZ  =   .0000000E+00
  MX  =   .0000000E+00
  MY  =   .0000000E+00
  MZ  =   39.82639

 SUMMATION POINT=  .00000E+00   .00000E+00   .00000E+00
```

The **NFORCE** command provides the force and moment summation for each selected node, in addition to an overall summation.

GUI:

Main Menu> General Postproc> Nodal Calcs> Sum @ Each Node

Sample NFORCE Output

```
        ***** POST1 NODAL TOTAL FORCE SUMMATION *****

  LOAD STEP=     3   SUBSTEP=     43

   THE FOLLOWING X,Y,Z FORCES ARE IN GLOBAL COORDINATES

   NODE      FX          FY          FZ
      1  -.4281E-01   .4212       .0000E+00
      2   .3624E-03   .2349E-01   .0000E+00
      3   .6695E-01   .2116       .0000E+00
      4   .4522E-01   .3308E-01   .0000E+00
      5   .2705E-01   .4722E-01   .0000E+00
      6   .1458E-01   .2880E-01   .0000E+00
      7   .5507E-02   .2660E-01   .0000E+00
      8  -.2080E-02   .1055E-01   .0000E+00
      9  -.5551E-03  -.7278E-02   .0000E+00
     10   .4906E-03  -.9516E-02   .0000E+00

  *** NOTE ***
  Summations based on final geometry and will not agree with solution
   reactions.

  ***** SUMMATION OF TOTAL FORCES AND MOMENTS IN GLOBAL COORDINATES *****
   FX  =   .1147202
   FY  =   .7857315
   FZ  =   .0000000E+00
   MX  =   .0000000E+00
   MY  =   .0000000E+00
   MZ  =   39.82639

   SUMMATION POINT=  .00000E+00  .00000E+00  .00000E+00
```

The **SPOINT** command defines the point (any point other than the origin) about which moments are summed.

GUI:

Main Menu> General Postproc> Nodal Calcs> Summation Pt> At Node
Main Menu> General Postproc> Nodal Calcs> Summation Pt> At XYZ Loc

5.3.4.3. Listing Element Table Data

To list specified data stored in the element table, use one of the following:
Command(s): PRETAB
GUI: Main Menu> General Postproc> Element Table> List Elem Table
Main Menu> General Postproc> List Results> Elem Table Data

To list the sum of each column in the element table, use the **SSUM** command (**Main Menu> General Postproc> Element Table> Sum of Each Item**).

Sample PRETAB and SSUM Output

```
   ***** POST1 ELEMENT TABLE LISTING *****

   STAT    CURRENT      CURRENT      CURRENT
   ELEM    SBYTI        SBYBI        MFORYI
      1   .95478E-10  -.95478E-10  -2500.0
      2  -3750.0       3750.0       -2500.0
```

```
        3   -7500.0      7500.0     -2500.0
        4   -11250.      11250.     -2500.0
        5   -15000.      15000.     -2500.0
        6   -18750.      18750.     -2500.0
        7   -22500.      22500.     -2500.0
        8   -26250.      26250.     -2500.0
        9   -30000.      30000.     -2500.0
       10   -33750.      33750.     -2500.0
       11   -37500.      37500.      2500.0
       12   -33750.      33750.      2500.0
       13   -30000.      30000.      2500.0
       14   -26250.      26250.      2500.0
       15   -22500.      22500.      2500.0
       16   -18750.      18750.      2500.0
       17   -15000.      15000.      2500.0
       18   -11250.      11250.      2500.0
       19   -7500.0      7500.0      2500.0
       20   -3750.0      3750.0      2500.0

     MINIMUM VALUES
     ELEM        11           1           8
     VALUE    -37500.     -.95478E-10  -2500.0

     MAXIMUM VALUES
     ELEM         1          11          11
     VALUE     .95478E-10  37500.      2500.0

     SUM ALL THE ACTIVE ENTRIES IN THE ELEMENT TABLE

     TABLE LABEL     TOTAL
     SBYTI     -375000.
     SBYBI      375000.
     MFORYI     .552063E-09
```

5.3.4.4. Other Listings

You can list other types of results with the following commands:

The **PRVECT** command (**Main Menu> General Postproc> List Results> Vector Data**) lists the magnitude and direction cosines of specified vector quantities for all selected elements.

The **PRPATH** command (**Main Menu> General Postproc> List Results> Path Items**) calculates and then lists specified data along a predefined geometry path in the model. You must define the path and map the data onto the path; see Section 5.3.5: Mapping Results onto a Path.

The **PRSECT** command (**Main Menu> General Postproc> List Results> Linearized Strs**) calculates and then lists linearized stresses along a predefined path.

The **PRERR** command (**Main Menu> General Postproc> List Results> Percent Error**) lists the percent error in energy norm for all selected elements.

The **PRITER** command (**Main Menu> General Postproc> List Results> Iteration Summry**) lists iteration summary data.

5.3.4.5. Sorting Nodes and Elements

By default, all tabular listings usually progress in ascending order of node numbers or element numbers. You can change this by first sorting the nodes or elements according to a specified result item. The **NSORT** command (**Main Menu> General Postproc> List Results> Sorted Listing> Sort Nodes**) sorts nodes based on a specified nodal solution item, and **ESORT** (**Main Menu> General Postproc> List Results> Sorted Listing> Sort Elems**) sorts elements based on a specified item stored in the element table. For example:

```
NSEL,...                    ! Selects nodes
NSORT,S,X                   ! Sorts nodes based on SX
PRNSOL,S,COMP               ! Lists sorted component stresses
```

See the **NSEL**, **NSORT**, and **PRNSOL** command descriptions in the *ANSYS Commands Reference* for further inform-
ation.

Sample PRNSOL,S and Output after NSORT

```
PRINT S     NODAL SOLUTION PER NODE

 ***** POST1 NODAL STRESS LISTING *****

 LOAD STEP=     3  SUBSTEP=     43
  TIME=    6.0000       LOAD CASE=    0

 THE FOLLOWING X,Y,Z VALUES ARE IN GLOBAL COORDINATES

   NODE    SX           SY           SZ           SXY          SYZ          SXZ
    111   -.90547      -1.0339      -.96928      -.51186E-01   .00000E+00   .00000E+00
     81   -.93657      -1.1249      -1.0256      -.19898E-01   .00000E+00   .00000E+00
     51   -1.0147      -.97795      -.98530       .17839E-01   .00000E+00   .00000E+00
     41   -1.0379      -1.0677      -1.0418      -.50042E-01   .00000E+00   .00000E+00
     31   -1.0406      -.99430      -1.0110       .10425E-01   .00000E+00   .00000E+00
     11   -1.0604      -.97167      -1.0093      -.46465E-03   .00000E+00   .00000E+00
     71   -1.0613      -.95595      -1.0017       .93113E-02   .00000E+00   .00000E+00
     21   -1.0652      -.98799      -1.0267       .31703E-01   .00000E+00   .00000E+00
     61   -1.0829      -.94972      -1.0170       .22630E-03   .00000E+00   .00000E+00
    101   -1.0898      -.86700      -1.0009      -.25154E-01   .00000E+00   .00000E+00
      1   -1.1450      -1.0258      -1.0741       .69372E-01   .00000E+00   .00000E+00

 MINIMUM VALUES
 NODE          1          81           1          111          111          111
 VALUE   -1.1450      -1.1249      -1.0741      -.51186E-01   .00000E+00   .00000E+00

 MAXIMUM VALUES
 NODE        111         101         111           1          111          111
 VALUE   -.90547      -.86700      -.96928       .69372E-01   .00000E+00   .00000E+00
```

To restore the original order of nodes or elements, use the following:

Command(s):
    NUSORT

GUI:
    Main Menu> General Postproc> List Results> Sorted Listing> Unsort Nodes

Command(s):
    EUSORT

GUI:
    Main Menu> General Postproc> List Results> Sorted Listing> Unsort Elems

5.3.4.6. Customizing Your Tabular Listings

In some situations you may need to customize result listings to your specifications. The **/STITLE** command (which
has no GUI equivalent) allows you to define up to four subtitles which will be displayed on output listings along
with the main title. Other commands available for output customization are: **/FORMAT**, **/HEADER**, and **/PAGE**
(also without GUI equivalents). They control such things as the number of significant digits, the headers that
appear at the top of listings, the number of lines on a printed page, etc. These controls apply only to the **PRRSOL**,
PRNSOL, **PRESOL**, **PRETAB**, and **PRPATH** commands.

5.3.5. Mapping Results onto a Path

One of the most powerful and useful features of POST1 is its ability to map virtually any results data onto an arbitrary path through your model. This enables you to perform many arithmetic and calculus operations along this path to calculate meaningful results: stress intensity factors and J-integrals around a crack tip, the amount of heat crossing the path, magnetic forces on an object, and so on. A useful side benefit is that you can see, in the form of a graph or a tabular listing, how a result item varies along the path.

> *Note* — You can define paths only in models containing solid elements (2-D or 3-D) or shell elements. They are not available for line elements.

Three steps are involved in reviewing results along a path:

1. Define the path attributes [**PATH** command].
2. Define the path points [**PPATH** command].
3. Interpolate (map) results data along the path [**PDEF** command].

Once the data are interpolated, you can review them using graphics displays [**PLPATH** or **PLPAGM** commands] and tabular listings or perform mathematical operations such as addition, multiplication, integration, etc. Advanced mapping techniques to handle material discontinuities and accurate computations are offered in the **PMAP** command (issue this command prior to **PDEF**).

Other path operations you can perform include archiving paths or path data to a file or an array parameter and recalling an existing path with its data. The next few topics discuss path definition and usage.

5.3.5.1. Defining the Path

To define a path, you first define the path environment and then the individual path points. Decide whether you want to define the path by picking nodes, by picking locations on the working plane, or by filling out a table of specific coordinate locations. Then create the path by picking or by using both of the commands shown below or one of the following menu paths:

> **Command(s): PATH**, **PPATH**
> **GUI: Main Menu> General Postproc> Path Operations> Define Path> By Nodes**
> **Main Menu> General Postproc> Path Operations> Define Path> On Working Plane**
> **Main Menu> General Postproc> Path Operations> Define Path> By Location**

Supply the following information for the **PATH** command:

- A path name (containing no more than eight characters).
- The number of path points (between 2 and 1000). Required only in batch mode, or when defining path points using the "By Location" option. When picking is used, the number of path points equals the number of picked points.
- The number of sets of data which may be mapped to this path. (Four is the minimum; default is 30. There is no maximum.)
- The number of divisions between adjacent points. (Default is 20; there is no maximum.)
- When using the "By Location" option, a separate dialog box appears for defining path points (**PPATH** command). Enter the Global Cartesian coordinate values of the path points. The shape of the interpolated path geometry will follow the currently active CSYS coordinate system. Alternatively, you can specify a coordinate system for geometry interpolation (*CS* argument on the **PPATH** command).

> *Note* — To see the status of path settings, choose the **PATH**,STATUS command.

The **PATH** and **PPATH** commands define the path geometry *in the active CSYS coordinate system*. If the path is a straight line or a circular arc, you need only the two end nodes (unless you want highly accurate interpolation, which may require more path points or divisions).

> *Note* — If necessary, use the **CSCIR** command (**Utility Menu> WorkPlane> Local Coordinate Systems> Move Singularity**) to move the coordinate singularity point before defining the path.

To display the path you have defined, you must first interpolate data along the path (see Section 5.3.5.3: Interpolating Data Along the Path). You then issue the **/PBC**,PATH,1 command followed by the **NPLOT** or **EPLOT** command. Alternatively, if you are using the GUI, choose **Main Menu> General Postproc> Path Operations> Plot Paths** to display the path on a node plot or choose **Utility Menu> Plot> Elements** followed by **Main Menu> General Postproc> Path Operations> Plot Paths** to display the path on an element plot. ANSYS displays the path as a series of straight line segments. The path shown below was defined in a cylindrical coordinate system:

Figure 5.12 A Node Plot Showing the Path
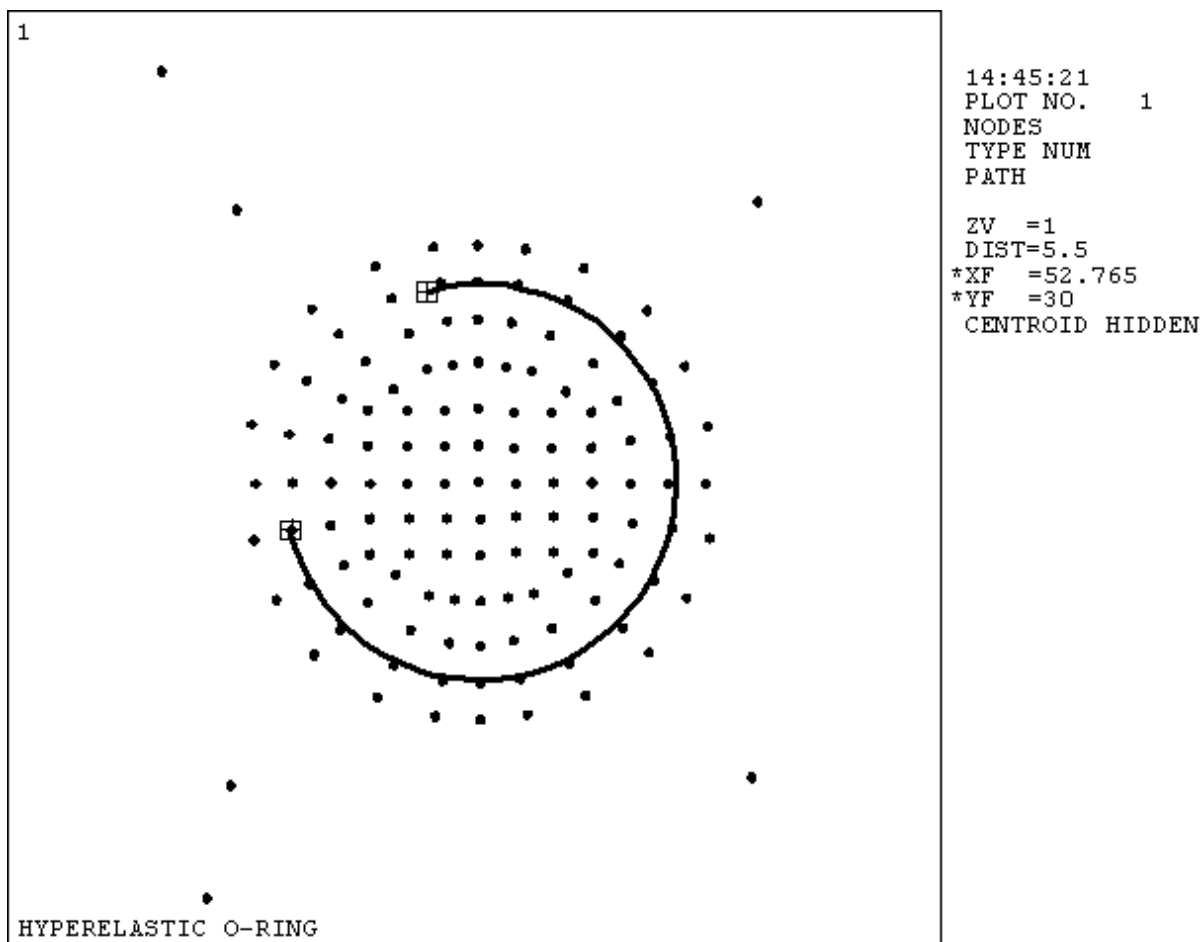
5.3.5.2. Using Multiple Paths

A maximum of 100 paths can exist within one model. However, only one path at a time can be the current path. To change the current path, choose the **PATH**,*NAME* command. Do not specify any other arguments on the **PATH** command. The named path will become the new current path.

5.3.5.3. Interpolating Data Along the Path

The following commands are available for this purpose:

Command(s):
   PDEF

GUI:
   Main Menu> General Postproc> Path Operations> *path operation*

Command(s):
   PVECT

GUI:
   Main Menu> General Postproc> Path Operations> Unit Vector

These commands require that the path be defined first.

Using the **PDEF** command, you can interpolate virtually any results data along the path *in the active results co-ordinate system*: primary data (nodal DOF solution), derived data (stresses, fluxes, gradients, etc.), element table data, FLOTRAN nodal results data, and so on. The rest of this discussion (and in other documentation) refers to an interpolated item as a *path item*. For example, to interpolate the thermal flux in the X direction along a path, the command would be as follows:

```
PDEF,XFLUX,TF,X
```

The XFLUX value is an arbitrary user-defined name assigned to the path item. TF and X together identify the item as the thermal flux in the X direction.

> *Note* — You can make the results coordinate system match the active coordinate system (used to define the path) by issuing the following pair of commands:
>
> ```
> *GET,ACTSYS,ACTIVE,,CSYS
> RSYS,ACTSYS
> ```

The first command creates a user-defined parameter (ACTSYS) that holds the value defining the currently active coordinate system. The second command sets the results coordinate system to the coordinate system specified by ACTSYS.

5.3.5.4. Mapping Path Data

POST1 uses {nDiv(nPts-1) + 1} interpolation points to map data onto the path (where *nPts* is the number of points on the path and *nDiv* is the number of path divisions between points [**PATH**]). When you create the first path item, the program automatically interpolates the following additional geometry items: XG, YG, ZG, and S. The first three are the global Cartesian coordinates of the interpolation points and S is the path length from the starting node. These items are useful when performing mathematical operations with path items (for instance, S is required to calculate line integrals). To accurately map data across material discontinuities, use the *DISCON* = MAT option on the **PMAP** command (**Main Menu> General Postproc> Path Operations> Define Path> Path Options**).

To clear path items from the path (except XG, YG, ZG, and S), issue **PDEF**,CLEAR. To form additional labeled path items by operating on existing path items, use the **PCALC** command (**Main Menu> General Postproc> Path Operations>***operation*).

The **PVECT** command defines the normal, tangent, or position vectors along the path. A Cartesian coordinate system must be active for this command. For example, the command shown below defines a unit vector tangent to the path at each interpolation point.

```
PVECT,TANG,TTX,TTY,TTZ
```

TTX, TTY, and TTZ are user-defined names assigned to the X, Y, and Z components of the vector. You can use these vector quantities for fracture mechanics J-integral calculations, dot and cross product operations, etc. For accurate mapping of normal and tangent vectors, use the ACCURATE option on the **PMAP** command. Issue the **PMAP** command prior to mapping data.

5.3.5.5. Reviewing Path Items

To obtain a graph of specified path items versus path distance, use one of the following:
> **Command(s): PLPATH**
> **GUI: Main Menu> General Postproc> Path Operations> Plot Path Item**

To get a tabular listing of specified path items, use one of the following:
> **Command(s): PRPATH**
> **GUI: Main Menu> General Postproc> List Results> Path Items**

You can control the path distance range (the abscissa) for **PLPATH** and **PRPATH** (**Main Menu> General Postproc> Path Operations> Path Range**) or the **PRANGE** command**.** Path defined variables may also be used in place of the path distance for the abscissa item in the path display.

You can use two other commands, **PLSECT** (**Main Menu> General Postproc> Path Operations> Linearized Strs**) and **PRSECT** (**Main Menu> General Postproc> List Results> Linearized Strs**), to calculate and review linearized stresses along a path *defined by the first two nodes on the* **PPATH** command. Typically, you use them in pressure vessel applications to separate stresses into individual components: membrane, membrane plus bending, etc. The path is defined in the active display coordinate system.

You can display a path data item as a color area contour display along the path geometry. The contour display offset from the path may be scaled for clarity. To produce such a display, use either of the following:
> **Command(s): PLPAGM**
> **GUI: Main Menu> General Postproc> Plot Results> Plot Path Items> On Geometry**

5.3.5.6. Performing Mathematical Operations among Path Items

Three commands are available for mathematical operations among path items:

The **PCALC** command (**Main Menu> General Postproc> Path Operations>** *operation*) lets you add, multiply, divide, exponentiate, differentiate, and integrate path items.

The **PDOT** command (**Main Menu> General Postproc> Path Operations> Dot Product**) calculates the dot product of two path vectors.

The **PCROSS** command (**Main Menu> General Postproc> Path Operations> Cross Product**) calculates the cross product or two path vectors.

5.3.5.7. Archiving and Retrieving Path Data to a File

If you wish to retain path data when you leave POST1, you must store it in a file or an array parameter so that you can retrieve it later. You first select a path or multiple paths and then write the current path data to a file:

Command(s):
> **PSEL**

GUI:
> **Utility Menu> Select> Paths**

Command(s):
> **PASAVE**

GUI:
> **Main Menu> General Postproc> Path Operations> Archive Path> Store> Paths in file**

To retrieve path information from a file and store the data as the currently active path data, use the following:
> **Command(s): PARESU**
> **GUI: Main Menu> General Postproc> Path Operations> Archive Path> Retrieve> Paths from file**

You can opt to archive or fetch only the path data (data mapped to path (**PDEF** command) or the path points (defined by the **PPATH** command). When you retrieve path data, it becomes the currently active path data (existing active path data is replaced). If you issue **PARESU** and have multiple paths, the first path from the list becomes the currently active path.

Sample input and output are shown below.

```
/post1
path,radial,2,30,35     ! Define path name, No. points, No. sets, No. divisions
ppath,1,,.2             ! Define path by location
ppath,2,,.6
pmap,,mat               ! Map at material discontinuities
pdef,sx,s,x             ! Interpret radial stress
pdef,sz,s,z             ! Interpret hoop stress
plpath,sx,sz            ! Plot stresses
pasave                  ! Store defined paths in a file
finish
/post1
paresu                  ! retrieve path data from file
plpagm,sx,,node         ! plot radial stresses on the path
finish
```

5.3.5.8. Archiving and Retrieving Path Data to an Array Parameter

Writing path data to an array is useful if you want to map a particle flow or charged particle trace onto a path (**PLTRAC**). If you wish to retain path data in an array parameter, use the command or one of the GUI paths shown below to write current path data to an array variable:
> **Command(s): PAGET**, *PARRAY*, *POPT*
> **GUI: Main Menu> General Postproc> Path Operations> Archive Path> Retrieve> Path from array**
> **Main Menu> General Postproc> Path Operations> Archive Path> Retrieve> Paths from file**

To retrieve path information from an array variable and store the data as the currently active path data, use one of the following:
> **Command(s): PAPUT**, *PARRAY*, *POPT*
> **GUI: Main Menu> General Postproc> Path Operations> Archive Path> Store> Path in array**
> **Main Menu> General Postproc> Path Operations> Archive Path> Store> Paths from file**

You can opt to archive or fetch only the path data (data mapped to path (**PDEF** command) or the path points (defined by the **PPATH** command). The setting for the *POPT* argument on **PAGET** and **PAPUT** determines what is stored or retrieved. You must retrieve path points prior to retrieving path data and labels. When you retrieve path data, it becomes the currently active path data (existing active path data is replaced).

Sample input and output are shown below.

```
/post1
path,radial,2,30,35     ! Define path name, No. points, No. sets, No. divisions
ppath,1,,.2             ! Define path by location
ppath,2,,.6
pmap,,mat               ! Map at material discontinuities
pdef,sx,s,x             ! Interpret radial stress
```

```
pdef,sz,s,z                  ! Interpret hoop stress
plpath,sx,sz                 ! Plot stresses
paget,radpts,points          ! Archive path points in array "radpts"
paget,raddat,table           ! Archive path data in array "raddat"
paget,radlab,label           ! Archive path labels in array "radlab"
finish
/post1
*get,npts,parm,radpts,dim,x  ! Retrieve number of points from array "radpts"
*get,ndat,parm,raddat,dim,x  ! Retrieve number of data points from array "raddat"
*get,nset,parm,radlab,dim,x  ! Retrieve number of data labels form array "radlab"
ndiv=(ndat-1)/(npts-1)       ! Calculate number of divisions
path,radial,npts,ns1,ndiv    ! Create path "radial" with number of sets ns1>nset
paput,radpts,points          ! Retrieve path points
paput,raddat,table           ! Retrieve path data
paput,radlab,labels          ! Retrieve path labels
plpagm,sx,,node              ! Plot radial stresses on the path
finish
```

Figure 5.13 A Sample PLPATH Display Showing Stress Discontinuity at a Material Interface
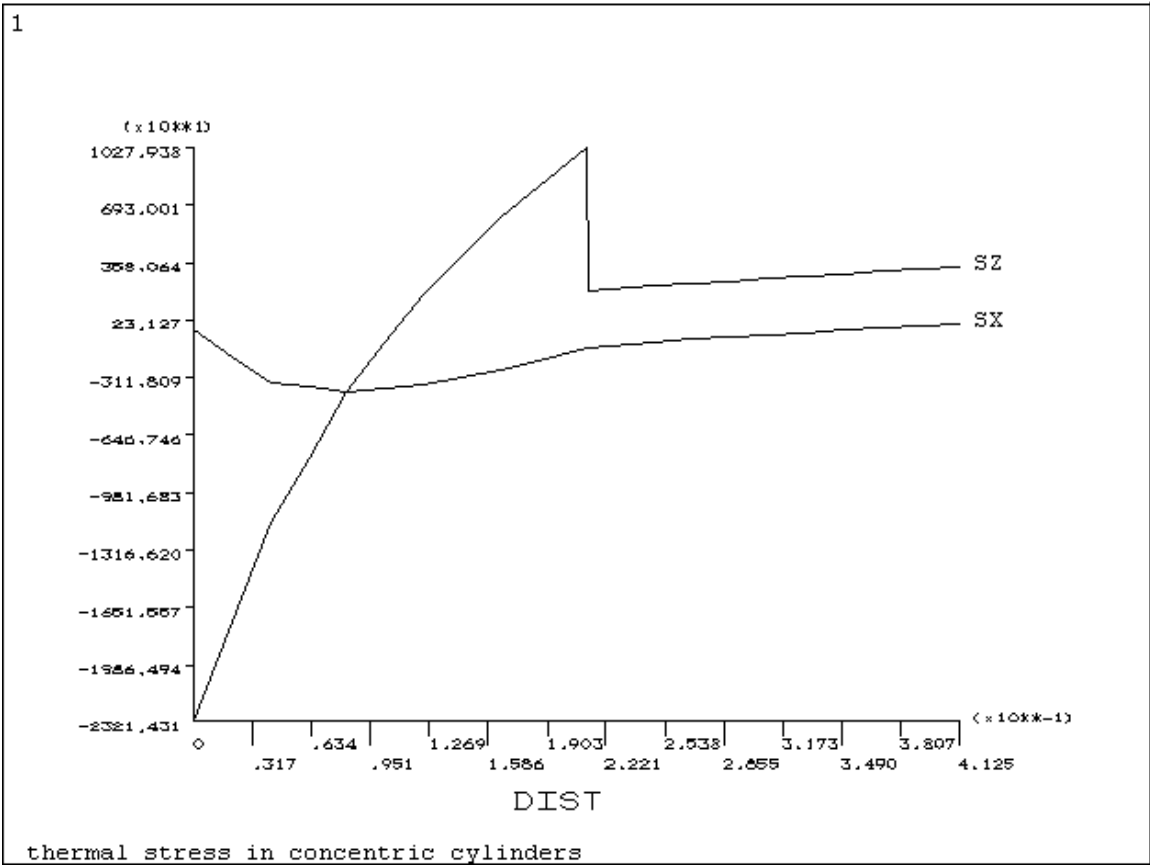
Figure 5.14 A Sample PLPAGM Display
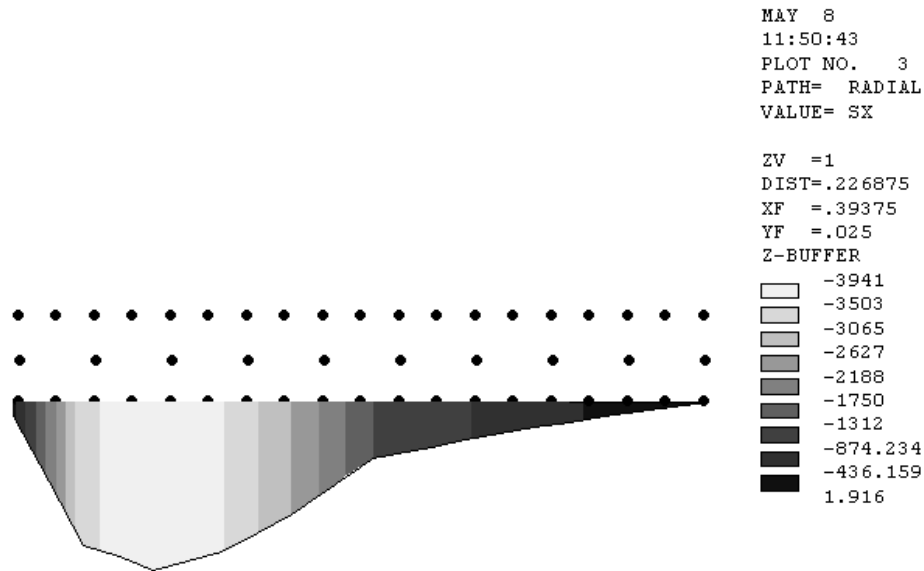
thermal stress in concentric cylinders

5.3.5.9. Deleting a Path

To delete one or more paths, use one of the following:
> **Command(s): PADELE**, *DELOPT*
> **GUI: Main Menu> General Postproc> Path Operations> Delete Path**
> **Main Menu> General Postproc> Path Operations> Delete Path**

You can opt to delete all paths or choose a path to delete by name. To review the current list of path names, issue the command **PATH**,STATUS.

5.3.6. Estimating Solution Error

One of the main concerns in a finite element analysis is the adequacy of the finite element mesh. Is the mesh fine enough for good results? If not, what portion of the model should be remeshed? You can get answers to such questions with the ANSYS error estimation technique, which estimates the amount of solution error due specifically to mesh discretization. This technique is available only for linear structural and linear/nonlinear thermal analyses using 2-D or 3-D solid elements or shell elements.

In the postprocessor, the program calculates an *energy error* for each element in the model. The energy error is similar in concept to the strain energy. The *structural* energy error (labeled SERR) is a measure of the discontinuity of the stress field from element to element, and the *thermal* energy error (TERR) is a measure of the discontinuity of the heat flux from element to element. Using SERR and TERR, the ANSYS program calculates a *percent error in energy norm* (SEPC for structural percent error, TEPC for thermal percent error).

> *Note* — Error estimation is based on stiffness and conductivity matrices that are evaluated at the reference temperatures (TREF). Error estimates, therefore, can be incorrect for elements with temperature-dependent material properties if those elements are at a temperature that is significantly different than TREF.

In many cases, you can significantly increase program speed by suppressing error estimation. This improved performance is most evident when error estimation is turned off in a thermal analysis. Therefore, you may want

to use error estimation only when needed, such as when you wish to determine if your mesh is adequate for good results.

You may turn error estimation off issuing **ERNORM**,OFF (**Main Menu> General Postproc> Options for Outp**). By default, error estimation is active. Since the value set by the **ERNORM** command is not saved on **Jobname.DB**, you will need to reissue **ERNORM**,OFF if you wish to again deactivate error estimation after resuming an analysis .

In POST1 then, you can list SEPC and TEPC for all selected elements using the **PRERR** command (**Main Menu> General Postproc> List Results> Percent Error**). The value of SEPC or TEPC indicates the relative error due to a particular mesh discretization. To find out *where* you should refine the mesh, simply produce a contour display of SERR or TERR and look for high-error regions.

Using this error estimation technique, you can set up an automated scheme whereby the mesh is automatically refined in high-error regions. This is called *adaptive meshing*. See Chapter 5, "Adaptive Meshing" in the *ANSYS Advanced Analysis Techniques Guide*. For theoretical details about error estimation, see the *ANSYS, Inc. Theory Reference*.

5.4. Using the PGR File in POST1

The PGR file is a specialized graphics object that stores the graphical information you develop during the Solution phase of your analysis. By storing this "precooked" information, you can access it in POST1 markedly (up to 10X) faster. Although the greatest benefit will be achieved when you designate the items to be saved before you initiate the solution, you can generate a PGR file, with all of the same information, from your existing results file(s). You can also append new solution data to your existing PGR file in POST1. For more information on creating your PGR file in Solution, see Section 3.12: Using the PGR File to Store Data for Postprocessing in Chapter 3 of this manual.

The three sections that follow cover the operations that are available in POST1. The links below correspond to the appropriate GUI dialog boxes.

Specifying a new PGR file in POST1, see Section 5.4.1: Specifying a New PGR File in POST1, below.

Appending an existing PGR file in POST1, see Section 5.4.2: Appending to an Existing PGR File in POST1, below.

Using the Results Viewer to view results information, see Section 5.4.3: Using the Results Viewer to Access Your Results File Data below.

5.4.1. Specifying a New PGR File in POST1

Although you obtain the greatest speed advantage by specifying the PGR file BEFORE solving your analysis, you can always generate a PGR file AFTER you have completed your solution. This process reads the data from your results file (***.RST, *.RFL, *.RTH, *.RMG**, etc.) and generates the display data, basically from scratch. Generating a new PGR file allows you to take advantage of the flexibility of the Results Viewer and the speed of the PGR file data access to view your solution data and generate plots, data tables and animations. The Results Viewer provides an ideal platform from which to review and process various portions of your solution data, or to rapidly access and compare data from multiple solutions.

Once you have completed your solution, or loaded your results file, you can create a PGR file in POST1. You create a PGR file in POST1 by selecting **Main Menu> General Postprocessor> Write PGR File**. The following dialog box appears:

Figure 5.15 The PGR File Options Dialog Box
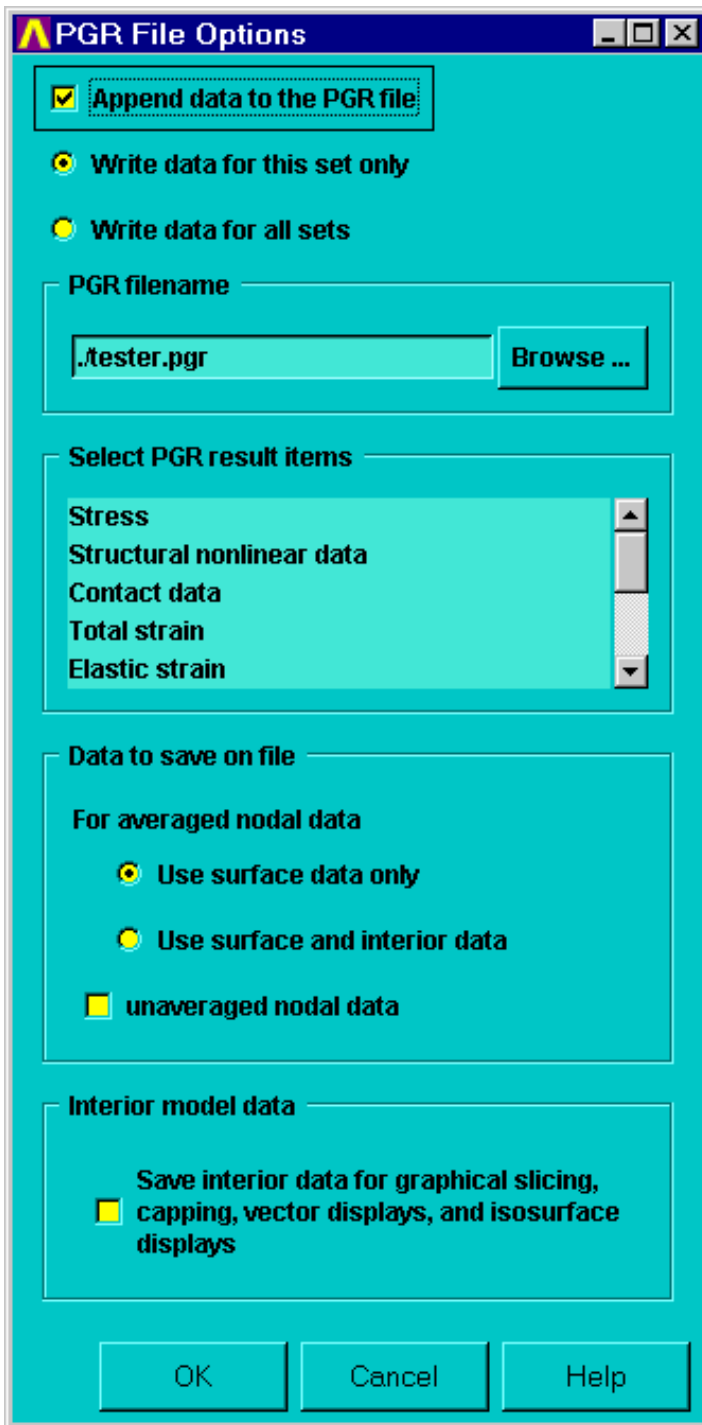
From this dialog box you choose between appending to an existing file or creating a new file. The default filename will be **Jobname.pgr**, and you can use the **Browse** button to either designate a new PGR file name and location, or to search for an existing PGR file to overwrite or append.

The Select PGR Result Items section of the PGR File Options dialog box lets you designate the data you wish to include in your PGR file. Only data that is available for the current analysis type will be displayed. Additional data can be included by returning to Solution and appending the file. You can specify the following data in PGR file options.

- Stress

- Structural nonlinear data

- Contact data (3-D only)

- Total Strain

- Elastic Strain

- Thermal Strain

- Creep Strain

- Thermal Gradient

- Thermal Flux

- Electric Field

- Electric Flux Density

- Magnetic Field Intensity

- Magnetic Flux Density

- Magnetic Forces

- Pressure Gradients

- Body Temperatures

- Densities for Topological Optimization

You can also create a PGR file from the Results Viewer. The Results Viewer file menu links to the PGR File Options dialog box, allowing you to write and append to a PGR file from a results file that is already loaded. You can then take advantage of the speed and flexibility of the Results Viewer and access your data faster, in a subsequent run. See Section 5.4.3.1.1: The Results Viewer Main Menu later in this section for more information on writing or appending to your PGR file from the viewer.

Stresses can only be displayed in the coordinate system that was active when the PGR file was written. If you wish to use the results viewer to view stresses in other coordinate displays, you must reload your results file (***.RST, *.RFL, *.RTH, *.RMG**, etc.).

The PGR file creation operations are controlled by the **PGWRITE** command, and can be accessed either via the menu, as described above, or via the command line.

5.4.2. Appending to an Existing PGR File in POST1

In solution, you are given the option to specify the items you want written to your PGR file. Often, you may wish to add additional data to your existing PGR file. You can append your PGR file and access all of the existing data, along with the new solution data, from the same analysis. You can also modify your analysis parameters (loading, step data, constraint data, etc.) and add that information to the PGR file. The only requirements are that you do not change any of the "data to save on file" criteria, and that you do not modify the geometry of your model. If you attempt to append a PGR file with different items in the "data to save on file" area selected, ANSYS will generate an error message and prevent you from proceeding. Once you have changed the information to align with the existing PGR data, you will be allowed to proceed. When you change the geometry of the model, ANSYS will again generate an error message. However, you will not be able to change the PGR parameters to append the file. You must solve the analysis again, and write a new PGR file.

When you choose to append to the PGR file, the current analysis filename is displayed. Changing this filename will create a new PGR file and will prevent appending the data to your existing file. If you have changed the

loading on the model, the new data is appended as an additional results set, which can be accessed with the Results Viewer, separately from the other data. If you have selected new PGR results items to be included with the file, this data is added to each of the results sets. The changes to the result item availability are written to each of the existing data sets and are available directly within the PGR file.

The PGR file append operations are controlled by the **PGWRITE** command, and can be accessed either via the menu, as described above, or via the command line.

5.4.3. Using the Results Viewer to Access Your Results File Data

The following links correspond to the three basic control areas on the Results Viewer:

For the Main Menu, see Section 5.4.3.1.1: The Results Viewer Main Menu
For the Toolbar, see Section 5.4.3.1.2: The Results Viewer Toolbar
For the Step/Sequence Data Access Control, see Section 5.4.3.2: The Results Viewer Step/Sequence Data Access Controls

Figure 5.16 The Results Viewer
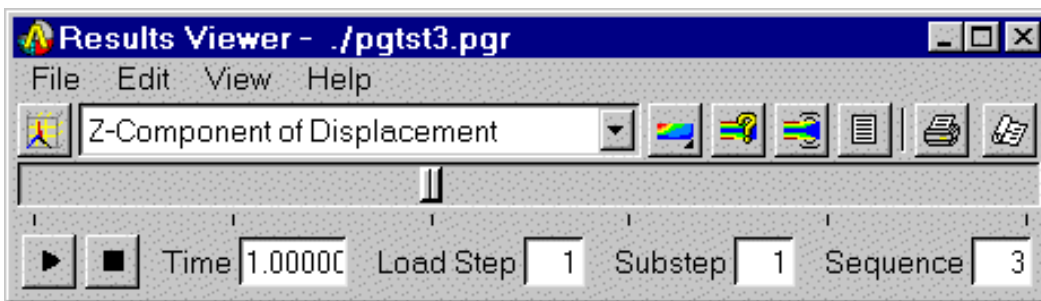
When you enter POST1, the available operations for the PGR data are "Results Viewer" or "Write PGR File." The "Write PGR File" options are explained above. Choosing the Results Viewer disables much of the standard ANSYS GUI functionality. Many of these operations are not available because of PowerGraphics limitations. However, a good deal of the POST1 functionality is contained in the Result Viewer menu structure, and in the right and middle mouse button context sensitive menus that are accessible when you use the Results Viewer. The Results Viewer is described in the following paragraphs.

The ANSYS Results Viewer is a compact toolbar for viewing your analysis results. Although it is designed to display the information in your PGR file, you can use it to access any data you have stored in a valid results file (***.RST, *.RFL, *.RTH, *.RMG**, etc.). When you open the Results Viewer, it accesses the PGR file you created for your current analysis, if one exists. You also have the option to open other PGR or results files. Because the viewer can access your results data without loading the entire database file, it is an ideal location from which to compare data from many different analyses.

Even if you have loaded other PGR or results files, you are still able to return to your original analysis. You can either reload the original PGR or results file from the current analysis before closing the Results Viewer, or after closing the Results Viewer, issue the **PGRAPH**,ON,S command, where S is the job name for your current analysis.

5.4.3.1. The Results Viewer Layout

There are three basic control areas on the Results Viewer: The Main Menu, The Toolbar and The Step/Sequence Data Access Control. Each of these areas is described below.

5.4.3.1.1. The Results Viewer Main Menu

The Main Menu is located along the top of the Results Viewer and provides access to the File, Edit, View and Help menus. The following functions can be accessed from each of these headings.

Figure 5.17 The Results Viewer File Menu
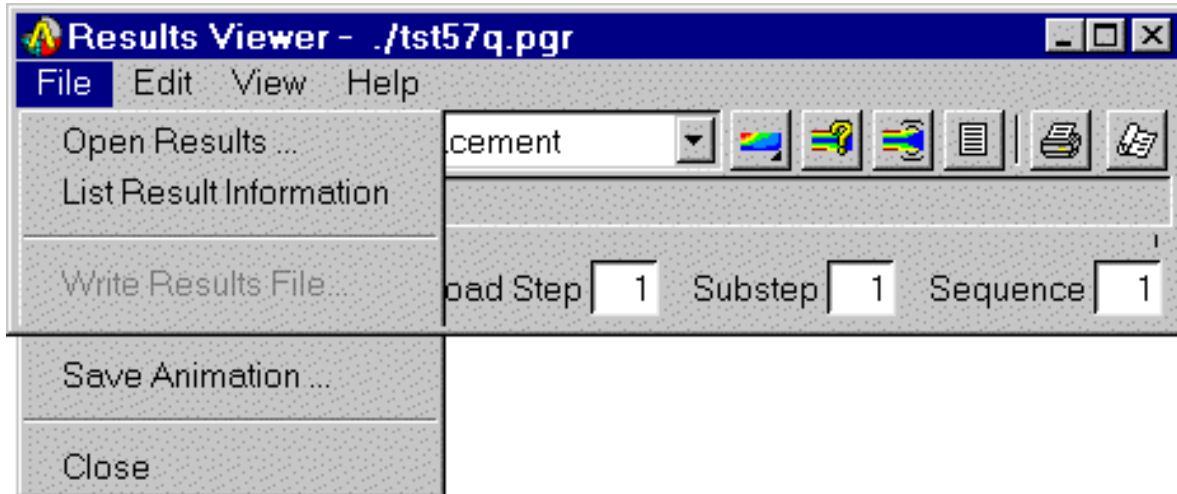
File -

Open Results -
> You can open any PGR file, or any results file (***.RST, *.RFL, *.RTH, *.RMG**, etc.) from any location on your file system.

List Result Information -
> This selection displays a list of all results data included in the current file, and information about the current sequence for a PGR file.

Write Results -
> You can use the data from your results file to create a new PGR file. This selection brings up the PGR File options dialog box and allows you to specify the creation of a new PGR file from any results file.

Save Animation -
> Save an animation file (*.anim, *.avi) to a specified location. Animations created from the Results Viewer are not stored in the PGR file and are not written to the data base.

Close -
> This option closes the Results Viewer and reverts back to the standard ANSYS GUI. If you have opened the results or PGR file from another analysis, you should return to your original file before closing the Results Viewer.

Edit -
> You can select subsets of the model based on model attributes (material, element type, real ID, and element component). This selection leads to a specialized PGR menu that allows you to select from a list of material identifications, element types, element component designations and real constant values. For the results files, this brings up the appropriate "Element Select" widget, or picking window.

Figure 5.18 The Results Viewer View Menu
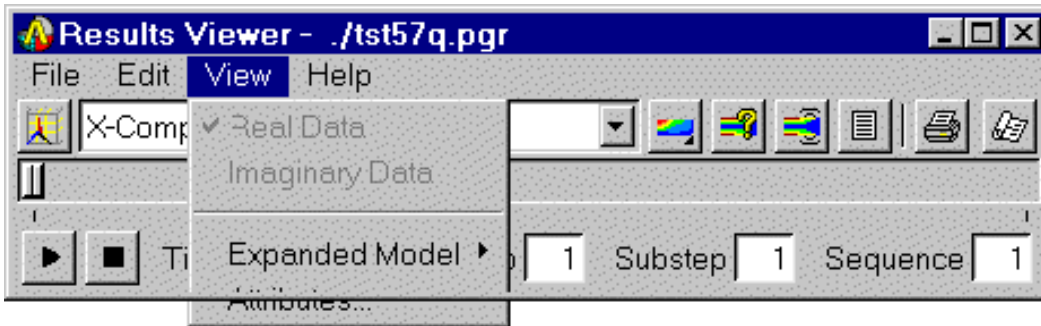
View -

> Real Data -
> > You can display the real data from your analysis in the graphics window. This selection is grayed out when only real data exists for your analysis.
>
> Imaginary Data -
> > You can display the imaginary data for you analysis in the graphics window. This selection is grayed out when no valid imaginary data exists.
>
> Expanded Model -
> > You can perform all of the periodic/cyclic, modal cyclic and axisymmetric expansions that are available from the **/EXPAND** command.
>
> Attributes -
> > The attributes of your model can be accessed according to the conventions in the **/PNUM** command.

Help -
> Selecting help directs you to the list of PGR commands and documentation links located at the beginning of this section. You can then navigate to the appropriate area of the documentation.

5.4.3.1.2. The Results Viewer Toolbar

The Results Viewer toolbar is located across the middle of the Results Viewer. You can choose the type of results data to plot, and designate how the information should be plotted. You can also query results data from the graphics display, create animations, generate results listings, plot or generate file exports of your screen contents, or open the HTML Report Generator to construct a report on the results data.

Figure 5.19 The Results Viewer Toolbar

Element Plot
> The first item on the toolbar is the element plot icon. This is the only model display available.

Result Item Selector
> This drop down menu allows you to choose from the various types of data. The choices displayed may not always be available in your results file.

Plot Type Selector
> You left mouse click and hold down on this button and it produces a "fly out" that allows you to access the four types of results plots available - Nodal, Element, Vector and Trace.

Query Results
> You use the query tool to retrieve results data directly from selected areas in the graphics window. The ANSYS picking menu is displayed, allowing you to select multiple items. The information is displayed only for the current view.

Animate Results
> You can create animations based on the information you have included in the PGR file. Because this information is created as a separate file, it is not saved within the PGR file. You must save the individual animations using the Results Viewer's Main Menu, Save Animation function.

List Results
> The list results button creates a text listing of all of the nodal results values for the selected sequence number and result item. . You can print this data directly, or save it to a file for use in other applications.

Image Capture
> You can plot the contents of the graphics window directly to a post script enabled printer, capture the contents to another window that is created automatically, or port the contents to an exportable graphics file in any one of the following popular formats:

> PNG - Portable Network Graphics
> EPS - Encapsulated Post Script
> JPEG - Joint Photographic Exchange Group
> WRL - Virtual Reality Meta Language
> EPSI - Encapsulated Postscript with TIFF Preview
> BMP - Windows Bitmap
> WMF - Windows Metafile
> EMF - Windows Enhanced Metafile

> For Windows (PC) use, you must have a postscript enabled printer installed in order to obtain these export formats. If a postscript printer is not installed, file export is not available.

Report Generator
> This function opens the ANSYS Report Generator. You use the report generator to capture your screen contents, animations, and result listings, and save them to a report assembly tool. This tool allows you to organize the data and add text in order to assemble a complete report. For more information on the ANSYS Report Generator, see Chapter 17, "The Report Generator" later in this manual for more information on the Report Generator.

5.4.3.2. The Results Viewer Step/Sequence Data Access Controls

When you access a PGR file or a results file, the data is presented according to the sequential data sets of your original analysis. These data sets correspond to a specific time, load step, and substep of your analysis. Data is also stored in a separate sequence when you append the PGR file, or perform additional loading during an existing analysis. You use the following controls to access these different result sets.

NOTE: When you append data to your PGR file, it may disrupt the normal chronological format of the standard ANSYS results file. Time related data access functions may not always be presented in a linear chronological format in your PGR file.

Figure 5.20 The Results Viewer Step/Sequence Data Access Controls

The Data Sequence Slider Bar

    The slider bar directly under the Results Viewer Toolbar corresponds to the individual data sequences that are available for the current results file. Each tick mark along the slider represents a data set. The data sequence number is displayed in the text box at the far right of the series of boxes below the slider. You can move to any data set either by moving the slider, or by entering the sequential number of the data set in the box.

        Note — Because PGR data is not added chronologically during append operations, the sequential order of the data sets corresponds to when the data is written, not to the time within the actual analysis.

The Play and Stop Buttons

    You use these buttons to move through the selected data sequence according to the defined load steps or substeps. The play button will step you through each of the data sets and when the final (maximum) set is reached, begin moving incrementally back down.

Time

    This text box displays the time for each data set.

Load Step

    Each individual load step number is displayed. You can enter a valid load step number here and that load step will be displayed.

Substep

    Each individual load substep number is displayed. You can enter a valid substep number here and that substep will be displayed.

Sequence

    The result sets are written sequentially to the results file during an analysis. This displays the sequence number from the results file. You also create additional sequences when you append the PGR file or add new loading data to your original analysis.

5.4.3.3. The Results Viewer Context Sensitive Menus

When you enter the Results Viewer, the rest of the ANSYS GUI is disabled. This prevents conflicts between the limited data available in PGR mode, and the functionality that can be accessed from the other GUI areas. Many of the functions you will need to deal with the results data have been moved to "context sensitive" menus that you access via the right and middle mouse buttons.

The Results Viewer places your cursor in "picking mode" anytime you place the cursor in the graphics window. This allows you to select data sets and other screen operations dynamically, many times without accessing the ANSYS Picking Menu. For the Results Viewer, the standard ANSYS method of using the right mouse button to alternate between picking and unpicking has been moved to the middle mouse button. The middle mouse button allows you to change between picking and unpicking. You can choose the mode of picking desired (selection of points on the working plane, selection of existing entities in the selected set and selection of points on the screen). You can also access entity filters to limit those entities that can be selected. For the two-button PC mouse, the middle button functions are activated by using the SHIFT-RIGHT MOUSE combination.

The right mouse button is used for context menus. These menus present choices that are applicable to the current selection. If you select "Screen Picking," you will get a legend data context menu if the cursor is over a legend item, and the graphics window context menu anywhere else.

If you select "WP Picking," and no points have been selected, you will get the legend or graphics context menus. If you have already selected points on the working plane, you will get a context menu that is applicable to the type of analysis you are performing (a FLOTRAN analysis will present a trace point selection menu, etc.). If you

select "Entity Picking," and no entities are selected, you will get the standard Graphics and Legend context menus. If you have selected entities, you will get a context menu that is applicable to the current type of analysis.

Graphics Window Context Menu
> The following items are available from the context menu you get when you right mouse click anywhere in the graphics window except over the legend information.

Figure 5.21 Graphics Window Context Menu
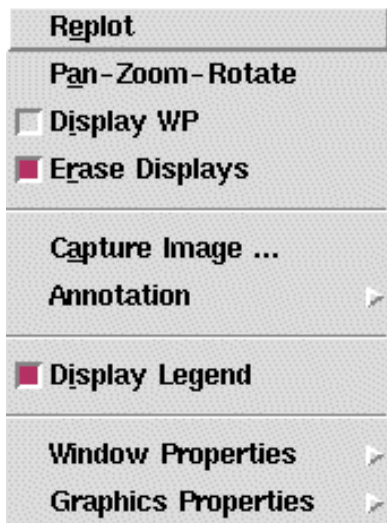
Replot -
> Replots the screen and integrates any changes you have made.

Pan-Zoom-Rotate -
> Opens the Pan-Zoom-Rotate dialog box. Specific help for this dialog box is available from the GUI.

Display WP -
> Toggles the Working Plane display (triad, grid, etc.) on and off.

Erase Displays -
> Toggles screen refreshes according to **/ERASE** - **/NOERASE** command functionality.

Capture Image -
> You can plot the contents of the graphics window directly to a printer, capture the contents to another window that is created automatically, or port the contents to an exportable graphics file in any one of the following popular formats:
>
>> PNG - Portable Network Graphics
>> EPS - Encapsulated Post Script
>> JPEG - Joint Photographic Exchange Group
>> WRL - Virtual Reality Meta Language
>> EPSI - Encapsulated Postscript with TIFF Preview
>> BMP - Windows Bitmap
>> WMF - Windows Metafile
>> EMF - Windows Enhanced Metafile
>
> For Windows (PC) use, you must have a postscript-enabled printer installed in order to obtain these export formats. If a postscript printer is not installed, file export is not available.

Annotation -
> You can choose between either static (2-D), or dynamic (3-D) annotations, and you can toggle your annotations on and off. See Section 14.1: Annotation Overview in this manual for more information on these types of annotation.

Display Legend -
> Toggles the legend display on and off

Window Properties -
> You can access limited legend and display property control, along with a number of viewing angle, rotational setting and magnification controls. The Legend Settings allow you to select which legend items will be displayed, and to specify their location in the graphics window. The Display Settings let you specify Hidden, Capped or Q-Slice displays, and to modify the Z-buffering options, including smoothing and directional light source functions or 3-D options.

Graphics Properties -
> This selection refers to those settings which affect all windows specified in a multi-window layout (**/WINDOW**) along with access to the Working Plane Settings and Working Plane Offset widgets, and the Window Layout controls dialog boxes.

Legend Area Context Menus
> The legend area context menus will vary according to the location of your cursor in the legend, and the content of the legend data already being displayed. Right mouse clicking on the legend data provides control of the legend information, while clicking on the logo provides control of the date display and other miscellaneous functions. Clicking on the contour legend area provides a complete set of menus to customize your contour legend.
>
> The legend setting and font control menus can be accessed from any of the legend context menus.

5.4.3.4. Associated PGR Commands

The following links lead to the commands associated with creating, appending and reading your PGR file:

Solution Commands
> **PGWRITE** and **POUTRES**.

POST1 Commands
> **POUTRES**, **PGSAVE**, **PGRAPH**, **PGRSET**, and **PGSELE**

5.5. Additional POST1 Postprocessing

This section documents additional methods of POST1 postprocessing.

5.5.1. Rotating Results to a Different Coordinate System

Results data, calculated during solution, consist of displacements (UX, UY, ROTX, etc.), gradients (TGX, TGY, etc.), stresses (SX, SY, SZ, etc.), strains (EPPLX, EPPLXY, etc.), etc. These data are stored in the database and on the results file in either the nodal coordinate system (for the primary, or nodal data) or the element coordinate system (for the derived, or element data). However, results data are *generally* rotated into the active results coordinate system (which is by default the global Cartesian system) for displays, listings, and element table data storage.

Using the **RSYS** command (**Main Menu> General Postproc> Options for Outp**), you can change the active results coordinate system to global cylindrical (**RSYS**,1), global spherical (**RSYS**,2), any existing local coordinate system (**RSYS**,*N*, where *N* is the local coordinate system number), or the nodal and element coordinate systems

used during solution (**RSYS**,SOLU). If you then list, display, or operate on the results data, they are rotated to this results coordinate system first. You may also set the results system back to global Cartesian (**RSYS**,0).

Figure 5.22: "Rotation of Results by RSYS" illustrates how displacements are reported for several different **RSYS** settings. The displacements are in terms of the nodal coordinate systems (which are always Cartesian systems), but issuing the **RSYS** command causes those nodal systems to be rotated into the specified system. For example, **RSYS**,1 causes the results to be rotated parallel to the global cylindrical system such that UX represents a radial displacement and UY represents a tangential displacement. (Similarly, AX and AY in a magnetic analysis and VX and VY in a fluid analysis are reported as radial and tangential values for **RSYS**,1.)

Figure 5.22 Rotation of Results by RSYS

(a) Default orientation - parallel to global Cartesian system (C.S. 0)

(b) Rotated parallel to local cylindrical system (RSYS,11)

(c) Rotated parallel to global cylindrical system (RSYS,1)
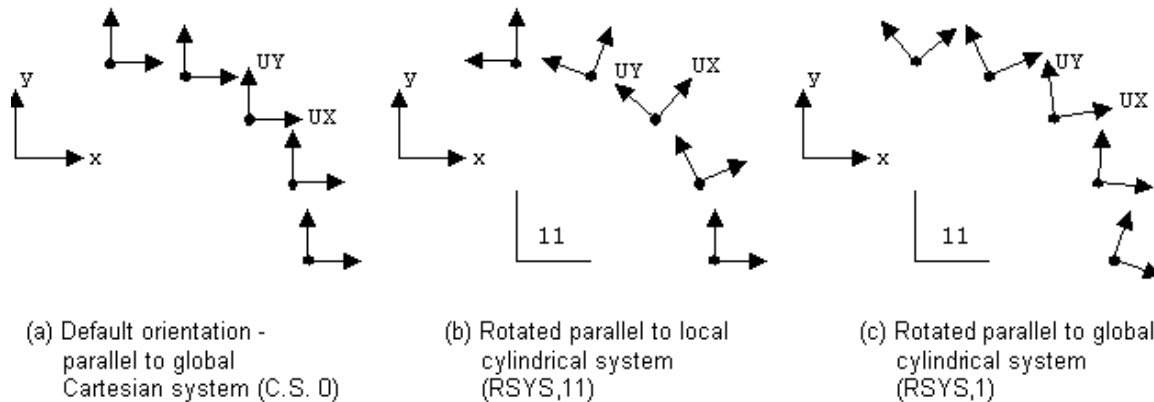
> **Caution:** Certain element results data are always output in the element coordinate system regardless of the active results coordinate system. These are miscellaneous result items that you would normally expect to be interpreted only in the element coordinate system. They include forces, moments, stresses, and strains for beam, pipe, and spar elements, and member forces and moments for some shell elements.

In most circumstances, such as when working with a single load case or during linear combinations of multiple load cases, rotating results data into the results coordinate system does not affect the final result values. However, most modal combination techniques (PSD, CQC, SRSS, etc.) are performed in the solution coordinate system and involve squaring operations. Since the squaring operation removes the sign associated with the data, some combined results may not appear as expected after being rotated into the results coordinate system. In these cases, you can use **RSYS**,SOLU to avoid a rotation and to keep the results data in the solution coordinate systems.

As an example of when you would need to change the results coordinate system, consider the case of a cylindrical shell model, in which you may be interested in the tangential stress results. The SY stress contours before and after results coordinate system transformation are shown below for such a case.

```
PLNSOL,S,Y              ! Display a: SY is in global Cartesian system (default)
RSYS,1
PLNSOL,S,Y              ! Display b: SY is in global cylindrical system
```

See the **RSYS** and **PLNSOL** command descriptions for further information.

Figure 5.23 SY in Global Cartesian and Cylindrical Systems
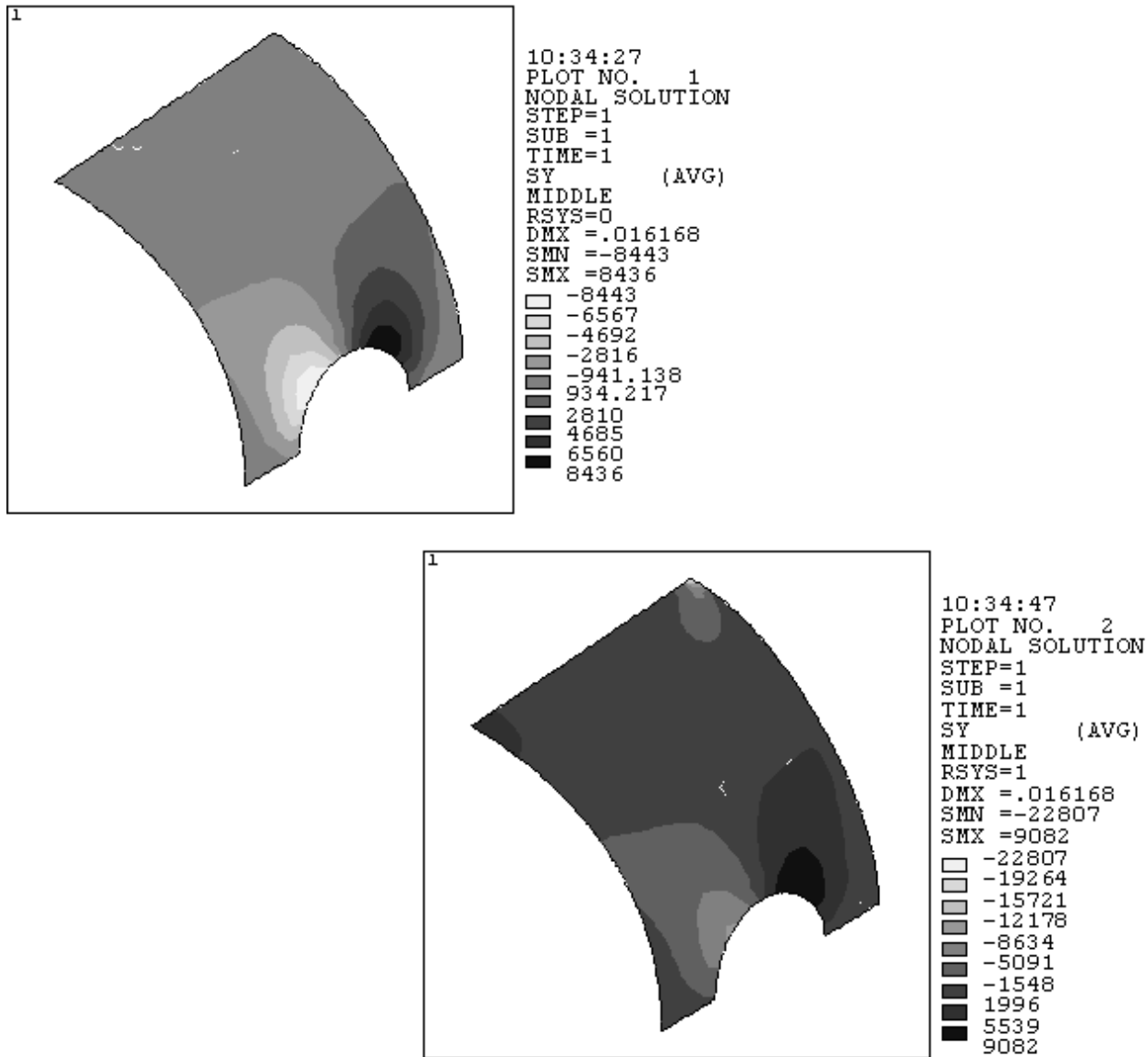
Plot 1 (top) illustrates SY in global Cartesian system. Plot 2 (bottom) illustrates SY in global cylindrical system (note that nodes and elements are still in global Cartesian system).

Note — In a large deformation analysis (for example, if you have issued the **NLGEOM**,ON command and the element has large deflection capability), the element coordinate system is first rotated by the amount of rigid body rotation of the element. Therefore the component stresses and strains and other derived element data include the effect of the rigid body rotation. The coordinate system used to display these results is the specified results coordinate system *rotated by the amount of rigid body rotation*. The exceptions to this are HYPER56, HYPER58, HYPER74, HYPER84, HYPER86, and HYPER158. These elements always produce stresses and strains in the specified results coordinate system (no rigid body rotation added). This is also true for elements that contain hyperelastic material properties (**TB**,HYPER), such as, SHELL181, PLANE182, PLANE183, SOLID185, SOLID186, SOLID187, SHELL208, and SHELL209. For these elements, when a hyperelastic material component is defined, the element component result is always in the initial global coordinate system, regardless of the coordinate system specified. For other (non-hyperelastic) material properties, the output of the element component results is in the initial global coordinate system (This is the default.). All component result transformations to other coordinate systems will be relative to the initial global system.

The primary data (for example, displacements) in a large deformation analysis *do not include the rigid body rotation effect*, because the nodal coordinate systems are not rotated by the amount of rigid body rotation.

5.5.2. Performing Arithmetic Operations Among Results Data

The earlier discussion of operations among path items was limited to items mapped on to a path. Using commands from the POST1 CALC module, you can perform operations among any results data in the database. The only requirement is that you must use the element table. The element table serves as a "worksheet" that allows arithmetic operations among its columns.

The procedure to do calculations among results data requires three simple steps:

1. Use the **ETABLE** command (**Main Menu> General Postproc> Element Table> Define Table**) to bring one or more result items into the element table or "worksheet."

2. Perform the desired arithmetic operations using commands from the CALC module (**SADD**, **SMULT**, **SEXP**, etc.).

3. Review the outcome of the operations using **PRETAB** (**Main Menu> General Postproc> Element Table> List Elem Table**) or **PLETAB** (**Main Menu> General Postproc> Element Table> Plot Elem Table**).

A discussion of the element table appears earlier in this section. The **ETABLE** command moves specified results data for all selected elements into the element table. One value is stored per element. For example, if you select 10 elements and issue the command shown below, an average UX value is calculated for each element from the nodal displacements and stored in the element table under the "ABC" column.

```
ETABLE,ABC,U,X
```

The element table will be ten rows long (because only ten elements were selected). If you now want to double these displacements, the command would be:

```
SMULT,ABC2,ABC,,2
```

For further information, see the **SMULT** command description in the *ANSYS Commands Reference*.

The element table now has a second column, labeled ABC2, containing twice the values in column ABC. To list the element table, simply choose **PRETAB** (**Main Menu> General Postproc> Element Table> List Elem Table**).

Sample Output from PRETAB

```
PRINT ELEMENT TABLE ITEMS PER ELEMENT
```

```
    ***** POST1 ELEMENT TABLE LISTING *****

    STAT     CURRENT     CURRENT
    ELEM     ABC         ABC2
       1     .21676      .43351
      11     .27032      .54064
      21     .23686      .47372
      31     .47783      .95565
      41     .36171      .72341
      51     .36693      .73387
      61     .13081      .26162
      71     .50835     1.0167
      81     .35024      .70049
      91     .25630      .51260

    MINIMUM VALUES
    ELEM          61          61
    VALUE     .13081      .26162

    MAXIMUM VALUES
    ELEM          71          71
    VALUE     .50835     1.0167
```

Another example of arithmetic operations is to calculate the total volume of selected elements. To do this, you might store all element volumes in the element table, select the desired elements, and sum them using the **SSUM** command:

```
ETABLE,VOLUME,VOLU            ! Store element volumes (VOLU) as VOLUME
ESEL,...                      ! Select desired elements
SSUM                          ! Calculate and print sum of VOLUME column
```

See the **ETABLE**, **ESEL**, and **SSUM** command descriptions for further information.

Notes

- All operation commands (**SADD**, **SMULT**, **SSUM**, etc.) work only on the selected elements.

- ANSYS does not update element table entries automatically when a different set of results is read into the database. An element table output listing displays headers which indicate the status of each column relative to the current database. CURRENT indicates that the column data is from the current database, PREVIOUS indicates that it is from a previous database, and MIXED indicates that it results from an operation between previous and current data. (Once a column is labeled mixed, it will not change status unless you erase or redefine it with all elements selected.) The following commands cause column headers to change from CURRENT to PREVIOUS:

SET	**Main Menu> General Postproc>** *selection criteria*
LCASE	**Main Menu> General Postproc> Load Case> Read Load Case**
LCOPER	**Main Menu> General Postproc> Load Case>** *operation*
LCZERO	**Main Menu> General Postproc> Load Case> Zero Load Case**
FLREAD	**Main Menu> General Postproc> Read Results> FLOTRAN 2.1A**
DESOL	**Main Menu> General Postproc> Define/Modify> Elem Results**

- The **ETABLE**,REFL option, which refills (updates) the element table with values currently in the database, does *not* affect calculated items. In the above double-the-UX example, if you read in a different set of results and then issue **ETABLE** only the ABC column will be updated ("current" status). The ABC2 column remains as is (keeps its "previous" status).

- You can use the fact that ANSYS does not update the element table automatically after a new **SET** command to good advantage: for example, to compare element results between two or more load steps, or even between two or more *analyses*.

- The following CALC module commands apply to calculations using the element table:

  The **SABS** command (**Main Menu> General Postproc> Element Table> Abs Value Option**) causes absolute values to be used in subsequent element table operations.

  The **SADD** command (**Main Menu> General Postproc> Element Table> Add Items**) adds two specified columns in the element table.

  The **SALLOW** command (**Main Menu> General Postproc> Safety Factor> `factor type`**) defines allowable stresses for safety factor calculations.

  The **SEXP** command (**Main Menu> General Postproc> Element Table> Exponentiate**) exponentiates and multiplies two columns in the element table.

  The **SFACT** command (**Main Menu> General Postproc> Safety Factor> Restore NodeStrs** or**Main Menu> General Postproc> Safety Factor> SF for Node Strs**) defines which safety factor calculations will be performed during subsequent display, select, or sort operations.

  The **SFCALC** command (**Main Menu> General Postproc> Safety Factor> SF for ElemTable**) calculates safety factors (for ETABLE items).

  The **SMAX** command (**Main Menu> General Postproc> Element Table> Find Maximum**) compares and stores the maximum of two columns.

  The **SMIN** command (**Main Menu> General Postproc> Element Table> Find Minimum**) compares and stores the minimum of two columns.

  The **SMULT** command (**Main Menu> General Postproc> Element Table> Multiply**) multiplies two specified columns in the element table.

  The **SSUM** command (**Main Menu> General Postproc> Element Table> Sum of Each Item**) calculates and prints the sum of each element table column.

  The **TALLOW** command (**Main Menu> General Postproc> Safety Factor> Allowable Strs> Reset Temps** or **Main Menu> General Postproc> Safety Factor> Allowable Strs> Temp-depend**) defines the temperature table for safety factor calculations.

  The **VCROSS** command (**Main Menu> General Postproc> Element Table> Cross Product**) calculates the cross product of two vectors stored in the element table.

  The **VDOT** command (**Main Menu> General Postproc> Element Table> Dot Product**) calculates the dot product of two vectors stored in the element table.

5.5.3. Creating and Combining Load Cases

In a typical postprocessing session, you read one set of data (load step 1 data, for instance) into the database and process it. Each time you store a new set of data, POST1 clears the results portion of the database and then brings in the new results data. If you want to perform operations between two entire sets of results data (such as comparing and storing the maximum of two sets), you need to create *load cases*.

A *load case* is a set of results data that has been assigned an arbitrary reference number. For instance, you can define the set of results at load step 2, substep 5 as *load case number 1*, the set of results at time = 9.32 as *load case number 2*, and so on. You can define up to 99 load cases, but you can store only one load case in the database at a time.

A *load case combination* is an operation between load cases, typically between the load case currently in the database and a load case on a separate results file (or on the *load case file*, explained later). The outcome of the operation overwrites the results portion of the database, which permits you to display and list the load case combination.

A typical load case combination involves the following steps:

1. Define load cases using the **LCDEF** command (**Main Menu> General Postproc> Load Case> Create Load Case**).

2. Read *one* of the load cases into the database using the **LCASE** command (**Main Menu> General Postproc> Load Case> Read Load Case**).

3. Perform the desired operation using the **LCOPER** command (**Main Menu> General Postproc> Load Case>** *operation*).

As an example, suppose the results file contains results for several load steps, and you want to compare load steps 5 and 7 and store the maximum in memory. The commands to do this would look like this:

```
LCDEF,1,5                 ! Load case 1 points to load step 5
LCDEF,2,7                 ! Load case 2 points to load step 7
LCASE,1                   ! Reads load case 1 into memory
LCOPER,MAX,2              ! Compares database with load case 2 and stores the
                         !  maximum in memory
```

The database now contains the maximum of the two load cases, and you can perform any desired postprocessing function.

> *Note* — Load case operations (**LCOPER**) are performed only on the raw solution results in the solution coordinate system.

The solution results are:

- Element component stresses and strains in the element coordinate systems
- Nodal degree-of-freedom values, forces and reaction forces in the nodal coordinate systems

To have a load case operation act on the principal/equivalent stresses instead of the component stresses, issue the **SUMTYPE**,PRIN command.

It is important that you know how load case operations are performed. Many load case operations, such as mode combinations, involve squaring, which renders the solution results unsuitable for transformation to the results coordinate system, typically the Global Cartesian, and unsuitable for performing nodal or element averages. A typical postprocessing function such as printing or displaying average nodal stresses [**PRNSOL**, **PLNSOL**], for example, involves both a coordinate system transformation to the results coordinate system and a nodal average. Furthermore, unless **SUMTYPE**,PRIN has been requested, principal/equivalent stresses are not meaningful when computed from squared component values.

To view correct mid-surface results for shells (**SHELL**, MID), use KEYOPT (8) = 2 (for SHELL181 and SHELL93) or KEYOPT (11) = 2 (SHELL63). These KEYOPT settings write the mid-surface results directly to the results file, and allow the mid-surface results to be directly operated on during squaring operations, instead of averaging the squared TOP and BOTTOM results.

Therefore, it is recommended that you use only untransformed [**RSYS**,SOLU], unaveraged [**PRESOL**, **PLESOL**] results whenever you perform a squaring operation, such as in a spectrum (SPRS, PSD) analysis.

5.5.3.1. Saving a Combined Load Case

By default, the results of a load case combination are stored in memory, overwriting the results portion of the database. To save these results - for later review or for subsequent combinations with other load cases - use one of the following methods:

- writing the data to a load case file

- appending the data to the results file.

Use the **LCWRITE** command (**Main Menu> General Postproc> Load Case> Write Load Case**) to write the load case currently in memory to a *load case file*. The file is named **Jobname.L**nn, where nn is the load case number you assign. Using nn in subsequent load case combinations will refer to the load case stored on the load case file.

The following example illustrates the use of the **LCWRITE** command:

```
LCDEF,1,5                  ! Load case 1 points to load step 5
LCDEF,2,7                  ! Load case 2 points to load step 7
LCDEF,3,10                 ! Load case 3 points to load step 10
LCASE,1                    ! Reads load case 1 into memory
LCOPER,MAX,2               ! Stores max. of database and load case 2 in memory
LCWRITE,12                 ! Writes current load case to Jobname.L12
LCASE,3                    ! Reads load case 3 into memory
LCOPER,ADD,12              ! Adds database to load case on Jobname.L12
LCDEF,STAT                 ! Results in the following output:
```

Sample Output from LCDEF,STAT

```
 LOAD CASE=  1  SELECT= 1  ABS KEY= 0  FACTOR=  1.0000
  LOAD STEP=      5 SUBSTEP=       2 CUM. ITER.=      4  TIME/FREQ=  .25000
  FILE=beam.rst
    Simply supported beam


 LOAD CASE=  2  SELECT= 1  ABS KEY= 0  FACTOR=  1.0000
  LOAD STEP=      7 SUBSTEP=       3 CUM. ITER.=     10  TIME/FREQ=  .75000
  FILE=beam.rst
    Simply supported beam


 LOAD CASE=  3  SELECT= 1  ABS KEY= 0  FACTOR=  1.0000
  LOAD STEP=     10 SUBSTEP=       2 CUM. ITER.=     12  TIME/FREQ=  1.0000
  FILE=beam.rst
    Simply supported beam


 LOAD CASE= 12  SELECT= 1  ABS KEY= 0  FACTOR=  1.0000
  LOAD STEP=      0 SUBSTEP=       0 CUM. ITER.=      0  TIME/FREQ=  .00000E+00
  FILE=beam.l12
    Simply supported beam
```

Using the **RAPPND** command (**Main Menu> General Postproc> Write Results**), you can append the load case currently in memory to the results file. The data are stored on the results file just like any other results data set except that:

- You, not the program, assign the load step and substep numbers used to identify the data.

- Only summable and constant data are available by default; non-summable data are not written to the results file unless requested (**LCSUM** command).

The following example illustrates use of the **RAPPND** command:

```
/POST1          !  Following a 2 load step analysis
SET,1                    !  Store load step 1
LCDEF,1,2                !  Identify load step 2 as load case 1
LCOPER,ADD,1             !  Add load case 1 to database (ls 1 + ls 2)
RAPPND,3,3               !  Append the combined results to the results file
!  as ls 3, time = 3
SET,LIST                 !  Observe addition of new load step to results file
```

You can use the **RAPPND** command to combine results from two results files (created with the same database.) You can use the POST1 **FILE** command (**Main Menu> General Postproc> Data & File Opts**) to "toggle" between the two results files to alternately store results from one and append to the other.

Notes

- You can define a load case by setting a pointer to a load case file with the **LCFILE** command (**Main Menu> General Postproc> Load Case> Create Load Case**). Then, you can use the **LCASE** or **LCOPER** commands to read data from the file into memory.

- You can erase any load case by issuing **LCDEF**,$LCNO$,ERASE, where $LCNO$ is the load case number. To erase all load cases, issue **LCDEF**,ERASE. These options not only delete all load case pointers, but also delete the appropriate load case files (those with the default file name extensions).

- To zero the results portion of the database, issue **LCZERO** (**Main Menu> General Postproc> Load Case> Zero Load Case**) or **LCOPER**,ZERO.

- The **LCABS** and **LCFACT** commands (**Main Menu> General Postproc> Load Case> Calc Options> Absolut Value** and **Main Menu> General Postproc> Load Case> Calc Options> Scale Factor**), allow you to specify absolute values and scale factors for specific load cases. The program uses these specifications when you issue either **LCASE** or **LCOPER**. ANSYS applies scale factors *after* it calculates absolute values.

- Results data read into the database from the results file (via **SET** or **LCASE** commands) will include boundary condition information (constraints and force loads). However, load cases read in from a load case file will not. Therefore, if boundary conditions appear on graphics displays after you issue an **LCASE** command (**Main Menu> General Postproc> Load Case> Read Load Case**), they are from a previously processed load case. The **LCASE** command does not reset the boundary condition information in memory.

- After a load case combination is performed for structural line elements, the principal stress data are not automatically updated in the database. Issue **LCOPER**,LPRIN to recalculate line element principal stresses based on the current component stress values.

- You can select a subset of load cases using the **LCSEL** command (**Main Menu> General Postproc> Load Case> Calc Options> Sele Ld Cases**). Once a subset is selected, you can use the label ALL in place of a load case number on load case operations.

- You cannot use load case combinations with FLOTRAN results data.

5.5.3.2. Combining Load Cases in Harmonic Element Models

For models with harmonic elements (axisymmetric elements with nonaxisymmetric loads), the loads are frequently applied in a series of load steps based on a Fourier decomposition (See the *ANSYS Elements Reference*). To get usable results combine the results of each load step in POST1. You can do so using load case combinations, by saving and summing all results data at a given circumferential angle. The following example illustrates this procedure:

```
/POST1
SET,1,1,,,,90            ! Read load step 1 with circumferential
                         !  angle of 90°
LCWRITE,1                ! Write load case 1 to load case file
SET,2,1,,,,90            ! Read load case 2, with circumferential
                         !  angle of 90°
LCOPER,ADD,1             ! Use load case operations to add results
```

```
                                   !  from first load case to second
        ESEL,S,ELEM,,1             ! Select element number 1
        NSLE,S                     ! Select all nodes on that element
        PRNSOL,S                   ! Calculate and list component stresses
        PRNSOL,S,PRIN              ! Calculate and list principal
                                   !  stresses S1, S2, S3; stress intensity
                                   !  SINT; and equivalent stress SEQV
        FINISH
```

See the **SET**, **LCWRITE**, **LCOPER**, **ESEL**, **NSLE**, and **PRNSOL** command descriptions in the *ANSYS Commands Reference* for further information.

5.5.3.3. Summable, Non-Summable, and Constant Data

By default, when you perform load case combinations in POST1, the ANSYS program combines only data that are valid for linear superposition, such as displacements and component stresses. Other data, such as plastic strains and element volumes, are not combined, because it is not appropriate or meaningful to combine such data. To determine which data should be combined and which should not, result items are grouped into *summable*, *non-summable*, and *constant* data. This grouping applies to the following POST1 database operations:

- Load case combinations (using **LCOPER**).

- Reading in a load case with active scale factors (using **LCFACT** or **LCASE**).

- Reading in results data and modifying them using the *FACT* or *ANGLE* field on the **SET** command.

Summable data are those that can "participate" in the database operations. All primary data (DOF solutions) are considered summable. Among the derived data, component stresses, elastic strains, thermal gradients and fluxes, magnetic flux density, etc. are considered summable (see Table 5.5: "Examples of Summable POST1 Results"). (For an inclusive list of summable data, see the description of the **ETABLE** command in the *ANSYS Commands Reference*.)

> *Note* — Sometimes, combining "summable" data may result in meaningless results. For example, adding nodal temperatures from two load cases of a linear, pure-conduction analysis gives meaningful results, but if convection is involved, the addition of temperatures is not meaningful. Therefore, exercise your engineering judgement when reviewing combined load cases.

Non-summable data are those that are not valid for linear superposition, such as nonlinear data (plastic strains, hydrostatic pressures), thermal strains, magnetic forces, Joule heat, etc. (see Table 5.6: "Examples of Non-Summable POST1 Results"). These data are simply set to zero when the programs performs a database operation. You may combine non-summable data using **LCSUM**,ALL before your **LCOPER** commands, but you are cautioned on interpreting these values appropriately."

Constant data are those that cannot be meaningfully combined, such as element volumes and element centroidal coordinates (see Table 5.7: "Examples of Constant POST1 Results"). These data are held constant (unchanged) when the program performs a database operation.

Table 5.5 Examples of Summable POST1 Results

"Vector" Data					
Item	Component	Item	Component	Item	Component
U	X, Y, Z	ROT	X, Y, Z	V	X, Y, Z
A	X, Y, Z	S	X, Y, Z, XY, YZ, XZ	EPEL	X, Y, Z, XY, YZ, XZ
TG	X, Y, Z	TF	X, Y, Z	PG	X, Y, Z
EF	X, Y, Z	D	X, Y, Z	H	X, Y, Z

"Vector" Data					
Item	**Component**	**Item**	**Component**	**Item**	**Component**
B	X, Y, Z	F	X, Y, Z	M	X, Y, Z
VF	X, Y, Z	CSG	X, Y, Z	JS	X, Y, Z
LS	(ALL)	LEPEL	(ALL)	SMISC	(ALL)

"Scalar" Data					
TEMP	MACH	STRM	NDEN	NVIS	EVIS
TBOT	HEAT	FLOW	AMPS	FLUX	ENDS
TE2	PRES	VOLT	MAG	ENKE	PTOT
TTOT	HFLU	HFLM	TCON	PCOE	ECON

Table 5.6 Examples of Non-Summable POST1 Results

"Vector" Data					
Item	**Component**	**Item**	**Component**	**Item**	**Component**
EPPL	X, Y, Z, XY, YZ, XZ	EPTH	X, Y, Z, XY, YZ, XZ	NL	SEPL, SRAT, HPRES, EPEQ, PSV, PLWK
FMAG	X, Y, Z	BFE	TEMP	LEPTH	(ALL)
LEPPL	(ALL)	LEPCR	(ALL)	NLIN	(ALL)
LBFE	(ALL)	NMISC	(ALL)		

"Scalar" Data			
EPSW	SENE	KENE	JHEAT

Table 5.7 Examples of Constant POST1 Results

"Vector" Data	
Item	**Component**
CENT	X, Y, Z

"Scalar" Data	
VOLU	

5.5.4. Mapping Results onto a Different Mesh or to a Cut Boundary

Just as the **PDEF** command (**Main Menu> General Postproc> Path Operations> Map onto Path**) maps results onto an arbitrary path in the model, POST1 also has the ability to map results on to an entirely new mesh or to a portion of a new mesh. This functionality is mainly used in *submodeling*, where you initially analyze a coarse mesh, build a finely meshed submodel of a region of interest, and map results data from the coarse model to the fine submodel.

POST1 offers two options for mapping results to a different mesh:

Command(s):
> **CBDOF**

GUI:
Main Menu> General Postproc> Submodeling> Interpolate DOF

Command(s):
BFINT

GUI:
Main Menu> General Postproc> Submodeling> Interp Body Forc

The **CBDOF** command maps degree-of-freedom results from the coarse model to the cut boundaries of the submodel. **BFINT** maps body force loads (mainly temperatures for a structural analysis) from the coarse model to the submodel. Both commands require a file of nodes to which results are to be mapped, and both commands *write* a file of appropriate load commands. These mapping functions or commands do not apply to offsets used with SHELL91 or SHELL99 (KEYOPT(11) $\neq$ 0]. Details of the submodeling technique are presented in the *ANSYS Advanced Analysis Techniques Guide*.

5.5.5. Creating or Modifying Results Data in the Database

You can do postprocessing without ever producing an ANSYS results file. All you need to do is create an ANSYS database containing nodes, elements, and property data, and then put your own results into the database using the following commands:

Command(s):
DESOL

GUI:
Main Menu> General Postproc> Define/Modify> Elem Results

Command(s):
DETAB

GUI:
Main Menu> General Postproc> Define/Modify> ElemTabl Data

Command(s):
DNSOL

GUI:
Main Menu> General Postproc> Define/Modify> Nodal Results

Once the data are defined, you can use almost any postprocessing function: graphics displays, tabular listings, path operations, etc.

> *Note* — Issuing the **DNSOL** command requires that you have placed the data type (stress/strain) in the element nodal records. To get around this requirement, use the **DESOL** command to add a "dummy" element stress/strain record.

ANSYS performs all load case combinations in the solution coordinate system, and the data resulting from load case combinations are stored in the solution coordinate system. The resultant data are then transformed to the active results coordinate system when listed or displayed. Therefore, unless **RSYS**,SOLU is set (no transformation of results data), you may see some unexpected results such as negative values after a square operation or negative values even when you request absolute values.

This feature is mainly intended for reading in data from your own, special-purpose program. By writing output data from that program in the form of the above commands, you can read them into POST1 and process them the same way you would ANSYS results. If you do have an ANSYS results file, it will not be affected by these commands.

5.5.6. Magnetics Command Macros

The following ANSYS magnetic command macros are also available for calculating and plotting results from a magnetic analysis:

- **CURR2D** (**Main Menu> General Postproc> Elec&Mag Calc> Element Based> Current**) calculates current flow in a 2-D conductor.

- **EMAGERR** (**Main Menu> General Postproc> Elec&Mag Calc > Element Based> Error Eval**) calculates the relative error in an electrostatic or electromagnetic field analysis.

- **EMF** (**Main Menu> General Postproc> Elec&Mag Calc> Path Based> EMF**) calculates the electromotive force (emf) or voltage drop along a predefined path.

- **FLUXV** (**Main Menu> General Postproc> Elec&Mag Calc> Path Based> Path Flux**) calculates the flux passing through a closed contour.

- **FMAGBC** (**Main Menu> Preprocessor> Loads> Define Loads> Apply> Electric> Flag> Comp. Force**) applies force boundary conditions to an element component.

- **FMAGSUM** (**Main Menu>General Postproc>Elec&Mag Calc> Component Based> Force**) summarizes electromagnetic force calculations on element components.

- **FOR2D** (**Main Menu> General Postproc> Elec&Mag Calc> Path Based> Mag Forces**) calculates magnetic forces on a body.

- **HFSWEEP** (**Main Menu> Solution> Solve> Electromagnet> HF Emag> Freq Sweep**) performs a harmonic response (over a range of frequencies) for a high-frequency electromagnetic waveguide analysis, and performs postprocessing calculations.

- **HMAGSOLV** (**Main Menu> Solution> Solve> Electromagnet> Harmonic Analys> Opt&Solv**) specifies 2-D harmonic electromagnetic solution options and initiates the solution for a harmonic analysis.

- **IMPD** (**Main Menu> General Postproc> Elec&Mag Calc> Path Based> Impedance**) calculates the impedance of a device at a particular reference plane.

- **LMATRIX** (**Main Menu> Solution> Solve> Electromagnet> Static Analysis> Induct Matrix**) calculates the inductance matrix and the total flux linkage in each coil for an arbitrary set of coils.

- **MAGSOLV** (**Main Menu> Solution> Solve> Electromagnet> Static Analysis> Opt&Solv**) specifies magnetic solution options and initiates the solution for a static analysis.

- **MMF** (**Main Menu> General Postproc> Elec&Mag Calc> Path Based> MMF**) calculates magnetomotive force along a path.

- **PERBC2D** (**Main Menu> Preprocessor> Loads> Define Loads> Apply> Magnetic> Boundary> Vector Poten> Periodic BCs**) generates periodic constraints for 2-D planar analysis.

- **PLF2D** (**Main Menu> General Postproc> Plot Results> Contour Plot> 2D Flux Lines**) generates a contour line plot of equipotentials.

- **PMGTRAN** (**Main Menu> TimeHist Postpro> Elec&Mag> Magnetics**) summarizes electromagnetic results from a transient analysis.

- **POWERH** (**Main Menu> General Postproc> Elec&Mag Calc> Element Based> Power Loss**) calculates the RMS power loss in a conducting body.

- **QFACT** (**Main Menu> General Postproc> Elec&Mag Calc> Cavity> Q-factor**) calculates the quality factor for high-frequency electromagnetic resonators from a mode-frequency solution.

- **RACE** (**Main Menu> Preprocessor> Modeling> Create> Racetrack Coil**) defines a "racetrack" current source.

- **REFLCOEF** (**Main Menu> General Postproc> Elec&Mag Calc> Port> ReflCoeff**) calculates the voltage reflection coefficient, standing wave ratio (VSWR), and return loss in a coaxial device.

- **SENERGY** (**Main Menu> General Postproc> Elec&Mag Calc> Element Based> Energy**) determines the stored magnetic energy or co-energy.

- **SPARM** (**Main Menu> General Postproc> Elec&Mag Calc> Port> S-Parameters**) calculates the scattering parameters between two ports of a rectangular waveguide with a TE10 mode excitation or a coaxial waveguide.

- **TORQ2D** (**Main Menu> General Postproc> Elec&Mag Calc> Path Based> Torque**) calculates torque on a body in a magnetic field.

- **TORQC2D** (**Main Menu> General Postproc> Elec&Mag Calc> Path Based> Circular Torq**) calculates torque on a body in a magnetic field based on a circular path.

- **TORQSUM** (**Main Menu> General Postproc> Elec & Mag Calc> Component Based> Torque**) summarizes electromagnetic Maxwell and Virtual work torque calculations on element components for 2-D planar problems.

See Chapter 10, "Electric and Magnetic Macros" in the *ANSYS Low-Frequency Electromagnetic Analysis Guide* for more information on ANSYS magnetic command macros.

Chapter 6: The Time-History Postprocessor (POST26)

You use the time-history postprocessor to review analysis results at specific locations in the model as a function of time, frequency, or some other change in the analysis parameters that can be related to time. In this mode, you can process your results data in many ways. You can construct graphics displays, chart representations or tabular listings, or you can perform math operations on your data sets. A typical time-history task would be to graph result items versus time in a transient analysis, or to graph force versus deflection in a nonlinear structural analysis.

The basic steps for using the Time-History Postprocessor are as follows:

1. Start the time-history processor, either interactively or via the command line. See Section 6.2: Entering the Time-History Postprocessor, in this chapter.

2. Define your Time-History variables. This involves not only identifying the variables, but also storing the variables. See Section 6.3: Defining Variables, in this chapter.

3. Process the variables to develop calculated data or to extract or generate related variable sets. See Section 6.4: Processing Your Variables to Develop Calculated Data , in this chapter.

4. Prepare your output. This can be via graph plots, tabular listings or file output. See Section 6.7: Reviewing the Variables, in this chapter.

6.1. The Time-History Variable Viewer

You can interactively define variables for time-history postprocessing using the variable viewer. A brief description of the variable viewer follows.

1. **TOOLBAR**

    Use the toolbar to control your Time-History operations. You can collapse the two expansion bars (2 and 4 below) and retain a compact toolbar that includes these items.

Add Data	Opens the "Add Time-History Variable" dialog. See Section 6.3: Defining Variables, later on in this chapter.
Delete Data	Clears selected variable from the Variable List
Graph Data	Graphs up to ten variables according to predefined properties. See Section 6.7: Reviewing the Variables, later on in this chapter.
List Data	Generates lists of data, including extremes, for six variables
Properties	You can specify selected variable and global properties
Import Data	Opens dialog for bringing information into the variable space. See Section 6.5: Importing Data later on in this chapter
Export Data	Opens dialog for exporting data to a file or an APDL array. See Section 6.6: Exporting Data later on in this chapter.
Overlay Data	Drop down list for selecting the data for graph overlay. See Section 6.5: Importing Data, later in this chapter
Clear Time-History Data	Clears all variables and returns settings to their default values (**RESET**).
Refresh Data	Updates variable list. This function is useful if some variables are defined outside of the variable viewer.
Results to View	Drop down list for choosing output form of complex variables (i.e. real, imaginary, amplitude or phase).

2. **Hide/Show Variable List**

    Clicking anywhere on this bar collapses the variable list in order to temporarily reduce the size of the viewer.

3. **Variable List**

    This area will display the defined Time-History variables. You can pick from within this list to select and process your variables.

4. **Hide/Show Calculator**

    Clicking anywhere on this bar collapses the calculator to reduce the size of the viewer.

5. **Variable Name Input Area**

    Enter the name (32 character max.) of the variable to be created.

6. **Expression Input Area**

    Enter the expression associated with the variable to be created.

7. **APDL Variable Drop Down List**

    Select a currently-defined APDL variable to use in the expression input.

8. **Time-History Variable Drop Down List**

    Select from previously-stored variables to use in the expression input.

9. **Calculator Area**

    Use the calculator to add standard mathematical operators and functions to the expression input. You click on the buttons to enter the function into the expression input area. Clicking on the INV button enables

the alternate selections shown above the buttons. For examples on how to use the calculator functions, see Section 6.4: Processing Your Variables to Develop Calculated Data in this chapter.

PARENTHESIS	Use the parenthesis to set off the hierarchy of operations, just as you would in any algebraic expression. Many functions will automatically insert parenthesis when needed.
MAX / MIN	Finds the largest of three variables (**LARGE**) / Finds the smallest of three variables (**SMALL**)
COMPLEX / CONJUGATE	Forms a complex variable / Forms the complex conjugate of a variable (**CONJUG**).
LN / e^X	Forms the natural log of a variable (**NLOG**) / Forms the exponential of a variable (**EXP**).
STO / RCL	Stores active information from the expression input area into a memory location / Recalls the memory location for repeated use in an expression.
CVAR	Computes the covariance between two variables (**CVAR**). Only available for random vibration (PSD) analyses.
RPSD	Computes the response power spectral density (**RPSD**). Only available for random vibration (PSD) analyses.
RESP	Computes the response power spectrum (**RESP**) from time history data. Available for transient analyses.
LOG	Forms the common log of a variable (**CLOG**).
ABS / INS MEM	Forms the absolute value of a variable. For a complex number, the absolute value is the magnitude (**ABS**) / Inserts the contents of a memory location into an expression.
ATAN	Forms the arctangent of a complex variable (**ATAN**).
X^2 / SQRT	Forms the square of a variable (**PROD**) / Forms the square root of a variable (**SQRT**) .
INV	Use this key to make the alternate calculator functions (shown above the buttons) available.
DERIV / INT	Forms the derivative of a variable (**DERIV**) / Forms the integral of a variable (**INT1**).
REAL / IMAG	Forms a variable using only the real part of a complex variable (**REALVAR**) / Forms a variable using only the imaginary part of a complex variable (**IMAGIN**).
11 KEY NUMBER PAD	Enters real numbers into the expression input area.
/	Computes the quotient of two variables (**QUOT**).
*	Computes the product of two variables (**PROD**).
−	Computes the difference between two variables (**ADD**).
+	Computes the sum of two variables (**ADD**).
CLEAR	Clears all data from the variable and expression input area.
BACKSPACE	Backspace from the current cursor location deleting preceding characters.
ENTER	Computes the expression in the expression input area and stores the result in the variable specified in the variable input area (**STORE**).

6.2. Entering the Time-History Postprocessor

You enter the time history processor to process time or frequency related results data. Once you have solved an analysis, ANSYS uses your results data to create a "Results File." The active results file (***.RST, *.RFL, *.RTH, *.RMG**, etc.) is automatically loaded when you enter postprocessing. If the current analysis contains no results file, you are queried for one. You can also use the file option to load any other results file for processing.

6.2.1. Interactive

Selecting **Main Menu> TimeHist PostPro** starts the Time-History Postprocessor and loads the time-history variable viewer. The following discussions of interactive mode deal with the variable viewer portion of the Graphical User Interface (GUI). Alternate GUI methods are discussed in the appropriate command descriptions. If you need to reopen the variable viewer while still in the Time-History Postprocessor, click **Variable Viewer** in the **TimeHist PostPro** menu.

6.2.2. Batch

The command **/POST26** opens the time-history postprocessor for batch and command line operations.

Notes:

- You must have your geometry loaded and a valid results file must be available in order to perform time-history post processing (interactive or batch)

- By default, the time-history processor looks for one of the results files mentioned in The General Postprocessor (POST1). You can specify a different file name using the **FILE** command (batch) or from the file menu of the variable viewer.

- The data sets and variable definitions you create in the time history postprocessor are maintained for the current ANSYS session. This allows you to move, for example, between POST1 and POST26 without losing stored information (see the **KEEP** command for more information).

- If you define variables outside of the variable viewer, but want to use it for postprocessing, you must refresh the variable viewer by either pressing the F5 button on your keyboard with the variable viewer selected, or by choosing the refresh button in the variable viewer's toolbar.

- Use the Clear Time-History Data button to remove all defined variables and return settings to their default values.

6.3. Defining Variables

Your time-history operations deal with *variables*, tables of result item versus time (or versus frequency). The result item may be the UX displacement at a node, the heat flux in an element, the force developed at a node, the stress in an element, the magnetic flux in an element, etc. You assign unique identifiers to each of your variables. Up to 200 such variables can be defined. TIME is reserved for the time value, and FREQ is reserved for the frequency value. All other identifiers must be unique, and can be made up of 32 letters and characters. If you don't supply a unique identifier, ANSYS will assign one. In addition to the unique identifiers, ANSYS uses numerical indices (reference numbers) to track and manipulate the variables. These numbers can be used interchangeably with the identifiers at the command level, and in some interactive operations. The numerical index is displayed, along with any name you choose in the data properties dialog box.

6.3.1. Interactive

Follow these steps to enter time-history data using the variable viewer.

1. Click on the **Add Data** button.

   Result: The "Add Time-History Variable" data selection dialog appears. Use the result item tree provided in the "Result Item" frame of this dialog to select the type of result you wish to add. Result items are presented in a hierarchical tree fashion from which you can select many standard result items (only result items available in your analysis will be displayed). A "favorites" section is provided to allow you to access previously selected data items. The last fifty entries are stored here.

2. Specify a name for the result item and provide additional information. The "Variable Name" field in the "Result Item Properties" area will display an ANSYS-assigned name, however, this field can be edited to use any name you choose. You will be asked to overwrite existing data if the name chosen is not unique. Depending on the type of result chosen from the "Result Item" area above, you may provide additional information about the item, such as the appropriate shell surface, force component or layer number information.

3. Click on the **OK** button.

    Result: If entity information is required, a picking window will appear, and you can choose the appropriate node and/or element from your model. The "Add Time-History Variable" window then closes and the appropriate variable appears in the variable viewer's variable list area.

    If you wish to enter more than one variable definition, click **Apply**, and the results data will be defined and entered into the variable list area, while still keeping the "Add Time-History Variable" window open.

4. (optional) Add or modify properties information.

    You may, depending on the type of results variable, wish to supply additional Time-History properties information. Time History Properties include specific variable information, X- axis definition data and list definition data. This information can be edited at any time via the **Data Properties** button.

Notes:

·   You can see all of your defined variables in the Variable List area. Specific element and node information, along with the appropriate range of values are all displayed here.

·   When you define your variable information with the variable viewer, you can easily modify and change various properties by clicking on the variable and using the **Data Properties** button. The subsequent "Time History Properties" tabbed dialog box allows you to modify or add specific (Variable) results data properties and also to modify global properties (X-Axis and List).

·   The variable names TIME and FREQ, as well as the reference number 1, are reserved.

·   In interactive mode, the **NUMVAR** command is automatically set to 200 variables; the variable viewer uses the last 10 of these variables for data manipulation, resulting in 190 variables available for the user.

·   All time points of your results file are automatically stored and made available in interactive mode.

·   If your variables are complex values (e.g. amplitude/phase angle), the MIN and MAX values displayed in the lister window will always be the "REAL" values.

6.3.2. Batch

In Interactive Mode (above), your data is automatically stored when you define it. From the command line, this process is accomplished in two separate parts, Defining and Storing.

You define the variable according to the result item in the results file. This means setting up pointers to the result item and creating labels for the areas where this data will be stored. For example, the following commands define time-history variables two, three and four:

```
NSOL,2,358,U,X,UX_at_node_358
ESOL,3,219,47,EPEL,X, Elastic_Strain
ANSOL,4,101,S,X ,Avtg_Stress_101
```

Variable two is a nodal result defined by the **NSOL** command. It is the UX displacement at node 358. Variable three is an element result defined by the **ESOL** command. It is the X component of elastic strain at node 47 for element 219. Variable four is an averaged element nodal result defined by the ANSOL command. It is the X-component of averaged element nodal stress at node 101. Any subsequent reference to these result items will

be through the reference numbers or labels assigned to them. Defining a new variable with the same number as an existing variable overwrites the existing variable. The following commands are used to define variables:

Commands used to define variables			
ANSOL	EDREAD	ESOL	FORCE*
GAPF	LAYERP26*	NSOL	RFORCE
	SHELL*	SOLU	
* Commands that define result location			

The second part is storing the variables (the **STORE** command). Storing means reading the data from the results file into the database. In addition to the **STORE** command, the program stores data automatically when you issue display commands (**PLVAR** and **PRVAR**) or Time-History data operation commands (**ADD**, **QUOT**, etc.).

An example of using the **STORE** command follows:

```
/POST26
NSOL,2,23,U,Y       ! Variable 2 = UY at node 23
SHELL,TOP           ! Specify top of shell results
ESOL,3,20,23,S,X    ! Variable 3 = top SX at node 23 of element 20
PRVAR,2,3           ! Store and then print variables 2 and 3
SHELL,BOT           ! Specify bottom of shell results
ESOL,4,20,23,S,X    ! Variable 4 = bottom SX at node 23 of element 20
STORE               ! By command default, place variable 4 in memory with 2 and 3
PLVAR,2,3,4         ! Plot variables 2,3,4
```

In some situations, you will need to explicitly request storage using the **STORE** command (**Main Menu> TimeHist Postpro> Store Data**). These situations are explained below in the command descriptions. If you use the **STORE** command after issuing the **TIMERANGE** command or **NSTORE** command (the GUI equivalent for both commands is **Main Menu> TimeHist Postpro> Settings> Data**), then the default is **STORE**,NEW. Otherwise, it is **STORE**,MERGE as listed in the command description below. This change in command default is required since the **TIMERANGE** and **NSTORE** commands redefine time (or frequency) points and time increment for data storage. You have the following options for storing data:

MERGE
: Adds newly defined variables to previously stored variables for the time points stored in memory. This is useful if you wish to store data using one specification (**FORCE**, **SHELL**, **LAYERP26** commands) and store data using another specification; see the example above.

NEW
: Replaces previously stored variables, erases previously calculated variables, and stores newly defined variables with current specifications.

APPEND
: Appends data to previously stored variables. That is, if you think of each variable as a column of data, the APPEND option adds rows to each column. This is useful when you want to "concatenate" the same variable from two files, such as in a transient analysis with results on two separate files. Use the **FILE** command (**Main Menu> TimeHist Postpro> Settings> File**) to specify result file names.

ALLOC,N
: Allocates space for N points (N rows) for a subsequent storage operation. Previously stored variables, if any, are zeroed. You normally do not need this option, because the program determines the number of points required automatically from the results file.

Notes:

- By default, batch mode allows you to define up to ten variables. Use the **NUMVAR** command to increase the number of variables up to the available 200.

- Time or Frequency will always be variable 1

- By default, the force (or moment) values represent the *total* forces (sum of the static, damping, and inertial components). The **FORCE** command allows you to work with the individual components.

  Note — The **FORCE** command only affects the output of element nodal forces.

- By default, results data for shell elements and layered elements are assumed to be at the top surface of the shell or layer. The **SHELL** command allows you to specify the top, middle or bottom surface. For layered elements, use the **LAYERP26** and **SHELL** commands to indicate layer number and surface location, respectively.

- Other commands useful when defining variables are:

  – **NSTORE** - defines the number of time points or frequency points to be stored.

  – **TIMERANGE** - defines the time or frequency range in which data are to be stored.

  – **TVAR** - changes the meaning of variable 1 from time to cumulative iteration number.

  – **VARNAM** - assigns a *name* (32 character max.) to a variable.

  – **RESET** - removes all variables and resets all specifications to initial defaults.

6.4. Processing Your Variables to Develop Calculated Data

Often, the specific analysis data you obtain in your results file can be processed to yield additional variable sets that provide valuable information. For example, by defining a displacement variable in a transient analysis, you can calculate the velocity and acceleration by taking derivatives with respect to time. Doing so will yield an entirely new variable that you may wish to analyze in conjunction with your other analysis data.

6.4.1. Interactive

The variable viewer provides an intuitive calculator interface for performing calculations. All of the command capability can be accessed from the calculator area. The calculator can be displayed or hidden by clicking on the bar above the calculator area.

Follow these steps to process your time history data using the variable viewer:

1. Specify a variable in the variable name input area. This must be a unique name, otherwise you will be prompted to overwrite the existing variable of that name.

2. Define the desired variable expression by clicking on the appropriate keys, or selecting time-history variables or APDL parameters from the drop down lists.

   Result: The appropriate operators, APDL parameters or other variable names appear in the Expression Input Area.

3. Click the "Enter" button on the calculator portion of the Variable viewer

   Result: The data is calculated and the resultant variable name appears in the variable list area. The expression will be available in the variable viewer for the variable name until the variable viewer is closed.

Notes:

- To find the derivative of a variable "UYBLOCK" with respect to another variable

VBLOCK = deriv ({UYBLOCK} , {TIME})

- To find the amplitude of a complex time-history variable "PRESMID"

AMPL_MID = abs ({PRESMID})

OR,

AMPL_MID = sqrt (real ({PRESMID}) ^2 + imag ({PRESMID}) ^2)

- To find the phase angle of a complex time-history variable "UYFANTIP"

PHAS_TIP = atan ({UYFANTIP}) * 180/pi

Where pi = acos (-1)

- To multiply a complex time-history variable "PRESMID" with a factor (2 + 3i)

SCAL_MID = cmplx (2,3)* {PRESMID}

- To fill a variable with ramped data use the following equation

RAMP_.25BY_0.5 = .25 + (.05 * ({nset} - 1))

- To fill a variable as a function of time use the following equation

FUNC_TIME_1 = 10 * ({TIME} - .25)

- To find the relative acceleration response PSD for a variable named UZ_4, use the following equation

RPSD_4 = RPSD({UZ_4},{UZ_4},3,2)

6.4.2. Batch

In batch mode, you use combinations of commands. Some identify the variable and the format for the output, while others identify the variable data to be used to create the new variable. The calculator operations themselves are performed by specific commands.

- To find the derivative of a variable "UYBLOCK " with respect to another variable "TIME"

```
NSOL,2,100,u,y,UYBLOCK   !Variable 2 is UY of node 100
DERIV, 3,2,1,,VYBLOCK    !Variable 3 is named VYBLOCK It is the
                         !derivative of variable 2 with respect
                         !to variable 1 (time)
```

- To find the amplitude of a complex time-history variable PRESMID

```
NSOL,2,123,PRES,,PRESMID  !Variable 2 is the pressure at node 123
ABS, 3,2,,,AMPL_MID       !Absolute value of a complex variable
                          !is its amplitude.
```

- To find the phase angle (in degrees of a complex time-history variable "UYFANTIP"

```
Pi = acos(-1)
ATAN,4,2,,,PHAS_MID,,,180/pi !ATAN function of a complex
                             !variable (a + ib) gives atan (b/a)
```

- To multiply a complex POST 26 variable "PRESMID" with a factor (2+3i):

```
CFACT,2,3         !Scale factor of 2+3i
ADD,5,2,,,SCAL_MID   !Use ADD command to store variable 2 into
                     !variable 5 with the scale factor of (2+3i)
```

- To fill a variable with ramped data

```
FILLDATA,6,,,,.25,.05,ramp_func  !Fill a variable with
                                 !ramp function data.
```

The following commands are used to process your variables, develop computed relationships and store the data. See the specific command reference for more information on processing your time-history variables.

Variable processing commands		
ABS	IMAGIN	SMALL
ADD	INT1	SQRT
ATAN	LARGE	RPSD
CLOG	NLOG	CVAR
CONJUG	PROD	RESP
DERIV	QUOT	
EXP	REALVAR	

6.5. Importing Data

This feature allows the user to read in set(s) of data from a file into Time history variable(s). This enables the user, for instance, to display and compare test results data against the corresponding ANSYS results data.

6.5.1. Interactive

The "Import Data" button in the variable viewer leads the user through the interactive data import process. Clicking on "Import Data" allows the user to browse and select the appropriate file. The data must be in the format below:

```
# TEST DATA FILE EXAMPLE
# ALL COMMENT LINES BEGIN WITH #
# Blank lines are ignored
#
# The first line without # sign must contain the variable names to be used
# for each column of data read into POST26. NOTE that for complex data only
# one variable name should be supplied per (real, imaginary) pair as shown below.
# The next line can either be left blank or have descriptors for each column
# such as REAL and IMAGINARY
#
# The data itself can be in free format with the columns "comma delimited",
# "tab delimited", or "blank delimited"
#
# The first column of data is always reserved for the independent variable
 (usually TIME or FREQUENCY)
#
   FREQ              TEST1                        TEST2
              REAL        IMAGINARY        REAL        IMAGINARY
1.00000E-02    -128.32      0.17764       5.6480      -4.47762E-03
2.00000E-02    -150.08      0.36474       5.6712      -8.99666E-03
3.00000E-02    -163.12      0.57210       5.7097      -1.35897E-02
4.00000E-02    -147.63      0.81364       5.7629      -1.82673E-02
5.00000E-02    -133.90      1.1091        5.8298      -2.29925E-02
6.00000E-02    -172.38      1.4886        5.9080      -2.76290E-02
```

The user has two choices, depending upon the data in the file.

- - Graph overlay information: This can be used when you are interested in simply overlaying the experimental or theoretical results on top of the Finite Element Analysis results in the same plot. The data set(s)

brought in using this method will show up in the "overlay data" drop down list. A data set selected in this drop down list will overlay the current variable graph display. You will need to choose "None" to not overlay the data. The sets of data brought in using this method can be overlaid on a variable graph, allowing a visual comparison of the test data against the finite element result.

- - Linear interpolation into variables: If you want to compare Finite Element Analysis results with your experimental or theoretical results at the same time points, you should use the Interpolate to FEA Time Points option. This option linearly interpolates the test data to calculate test results at the ANSYS time/frequency points. The interpolated data is then stored as a time-history variable(s) and is added to the list of variables in the variable viewer. These variables can then be displayed, listed, or operated on as any other time-history variable. You must ensure that linear interpolation is valid for the data imported. In addition, the non-interpolated "raw" data from the file is available in the "overlay data" drop down list, as explained above.

6.5.2. Batch Mode

You import data from a file into a time history variable using one of the following methods:

- Use the **DATA** command to read in a pre-formatted file. The file should be in Fortran format as described in the **DATA** command.

- Read the data from a free format, "comma," "blank," or "tab" delimited file. You can store it as a time history variable using the two step procedure below:

  1. Read the file into a table array using the ***TREAD** command. This step requires that you know the number of data points in the file since you will need to prespecify the table array size (***DIM**).

  2. Use the **VPUT** command to store the array into a time history variable. You can store one array at a time into a time history variable

- The following two 'external' commands are available to facilitate easy import of data into time-history variables.

  1. ~eui, 'ansys::results::timeHist::TREAD directorypath/filename arrayname'

     The above command will determine the size of the data file, create a table array of name 'arrayname', appropriately dimension it based on the number of data sets in the file, and then read the data into this array. This command must be issued prior to the command shown below.

  2. ~eui,'ansys::results::timeHist::vputData arrayname variablenumber'

     The above command assumes that you have already created a table array 'arrayname' as described in 1) above. This command will put the data stored in the 'arrayname' table into the time history variables starting with variable id 'variable number'.

     For Example:

     ```
     ~eui,'ansys::results::timeHist::TREAD  d:\test1\harmonic.prn TESTMID'
     ~eui,'ansys::results::timeHist::vputData TESTMID 5'
     ```

     The first command above will read data set(s) from the file harmonic.prn in the directory d:\test1 and store this data in to the table array 'TESTMID'. The next command will then import the data from TESTMID array into ANSYS time history variable starting from variable number 5. If multiple data sets are in 'harmonic.prn' then the first data set will be stored in variable 5, the next data set in variable 6 and so on. If these variables have already been defined they will be overwritten.

6.6. Exporting Data

This feature allows the user to write out selected time history variable(s) to an ASCII file or to APDL array/table parameter. This enables you to perform several functions such as pass data on to another program for further processing or to archive data in an easily retrievable format.

6.6.1. Interactive Mode

The "Export Data" button is used to export the currently selected variables from the variable viewer's listing window to a file. Clicking on this button provides user with a choice of three export options:

- Export to file:

  Use this option to export the selected time history variables to an ASCII file, which then can be used by other programs for further processing. The format of this file is identical to the one discussed in Importing Data above. The data in the file can be in one of two formats: Comma separated (file extension csv), or Space delimited (file extension prn). The number of items that can be exported at one time is limited to four variables (if complex) plus time variable or nine variables (if real) plus time variable. The variable names from the variable viewer's list window are used in the column header information line.

- Export to APDL table:

  This option will store the time history variable data into the table name specified by the user. This option allows the user to operate on time history data with the extensive APDL capabilities available in ANSYS (such as ***VFUN**, ***VOPER**, etc.). The index of the table (0th column) is always the independent variable (usually Time or Frequency). If multiple time history variables are exported they will be stored in consecutive columns starting with column 1. If the variables contain complex number data, 2 columns are used per variable, one column of real and one for imaginary data.

  NOTE: When multiple variables are selected in the variable viewer for export, they are stored in the order in which they are displayed in the variable viewer lister box at that time (top to bottom). It is the user's responsibility to note down this order.

- Export to APDL array:

  This option will store the time history variable data into an array parameter specified by the user. This option allows user to operate on the time history data using the extensive APDL capabilities of ANSYS. The first column of the array is reserved for the independent variable (usually time or frequency). The time history variables are stored starting in column 2 in the order in which they are shown in the variable viewer's list window.

6.6.2. Batch Mode

Exporting data from a time-history variable into a file is a two step process:

1. Export a time-history variable data to an array parameter. The command **VGET** allows you to export a single time-history variable into a properly dimensioned (***DIM**) array parameter. The size of this array can be determined via ***GET** ,size,vari,,nsets.

2. Once the array is filled then the data can be written out to a file via ***VWRITE** command as shown below.

Example:

```
NSOL,5,55,U,X
STORE,MERGE            ! Store UX at node 55
*GET,size,VARI,,NSETS
```

```
*dim,UX55,array,size    ! Create array parameter
VGET,UX55(1),5          ! Store time history data of variable 5 into ux55
*CFOPEN,disp,dat
*VWRITE,UX55(1)         ! Write array in given format to file "disp.dat"
(6x,f12.6)
*CFCLOSE
```

6.7. Reviewing the Variables

Once the variables are defined, you can review them via graph plots or tabular listings.

6.7.1. Plotting Result Graphs

The description for graph plotting, both with the variable viewer and from the command line follows:

6.7.1.1. Interactive

The "Graph Data" button in the variable viewer allows you to plot all the selected variables. A maximum of 10 variables can be plotted on a single graph. By default, the variable used for the X-axis of the graphs is TIME for static and transient analyses or FREQUENCY for harmonic analysis. You can select a different variable for the X-axis of the graph using the radio button under the column X-AXIS in the list of variables.

When plotting complex data such as from a harmonic analysis, use the 'results to view' drop-down list on the right top corner of the variable viewer to indicate whether to plot Amplitude (default), Phase angle, Real part or Imaginary part.

The variable viewer stores all the time points available on the results file. You can display a portion of this data by selecting a range for the X-axis value. This is useful when you want to focus on the response around a certain time point e.g., around the moment of impact in a drop test analysis. This is available in the "Data Properties" dialog under the X-AXIS tab. Note that this is a global setting i.e. this setting is used for all subsequent graph plots.

6.7.1.2. Batch

The **PLVAR** command (**Main Menu> TimeHist Postpro> Graph variables**) graphs up to 10 variables at a time on a graph. By default, the variable used for the X-axis of the graphs is TIME for static and transient analyses or FREQUENCY for harmonic analysis. You can specify a different variable for the X-axis (e.g. deflection or strain) by using the **XVAR** command (**Main Menu> TimeHist Postpro> Settings> Graph**).

When plotting complex data such as from a harmonic analysis, **PLVAR** plots amplitude by default. You can switch to plotting Phase angle or Real part or Imaginary part via the **PLCPLX** command (**Main menu> TimeHist Postpro> Settings> Graph**).

You can display a portion of the stored data by selecting a range for the X-axis values via the **/XRANGE** command.

Two sample plots are shown below:

Figure 6.1 Time-History Plot Using XVAR = 1 (time)
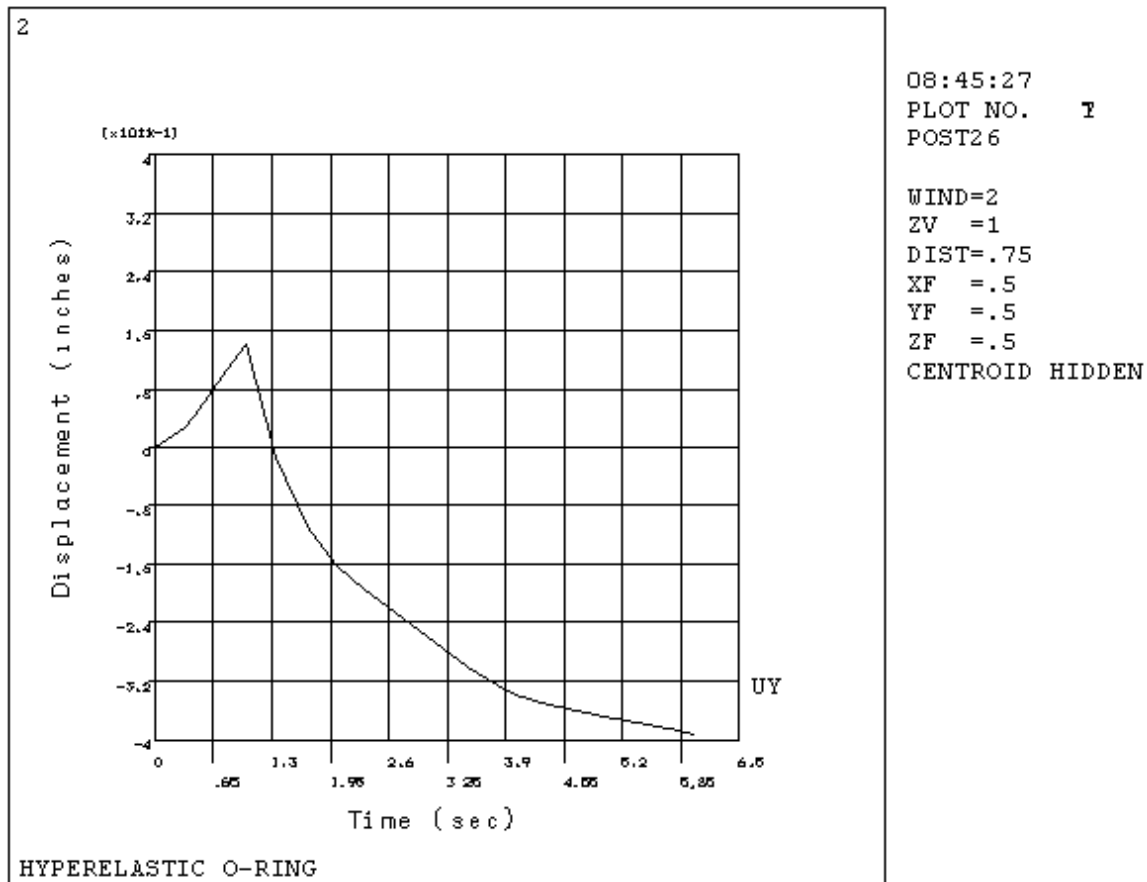
Figure 6.2 Time-History Plot Using XVAR $\neq$ 1
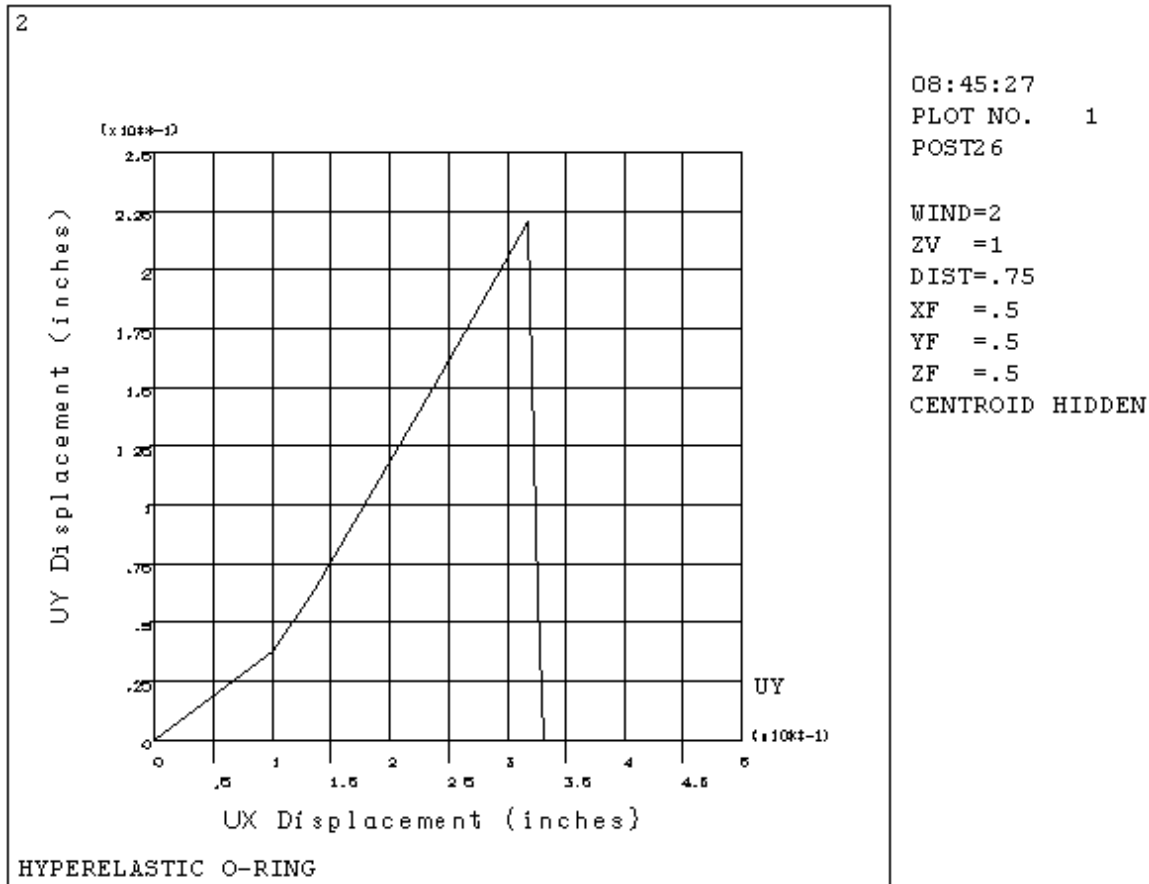
For more information on adjusting the look and feel of your graph plots, see Chapter 13, "Creating Graphs" later on in this manual.

6.7.2. Listing Your Results in Tabular Form

To create tabular data lists, both interactively and from the command line, use the following procedures.

6.7.2.1. Interactive

The "List Data" button of the variable viewer can be used to list up to six variables in the variable viewer.

When listing complex data such as from a harmonic analysis, use the 'results to view' drop-down list on the right top corner of the variable viewer to indicate whether to printout "amplitude and phase angle" or "real and imaginary parts" in the listing. Select amplitude or phase to list "Amplitude and Phase Angle" results. Select real or imaginary to list "Real and Imaginary" results.

You can restrict data being listed to a range of time or frequency. This and other listing controls are available through the "Lists" tab under Data Properties dialog. In addition to setting the range of time or frequency, this dialog also allows you to:

- Control the number of lines before repeating headers on the listings.
- Additionally print the extreme values of the selected. variables.
- Specify printing every 'n'th data point.

6.7.2.2. Batch

You can use the **PRVAR** command (**Main Menu> TimeHist Postpro> List Variables**) to list up to six variables in tabular form. This is useful if you want to find the value of a result item at a specific time or frequency. You can control the times (or frequencies) for which variables are to be printed. To do so, use one of the following:

> **Command(s): NPRINT**, **PRTIME**
> **GUI: Main Menu> TimeHist Postpro> Settings> List**

You can adjust the format of your listing somewhat with the **LINES** command (**Main Menu> TimeHist Postpro> Settings> List**). A sample **PRVAR** output is shown below.

Sample Output from PRVAR

```
         ***** ANSYS time-history VARIABLE LISTING *****

  TIME           51 UX        30 UY
                 UX           UY
 .10000E-09    .000000E+00   .000000E+00
 .32000        .106832       .371753E-01
 .42667        .146785       .620728E-01
 .74667        .263833       .144850
 .87333        .310339       .178505
 1.0000        .356938       .212601
 1.3493        .352122       .473230E-01
 1.6847        .349681      -.608717E-01
```

When a complex variable consists of real and imaginary parts, the **PRVAR** command lists both the real and imaginary parts by default. You can work with real and imaginary, or amplitude and phase angle using the **PRCPLX** command.

Another useful listing command is **EXTREM** (**Main Menu> TimeHist Postpro> List Extremes**), which prints the maximum and minimum Y-variable values within the active X and Y ranges. You can also assign these extreme values to parameters using the ***GET** command (**Utility Menu> Parameters> Get Scalar Data**). A sample **EXTREM** output is shown below.

Sample Output from EXTREM

```
           time-history SUMMARY OF VARIABLE EXTREME VALUES
   VARI TYPE     IDENTIFIERS    NAME     MINIMUM    AT TIME    MAXIMUM   AT TIME

     1 TIME    1 TIME      TIME     .1000E-09  .1000E-09  6.000     6.000
     2 NSOL   50 UX        UX       .0000E+00  .1000E-09  .4170     6.000
     3 NSOL   30 UY        UY      -.3930      6.000      .2146     1.000
```

6.8. Additional Time-History Postprocessing

6.8.1. Random vibration (PSD) Results Postprocessing

Covariance and response PSD are of interest when postprocessing random vibration analysis results. The calculations use Jobname.rst and Jobname.psd files from a random vibration analysis.

6.8.1.1. Interactive

To choose whether to compute the covariance or the response PSD when reviewing the results of a random vibration analysis, follow these steps:

1. Launch the variable viewer by selecting **Main Menu> TimeHist Postpro**. If the variable viewer is already open, click the **Clear Time-History Data** button, located in the toolbar. The Spectrum Usage dialog box appears. (See Figure 6.3: "Spectrum Usage Dialog Box".)

2. Select "Find the covariance of quantities" or "Create response power spectral density (PSD)."

    Note — You can improve the "smoothness" of the response PSD curves by specifying the number of points on either side of a natural frequency point (**STORE**,PSD) with the slider, shown in Figure 6.3: "Spectrum Usage Dialog Box".

3. Click **OK**.

Figure 6.3 Spectrum Usage Dialog Box
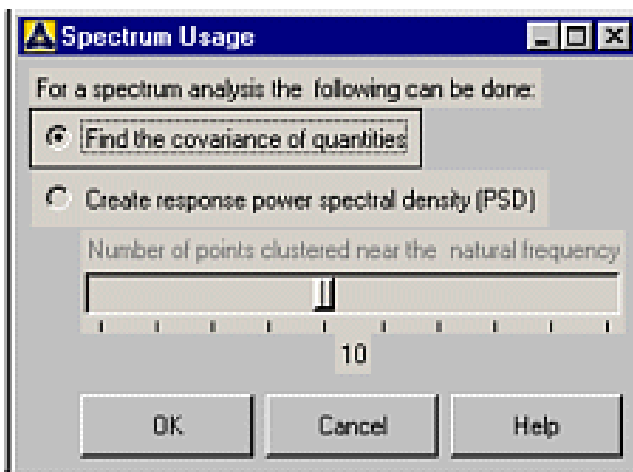
6.8.1.1.1. Covariance

Follow these steps to calculate covariance using the variable viewer:

1. Select "Find the covariance of quantities" from the Spectrum Usage dialog box and click **OK**.

    Note — If you have performed RPSD calculations, click the **Clear Time-History Data** button to load the Spectrum Usage dialog box.

2. Using the Variable Viewer, define the variables between which covariance is to be calculated.

3. Specify a variable in the variable name input area of the variable viewer. The name must be unique or you will be asked to overwrite the existing variable.

4. Click the **CVAR** button in the calculator area of the variable viewer. The following dialog box appears.
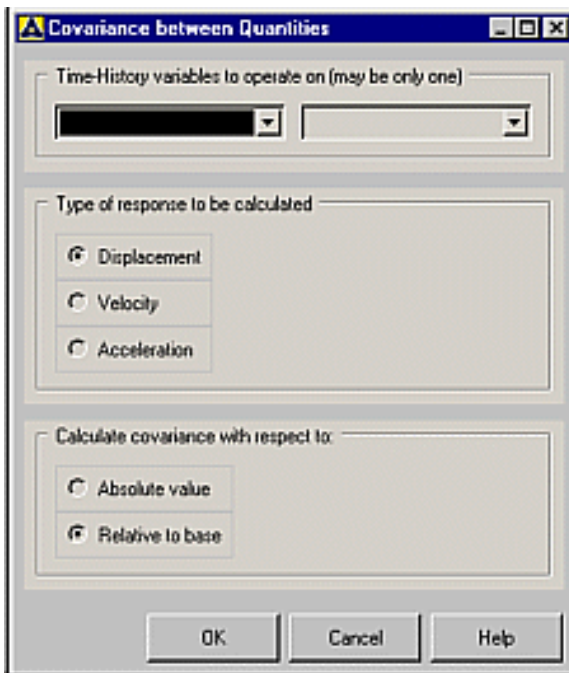
5. Select the variables to operate on from one or both of the pull down lists (corresponds to the IA, IB argument for the **CVAR** command).

6. Select the type of response to be calculated (corresponds to the $ITYPE$ argument for the **CVAR** command).

7. Choose whether to calculate the covariance with respect to the absolute value or relative to the base (corresponds to the $DATUM$ argument for the **CVAR** command).

8. Click OK to save your preferences and close the dialog box. The function cvar(IA,IB,ITYPE,DATUM) will be displayed in the expression area of the calculator.

9. Click **Enter** in the calculator portion of the variable viewer to start the evaluation.

When the evaluation is finished, the covariance value is stored; the variable name is displayed in the variable list area for further postprocessing.

6.8.1.1.2. Response PSD

Follow these steps to calculate the Response PSD using the variable viewer.

1. Select "Create response power spectral density (PSD)" from the Spectrum Usage dialog box and click **OK**.

2. Using the Variable Viewer, define the variables for which the Response PSD is to be calculated.

3. Specify a variable in the variable name input area of the variable viewer. The name must be unique or you will be asked to overwrite the existing variable.

4. Click the **RPSD** button in the calculator area of the variable viewer. The following dialog box appears.
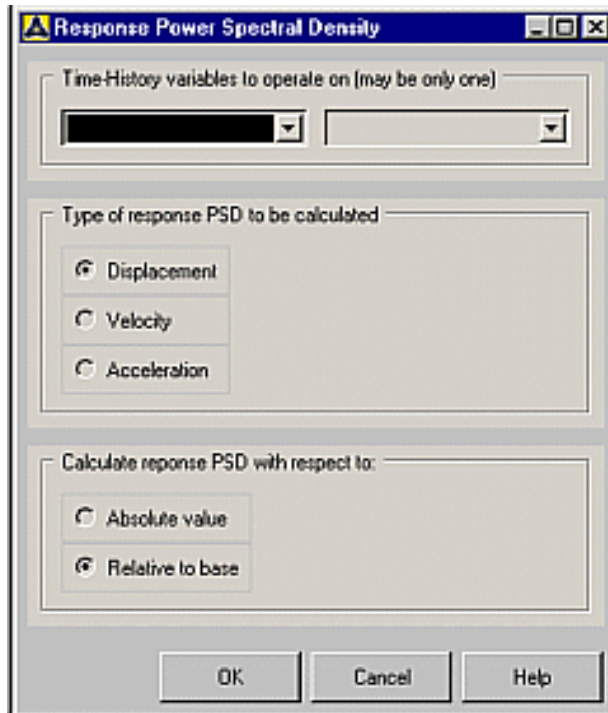
5.   Select the variables to be operated on (corresponds to the IA,IB argument for the **RPSD** command).

6.   Select the type of PSD to be calculated (corresponds to the $ITYPE$ argument for the **RPSD** command).

7.   Choose whether to calculate the response PSD with respect to the absolute value or relative to the base (corresponds to the $DATUM$ argument for the **RPSD** command).

8.   Click **OK** to save your preferences and close the dialog box. The function rpsd(IA,IB,ITYPE,DATUM) will be displayed in the expression area of the calculator.

9.   Click **Enter** in the calculator portion of the variable viewer to start the evaluation.

When the evaluation is finished, the response PSD value is stored; the variable name is displayed in the variable list area for further postprocessing.

6.8.1.2. Batch

Response PSDs and covariance values can be calculated for any results quantity using **Jobname.RST** and **Jobname.PSD** from a random vibration analysis. The procedure for performing this calculation is described in detail in Calculating Response PSDs in POST26 in the *ANSYS Structural Analysis Guide*.

6.8.2. Generating a Response Spectrum

This feature allows you to generate a displacement, velocity, or acceleration response spectrum from a given displacement time-history. The response spectrum can then be specified in a spectrum analysis to calculate the overall response of a structure.

6.8.2.1. Interactive

Generating a response spectrum requires two previously-defined variables: one containing frequency values for the response spectrum (corresponding to the $LFTAB$ argument for the **RESP** command), and the other containing the displacement time-history (corresponding to the $LDTAB$ argument for the **RESP** command). The frequency values represent the abscissa of the response spectrum curve and the frequencies of the one-degree-of-freedom

oscillators used to generate the response spectrum. You can create the frequency variable by using either the calculator portion of the variable viewer to define an equation or the variable viewer's import options.

> *Note* — The displacement time-history values usually result from a transient dynamic analysis. You can also create the displacement variable using the import options (if the displacement time-history is on a file) or add displacement as a variable.
>
> You must have a time variable defined as the first variable in the variable list (variable 1).

Once you have defined the frequency and displacement time history variables, follow these steps to calculate a response spectrum using the variable viewer.

1. Specify a variable name in the variable name input area. The name must be unique or you will be asked to overwrite the existing variable.

2. Click the **RESP** button in the calculator portion of the variable viewer. The following dialog box appears.
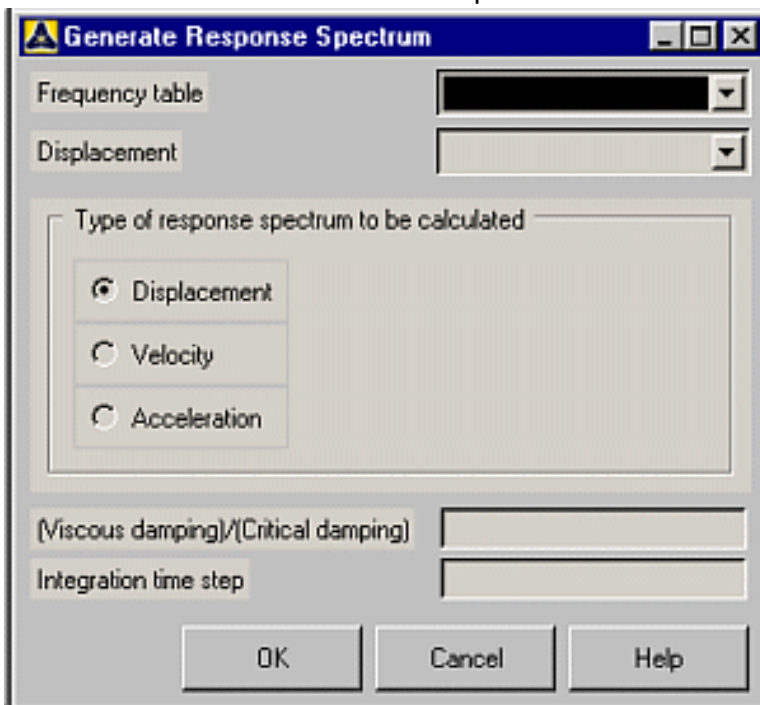
3. Select the reference number of the variable containing the frequency table from the pull down list (corresponds to the *LFTAB* argument for the **RESP** command).

4. Select the reference number of the variable containing the displacement time-history from the pull down list (corresponds to the *LDTAB* argument for the **RESP** command).

5. Select the type of response spectrum to be calculated (corresponds to the *ITYPE* argument for the **RESP** command).

6. Enter the ratio of viscous damping to critical damping as a decimal (corresponds to the *RATIO* argument for the **RESP** command).

7. Enter the integration time step (corresponds to the *DTIME* argument for the **RESP** command).

8. Click **OK** to save your preferences and close the dialog box. The function resp(LFTAB,LDTAB,ITYPE,RA-TIO,DTIME) is displayed in the expression area of the calculator.

9. Click **Enter** in the calculator portion of the variable viewer to start the evaluation.

When the evaluation is finished, the response spectrum is stored; the variable name is displayed in the variable list area for further postprocessing.

6.8.2.2. Batch

The **RESP** command in time-history is used to generate the response spectrum, use either of the following:

> **Command(s): RESP**
> **GUI: Main Menu> TimeHist Postpro> Generate Spectrm**

RESP requires two previously defined variables: one containing frequency values for the response spectrum (field $LFTAB$) and the other containing the displacement time-history (field $LDTAB$). The frequency values in $LFTAB$ represent not only the abscissa of the response spectrum curve, but also the frequencies of the one-degree-of-freedom oscillators used to generate the response spectrum. You can create the $LFTAB$ variable using either the **FILLDATA** command or the **DATA** command.

The displacement time-history values in $LDTAB$ usually result from a transient dynamic analysis of a single-DOF system. You can create the $LDTAB$ variable using the **DATA** command (if the displacement time-history is on a file) or the **NSOL** command (**Main Menu> TimeHist Postpro> Define Variables**). A numerical time-integration scheme is used to calculate the response spectrum.

6.8.3. Data Smoothing

If you are working with noisy results data such as from an explicit dynamic analysis, you may want to "smooth" the response. This may allow for better understanding / visualization of the response by smoothing out local fluctuations while preserving the global characteristics of the response. The time-history "smooth" operation allows fitting a 'n'th order polynomial to the actual response.

This operation can be used only on static or transient results i.e., complex data cannot be fitted.

6.8.3.1. Interactive

This capability is available in the variable viewer's calculator through a function smooth (x1,x2,n) where x2 is the dependent time history variable (such as TIME), and x1 is the independent time history variable (such as response at a point), and 'n' is the order of fit. This function is available only by typing in the expression portion of the calculator.

For example to evaluate a second order fit for the UY response at the midpoint of a structure: (smooth variable x1 with respect to variable x2 of order "n"):

```
Smoothed_response = SMOOTH ({UY_AT_MIDPOINT},{TIME},2)
```

6.8.3.2. Batch

If you're working with noisy results data, you may want to "smooth" that data to a smoother representative curve.

Four arrays are required for smoothing data. The first two contain the noisy data from the independent and the dependent variables, respectively; the second two will contain the smoothed data (after smoothing takes place) from the independent and dependent variables, respectively. You must always create the first two vectors (***DIM**) and fill these vectors with the noisy data (**VGET**) before smoothing the data. If you are working in interactive mode, ANSYS automatically creates the third and fourth vector, but if you are working in batch mode, you must also create these vectors (***DIM**) before smoothing the data (ANSYS will fill these with the smoothed data).

Once these arrays have been created, you can smooth the data using the **SMOOTH** command (**Main Menu> TimeHist Postpro> Smooth Data**). You can choose to smooth all or some of the data points using the $DATAP$ field, and you can choose how high the fitting order for the smoothed curve is to be using the $FITPT$ field. $DATAP$ defaults to all points, and $FITPT$ defaults to one-half of the data points. To plot the results, you can choose to plot unsmoothed, smoothed, or both sets of data.

Chapter 7: Selecting and Components

7.1. What Is Selecting?

If you have a large model, it is helpful to work with just a portion of the model data to apply loads, to speed up graphics displays, to review results selectively, and so on. Because all ANSYS data are in a database, you can conveniently choose subsets of the data by using selecting.

Selecting enables you to select subsets of nodes, elements, keypoints, lines, etc. so that you can work with just a handful of entities. The ANSYS program uses a database to store all the data that you define during an analysis. This database design allows you to select only a portion of the data without destroying other data.

Typically, you perform selecting when you specify loads. By selecting nodes on a surface, for example, you can conveniently apply a pressure on *all* nodes in the subset instead of applying it to each individual node.

Another useful feature of selecting is that you can select a subset of entities and name that subset. For example, you can select all elements that make up the fin portion of a heat exchanger model and call the resulting subset FIN. Such named subsets are called *components*. You can even group several components into an *assembly*.
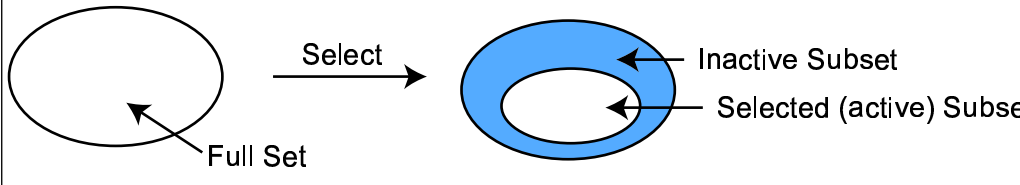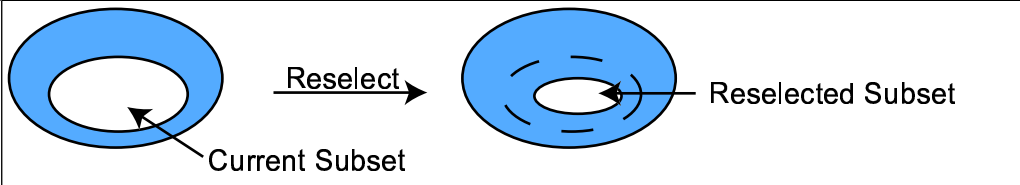
7.2. Selecting Entities
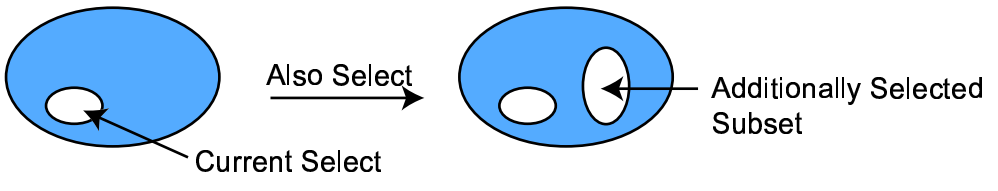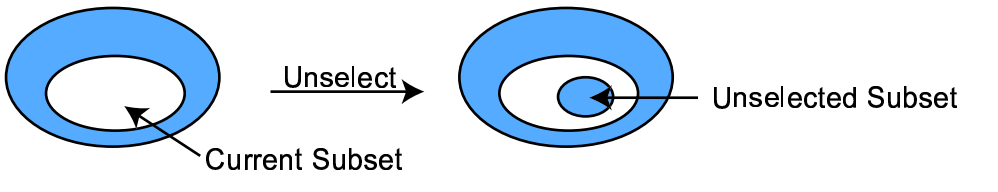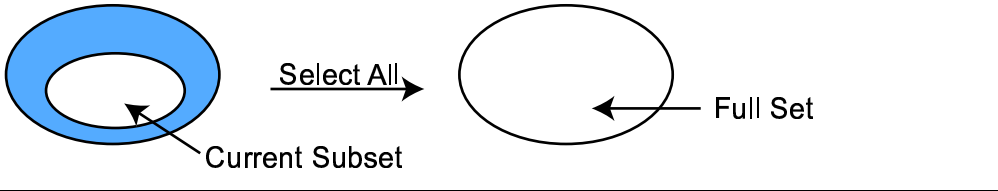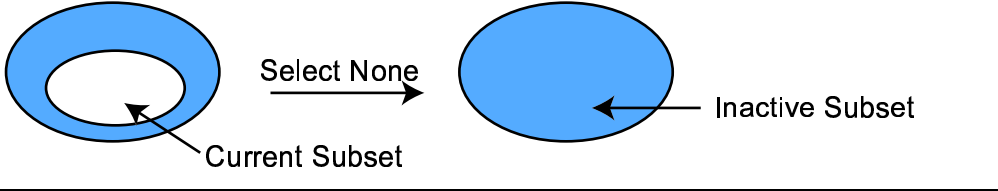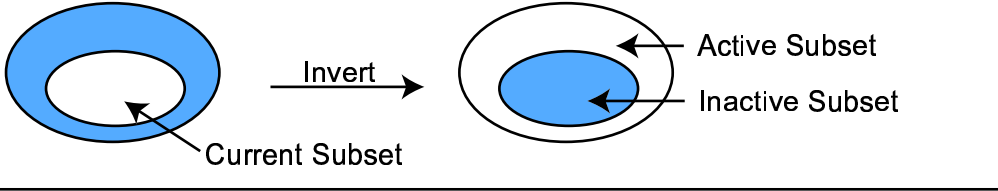
You can select a subset of entities using a combination of seven basic select functions:

- Select
- Reselect
- Also Select
- Unselect
- Select All
- Select None
- Invert

These functions are illustrated and described in the following table.

Table 7.1 Selection Functions

Also Select Adds a different subset to the current subset.	Also Select → Current Select / Additionally Selected Subset
Unselect Subtracts a portion of the current subset.	Unselect → Current Subset / Unselected Subset
Select All Restores the full set.	Select All → Current Subset / Full Set
Select None Deactivates the full set (opposite of Select All).	Select None → Current Subset / Inactive Subset
Invert Switches between the active and inactive portions of the set.	Invert → Current Subset / Active Subset / Inactive Subset

These functions are available for all entities (nodes, elements, keypoints, lines, areas, and volumes) in the **Utility Menu** of the Graphical User Interface as well as by command.

For additional information on picking, see Chapter 5, "Graphical Picking" in the *ANSYS Operations Guide*.

7.2.1. Selecting Entities Using Commands

Table 7.2: "Select Commands" shows a summary of commands available to select subsets of entities. Notice the "crossover" commands: commands that allow you to select one entity based on another entity. For example, you can select all keypoints attached to the current subset of lines. Here is a typical sequence of select commands:

```
LSEL,S,LOC,Y,2,6     ! Select lines that have center locations between Y=2 and Y=6
LSEL,A,LOC,Y,9,10    ! Add lines with center locations between Y=9 and Y=10
NSLL,S,1             ! Select all nodes on the selected lines
ESLN                 ! Select all elements attached to selected nodes
```

See the **LSEL**, **NSLL**, and **ESLN** command descriptions in the *ANSYS Commands Reference* for further information.

Note — Crossover commands for selecting finite element model entities (nodes or elements) from solid model entities (keypoints, areas, etc.) are valid only if the finite element entities were generated by a meshing operation on a solid model that contains the associated solid-model entities.

Table 7.2 Select Commands

Entity	Basic Commands	Crossover Command(s)
Nodes	**NSEL**	**NSLE**, **NSLK**, **NSLL**, **NSLA**, **NSLV**
Elements	**ESEL**	**ESLN**, **ESLL**, **ESLA**, **ESLV**
Keypoints	**KSEL**	**KSLN**, **KSLL**
Hard Points	**KSEL**, **ASEL**, **LSEL**	None
Lines	**LSEL**	**LSLA**, **LSLK**
Areas	**ASEL**	**ASLL**, **ASLV**
Volumes	**VSEL**	**VSLA**
Components	**CMSEL**	None

7.2.2. Selecting Entities Using the GUI

The GUI path equivalent to issuing most of the commands listed in Table 7.2: "Select Commands" is **Utility Menu> Select> Entities**. This GUI choice displays the **Select Entities** dialog box, from which you choose, among other things, the type of entities you want to select and the criteria by which you will select them. For example, you can choose **Elements** and **By Num/Pick** to select elements by number or by picking. You can press the **Help** button from within the **Select Entities** dialog box for detailed information about selecting via the GUI; the help is context-sensitive and will reflect any choices you've made in the **Select Entities** dialog box.

Often, you will want to combine entities into components or assemblies for clarity or ease of reference. The following GUI paths provide selection options for defined components or assemblies:

GUI:

> **Utility Menu> Select> Comp/Assembly> Select All**
> **Utility Menu> Select> Comp/Assembly> Select Comp/Assembly**
> **Utility Menu> Select> Comp/Assembly> Pick Comp/Assembly**
> **Utility Menu> Select> Comp/Assembly> Select None**

7.2.3. Selecting Lines to Repair CAD Geometry

When CAD geometry is imported into ANSYS, the transfer may define the display of short line elements, which are difficult to identify on screen.

By choosing the line selection option, you can find and display these short lines:

> **Command(s): LSEL**
> **GUI: Utility Menu> Select> Entities> Lines> By Length/Radius**

Enter the minimum and maximum length or radius in the $VMIN$ and $VMAX$ fields. These fields, as they are used in this option, represent the range of values which corresponds to the length or radius of the short line elements. You should enter reasonable values in $VMIN$ and $VMAX$ to assure that the selected set only includes those short lines that you want to display. When the selected set appears on screen, you can pick individual lines within the set and repair the geometry as necessary.

> *Note* — A line which is not an arc returns a zero radius. RADIUS is only valid for lines that are circular arcs.

7.2.4. Other Commands for Selecting

To restore all entities to their full sets, use one of the following:

> **Command(s): ALLSEL**
> **GUI: Utility Menu> Select> Everything Below> Selected Areas**
> **Utility Menu> Select> Everything Below> Selected Elements**
> **Utility Menu> Select> Everything Below> Selected Lines**
> **Utility Menu> Select> Everything Below> Selected Keypoints**
> **Utility Menu> Select> Everything Below> Selected Volumes**

This one command has the same effect as issuing a series of **NSEL**,ALL; **ESEL**,ALL; **KSEL**,ALL; etc. commands.

You also can use **ALLSEL** or its GUI equivalents to select a set of related entities in a hierarchical fashion. For example, given a subset of areas, you can select all lines defining those areas, all keypoints defining those lines, all elements belonging to these areas, lines, and keypoints, and all nodes belonging to these elements, by simply issuing one command: **ALLSEL**,BELOW,AREA

To select a subset of degree of freedom and force labels, use one of the following:

> **Command(s): DOFSEL**
> **GUI: Main Menu> Preprocessor> Loads> Define Loads> Operate> Scale FE Loads> Constraints**
> **Main Menu> Preprocessor> Loads> Define Loads> Operate> Scale FE Loads> Forces**
> **Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs Add> Constraints**
> **Main Menu> Preprocessor> Loads> Define Loads> Settings> Replace vs Add> Forces**
> **Main Menu> Solution> Define Loads> Operate> Scale FE Loads> Constraints**
> **Main Menu> Solution> Define Loads> Operate> Scale FE Loads> Forces**
> **Main Menu> Solution> Define Loads> Settings> Replace vs Add> Constraints**
> **Main Menu> Solution> Define Loads> Settings> Replace vs Add> Forces**

By selecting a subset of these labels, you can simply use ALL in the *Label* field of some commands to refer to the entire subset. For instance, the command **DOFSEL**,S,UX,UZ followed by the command **D**,ALL,ALL would put UX and UZ constraints on all selected nodes. **DOFSEL** does not affect the solution degrees of freedom.

7.3. Selecting for Meaningful Postprocessing

Selecting can also help you during postprocessing. For instance, in POST1, you can select just a portion of your model to display or list the results. You *should always* use selecting to obtain meaningful results in POST1 when the model has discontinuities.

When you request contour displays with the **PLNSOL** command (**Utility Menu> Plot> Results> Contour Plot> Nodal Solution**), the ANSYS program produces smooth, continuous contours by averaging the data at nodes. This averaging is acceptable as long as the model contains no discontinuities, such as:

- Two different materials modeled adjacent to each other or a model with different thicknesses. (See Figure 7.1: "Shell Model with Different Thicknesses")

- Adjacent layered shells having a different number of layers. (See Figure 7.2: "Layered Shell (SHELL91 or SHELL99) with Nodes Located at Midplane" and Figure 7.3: "Layered Shell (SHELL91 or SHELL99) with Nodes Located at Bottom Surface")

When such discontinuities are present, be careful to process each side of the discontinuity separately by using selecting.

Figure 7.1 Shell Model with Different Thicknesses
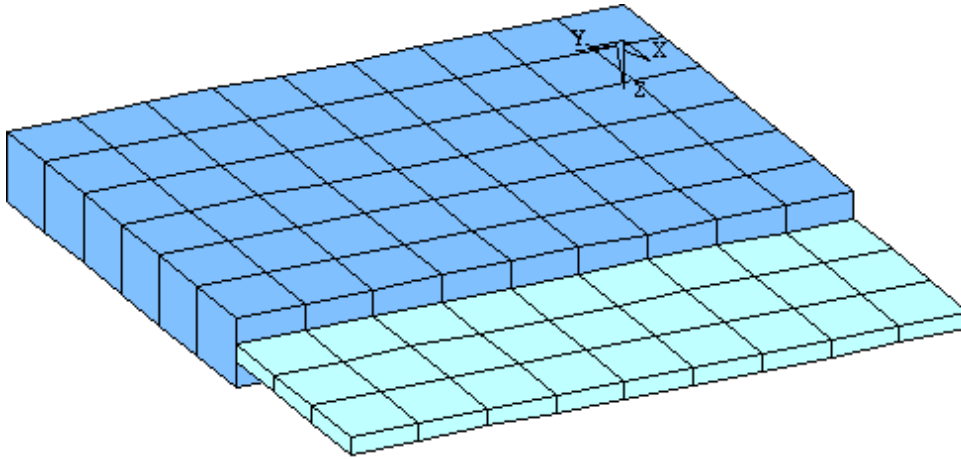
Figure 7.2 Layered Shell (SHELL91 or SHELL99) with Nodes Located at Midplane
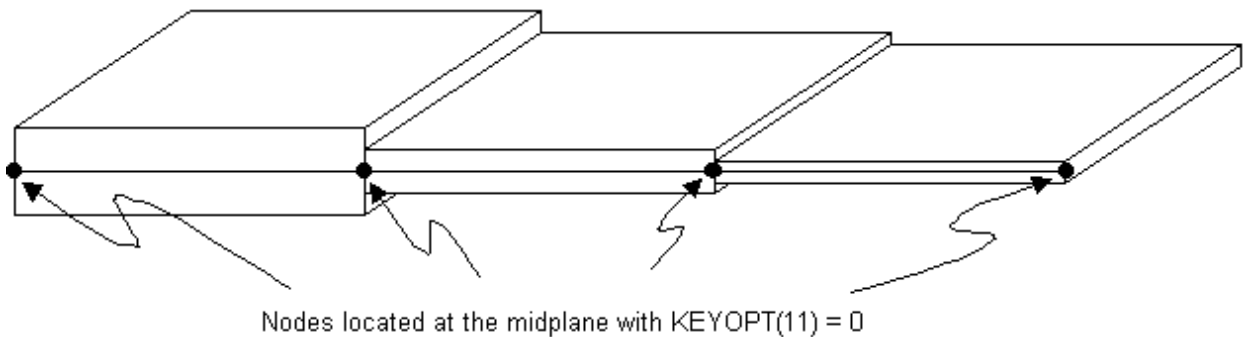
Nodes located at the midplane with KEYOPT(11) = 0

Figure 7.3 Layered Shell (SHELL91 or SHELL99) with Nodes Located at Bottom Surface

Nodes located on bottom surface with KEYOPT(11) = 1
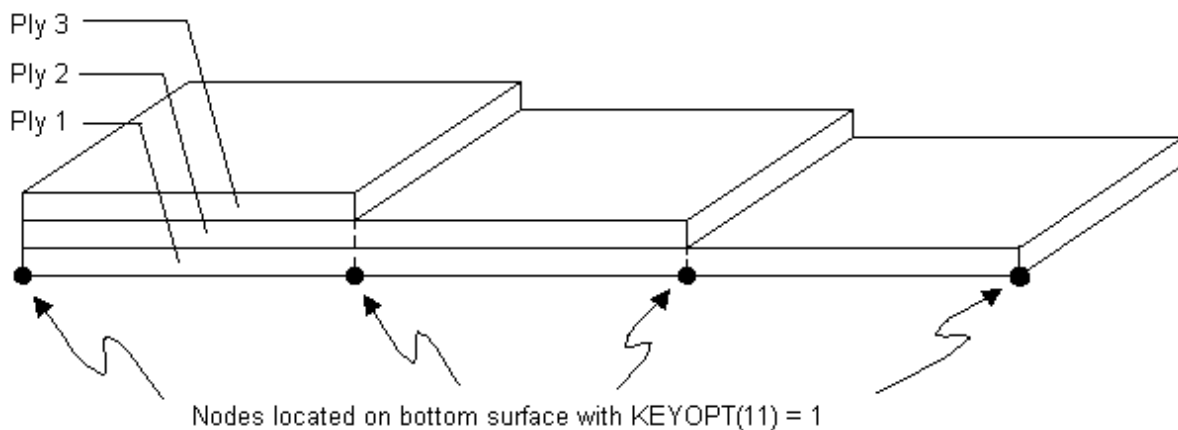
7.4. Grouping Geometry Items into Components and Assemblies

Sometimes it is convenient to group portions of the model and give them recognizable names, such as FLANGE, WHEEL2, FIN7, IRONCORE, STATOR, ROTOR, etc. You can then conveniently select all items belonging to, say, WHEEL2, and work with them: apply boundary conditions, mesh them with nodes and elements, produce graphics displays, and so forth.

The groupings may be *components* or *assemblies*. A *component* consists of *one* type of entity: nodes, elements, keypoints, lines, areas, or volumes.

The Component Manager (**Utility Menu> Select> Component Manager**) provides convenient access to your component operations. The Component Manager provides a coordinated and integrated interface to the capabilities of the following commands:

CM	CMDELE	CMEDIT
CMGRP	CMLIST	CMMOD
CMPLOT	CMSEL	

You can access each command's capability either through the Component Manager, or through individual GUI paths, as noted. The following sections describe the individual component commands and the function you can perform with them. See the appropriate command documentation for specific capabilities and limitations.

> *Note* — Using the Component Manager toolbar buttons (except **Select Component/Assembly** and **Unselect Component/Assembly** will perform the specified operation on the highlighted component(s), but the select status of the entities in the database will not be affected.

7.4.1. Creating Components

Use the **CM** command (**Utility Menu> Select> Comp/Assembly> Create Component**) to define a component. For example, you can select all elements that constitute the rotor portion of a motor model and group them into a component:

```
ESEL,,MAT,,2  ! Select rotor elements (material 2)
CM,ROTOR,ELEM ! Define component ROTOR using all selected elements
```

The *ANSYS Commands Reference* describes the **ESEL** and **CM** commands in more detail.

An *assembly* may consist of any number of components and other assemblies. Use the **CMGRP** command (**Utility Menu> Select> Comp/Assembly> Create Assembly**) to define an assembly. For example, you can group the components ROTOR and WINDINGS (both of which must have been previously defined) into an assembly RO-TORASM:

```
NSEL,...                    ! Select appropriate nodes and
ESLN                        ! elements that constitute the windings
CM,WINDINGS,ELEM            ! Define component WINDINGS
CMGRP,ROTORASM,WINDINGS,ROTOR ! Define the assembly ROTORASM
```

The *ANSYS Commands Reference* describes the **NSEL**, **ESLN**, **CM**, and **CMGRP** commands in more detail.

7.4.2. Nesting Assemblies

You can also nest assemblies up to five levels deep. For example, you can build an assembly named MOTOR from other assemblies and components as shown in the schematic below.

Figure 7.4 Nested Assembly Schematic
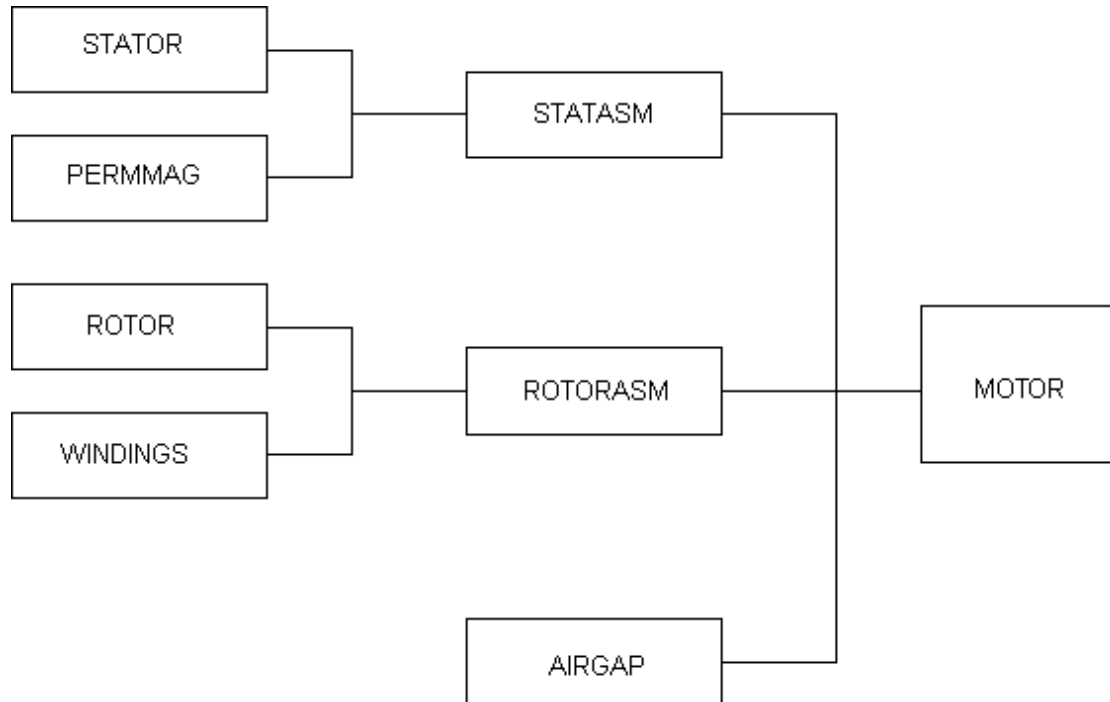
Assuming that the assembly ROTORASM and components STATOR, PERMMAG, and AIRGAP have been defined, the commands to define the assembly MOTOR would look like this:

```
CMGRP,STATASM,STATOR,PERMMAG
CMGRP,MOTOR,STATASM,ROTORASM,AIRGAP
```

See the *ANSYS Commands Reference* for more information about the **CMGRP** command.

7.4.3. Selecting Entities by Component or Assembly

The main advantage of defining a component or an assembly is that you can conveniently select items that belong to it using a combination of the **CMSEL** and **ALLSEL** commands. The **CMSEL** command selects all entities belonging to a component or assembly by its name. You can then issue **ALLSEL**,BELOW to select all attached lower entities. For example, you can select all elements belonging to the WINDINGS component, apply a current density loading to all of them, and then select all nodes attached to those elements:

```
CMSEL,,WINDINGS
BFE,ALL,JS,,-1000
ALLSEL,BELOW,ELEM
```

You can also use the picker to select components. By choosing **Utility Menu> Select> Comp/Assembly> Pick Comp/Assembly**, you can select a defined component and all of the items belonging to it. The item is displayed in the prompt window during the select process.

For more information about the **CMSEL**, **BFE** and **ALLSEL** commands, and the **CMEDIT**, **CMDELE**, and **CMLIST** commands mentioned below, see the *ANSYS Commands Reference*.

7.4.4. Adding or Removing Components

Issuing the **CMEDIT** command (**Utility Menu> Select> Comp/Assembly> Edit Assembly**) allows you to add components to or remove components from an assembly. For example, the following command removes AIRGAP from the assembly MOTOR:

```
CMEDIT,MOTOR,DELE,AIRGAP
```

You can delete a component or assembly definition, using the **CMDELE** command (**Utility Menu> Select> Comp/Assembly> Delete Comp/Assembly**). You can list the entities that make up a particular component with the **CMLIST** command (**Utility Menu> Select> Comp/Assembly> List Comp/Assembly**). You can use **CMLIST** to generate expanded, detailed listings of the entities that make up specific components by using it along with the **CMSEL** command.

The **CMSEL** command also allows you to use components to narrow your selection or increase your selection criteria. Issuing **CMSEL**,ALL will select all defined components **in addition to** any items you already have selected.

7.4.5. Modifying Components or Assemblies

You can modify the specification of a component with the **CMMOD** command.

If an entity is modified (e.g., via the **KMODIF** command), that entity may be deleted and then redefined. The deletion may cause the entity to be removed from the component. If all of the entities are removed from the component, the component will also be deleted.

Chapter 8: Getting Started with Graphics

8.1. Overview

The ANSYS program (and the associated DISPLAY program) enable you to portray almost any aspect of your model in pictures or graphs that you can view on your terminal screen, store on a file, or plot out as hard copy. ANSYS has numerous features to enable you to customize or enhance your graphics displays to suit your individual needs.

8.2. Interactive Versus External Graphics

Any discussion of graphics might seem to imply that you are running the ANSYS program interactively and viewing graphics images on your terminal screen. For the most part, this chapter is written for such a scenario. However, you can run the ANSYS program in either interactive *or* batch mode and store graphics images on a file for later viewing and processing. This process is called creating *external graphics*. Chapter 16, "External Graphics" discusses the procedures for external graphics. Chapter Chapter 9, "General Graphics Specifications" through Chapter 15, "Animation" pertain to obtaining graphics displays interactively on your screen.

8.3. Identifying the Graphics Device Name (for UNIX)

When using the ANSYS program, one of the first things you must do is specify the graphics device name (sometimes referred to as the graphics *driver*). ANSYS requires this information to properly direct graphics instructions to your display device. The default graphics device name for most systems is **X11**. You can change it from X11 to, say, **3D** if you have a 3-D graphics device for running ANSYS.

You must define the graphics device name before you activate the Graphical User Interface (GUI). Once you have activated the GUI, you cannot change graphics device names. Refer to the *ANSYS Operations Guide* for more information about using the GUI.

The best way to identify the graphics device name is to do so directly at program start-up. You can enter the graphics device name in the launcher from the **ANSYS GUI Settings** dialog box, accessed by selecting **Edit> Preferences> ANSYS GUI Settings**. By defining the graphics device name at start-up, you can activate the GUI immediately upon entering the ANSYS program. Alternatively, you can specify the graphics device name using **/SHOW** command once you have entered the program (but before you have activated the GUI).

8.3.1. Graphics Device Names Available

X11 (or X11C) and 3D are common graphics device names supported by the ANSYS program. Each of these are described briefly below.

8.3.1.1. X11 and X11C

Graphics Device Name = X11: The X11 graphics driver incorporates X - a distributed windowing system developed at Massachusetts Institute of Technology that a variety of platforms support. It provides 2-D graphics capability. The ANSYS program currently supports Version 11 (thus, "X11") Release 6 of the X-Window system.

X separates the functionality of traditional graphics systems into two parts: the X server and the X client. The *server* is the part of the system that controls the physical display device. A *client* is a piece of application software, such as the ANSYS or DISPLAY programs. A single server may respond to multiple clients. The server and client may reside on different machines connected to a network. X transparently handles all communication between server and client.

Graphics Device Name = X11C: On 2-D display devices that have more than 16 colors (more than four graphics bit planes; usually eight), the ANSYS program displays the model using light-source shading. Light-source shading means that when the model is viewed obliquely, the display appears to be 3-D. You can activate the extra colors using the *NCPL* field on the **/SHOW** command (**Utility Menu> PlotCtrls> Device Options**).

These devices also offer a 128-contour color option ("C-option"). This option allows contour displays to use the extra colors by adding *more* colors with a single intensity each. By default, the extra colors are used to display nine contour colors with varying intensities that simulate light-source shading. You activate the 128-contour color option by using X11C for the graphics device name on the **/SHOW** command.

Individual items can also be selected and displayed with varying degrees of translucency on 2-D devices. Translucent items will show black on the initial replot, since the 2-D driver generates only the visible face. The **/SHRINK** command (**Utility Menu> PlotCtrls> Style> Translucency**) will force the hardware to plot all of the faces and provide the desired translucent effect.

8.3.1.2. 3D

Graphics Device Name = 3D: If you have a 3-D graphics device, you should specify 3D as the graphics device name. A 2-D device contains a "flat" 2-D projection of your model (image manipulation is performed in software), but a 3-D device contains a 3-D model in its local memory (image manipulation is performed by the display hardware). As a result, 3-D devices perform certain graphics functions in ANSYS more efficiently, and 2-D devices do not support certain functions. The 3-D functions in ANSYS include "real-time" dynamic transformation (rotation, translation, etc.) of your *actual* model, translucency, and control of various lighting options, including reflectance, intensity, light direction, and shading. If you are using a 3-D device, you can set certain display option modes using the **/DV3D** command (**Utility Menu> PlotCtrls> Device Options**).

8.3.2. Graphics Drivers and Capabilities Supported on UNIX Systems

Table 8.1: "ANSYS-Supported 3-D Drivers and Capabilities for UNIX" lists the capabilities that ANSYS supports in various UNIX environments. The supported capabilities are noted with a **Y** in the driver column:

Table 8.1 ANSYS-Supported 3-D Drivers and Capabilities for UNIX

	DEC OpenGL	HP OpenGL	IBM OpenGL	SGI OpenGL	Sun OpenGL
Window Device	Y	Y	Y	Y	Y
Hot Keyboard/Mouse	Y	Y	Y	Y	Y
3-Button Mouse	Y	Y	Y	Y	Y
Remote Network Access	**Y**[1]	Y	Y	Y	Y
Hidden Line Removal	Y	Y	Y	Y	Y
Translucency	Y	Y	Y	Y	Y
Light Source Shading	Y	Y	Y	Y	Y
3-D Local Transformations	Y	Y	Y	Y	Y
Double Buffering	Y	Y	Y	Y	Y
Degenerate Mode	Y	Y	Y	Y	Y

1.   Remote Network Access is restricted to systems that support OpenGL.

8.3.3. Graphics Device Types Supported on UNIX Systems

Table 8.2: "ANSYS-Supported Graphics Device Types (for UNIX)" summarizes the graphics device types that ANSYS supports in various UNIX environments:

Table 8.2 ANSYS-Supported Graphics Device Types (for UNIX)

Platform	Device	Description
HP AlphaServer (Tru64), HP, IBM, SGI, Sun UltraSPARC	X11 or x11 X11C or x11c	X11Color, X11Color contour
HP AlphaServer (Tru64)	3D or 3d	3-D OpenGL Graphics
HP	3D or 3d	3-D OpenGL Graphics
IBM (32 and 64 bit)	3D or 3d	3-D OpenGL Graphics
SGI	3D or 3d	3-D OpenGL Graphics
Sun Solaris UltraSPARC (32 and 64 bit)	3D or 3d	3-D OpenGL Graphics

8.3.4. Graphics Environment Variables

Table 8.3: "Graphics Environment Variables" lists the environment variables you can set before executing the ANSYS program or the DISPLAY program. Setting these variables alters the behavior of the X11 device driver and (where explicitly stated) also modifies 3-D graphics behavior.

Table 8.3 Graphics Environment Variables

Environment Variable	Affected Driver	Description/Example
ANSCURS	X11	Select cursor style; for example: `setenv ANSCURS 22`
ANSCREV	X11	Reverse cursor color. Used only when **ANSCURS** is set.
ANSVIS	X11	ANSYS visual key; instructs ANSYS to use a specific visual.
ANS_SNGLBUF	3-D	Disables double buffering. Applies to HP and SGI 12-bit plane systems only.

8.4. Specifying the Graphics Display Device Type (for Windows)

For Windows users, ANSYS supports these drivers and capabilities:

- A window device
- Hot keyboard/mouse
- Two- or three-button mouse
- Hidden line removal
- Light source shading

Note — On a two-button mouse, the shift-right button functions like the middle button of a three-button mouse.

If you are running the program on Windows platforms, you have three alternatives for specifying the graphics device type:

- Double-click on the Interactive icon in the ANSYS Program Folder. Click on the down arrow next to **Graphics device name** and choose the appropriate device.

- Within the ANSYS program, issue the ANSYS **/SHOW** command (**Utility Menu> PlotCtrls> Device Options**).

- Include the device type on the ANSYS execution command line. The command option **-d** or **-D** must precede the device type, as shown below:

```
ansys81 -d device_type
```

The device type is one of the following:

- WIN32

- WIN32c

- 3D

We recommend using a color setting higher than 256 colors.

Specifying an invalid device type causes ANSYS to divert the graphics to a disk file and inhibits the opening of the ANSYS menu system, even if you included the **-g** option on the ANSYS execution command.

8.5. System-Dependent Graphics Information

This section describes factors affecting how ANSYS graphics display on different hardware systems. *You should read this information before you activate the ANSYS graphical user interface.*

8.5.1. Adjusting Input Focus

To enable the display, meshing, and listing interrupts to work correctly, you must set the input focus in the text window from which the ANSYS program is executing. You can set the focus in either of two ways:

- Position the mouse pointer within the text window. (Use this method only if the window manager sets the focus automatically.)

- Place the mouse pointer on the text window and click the mouse button.

8.5.2. Deactivating Backing Store

When you are using the X11 graphics driver on Sun SPARC systems, backing store is turned on by default. For faster graphics response turn backing store off by issuing the command shown below:

```
setenv ANSBACK 0
```

8.5.3. Setting Up IBM RS/6000 3-D OpenGL Supported Graphics Adapters

For 3-D OpenGL, initialize the window manager using the command below:

```
xinit -- -x abx  -x dbx  -x glx
```

3-D OpenGL does not apply to Sabine, GT4E, and GT0 graphics adapters.

8.5.4. Displaying X11 Graphics over Networks

You can display X11 graphics within the ANSYS program over the network if the following conditions exist:

- All computer systems have X11 software installed.
- The ANSYS program is linked with the X11 driver.
- A **/SHOW** device type of x11 or x11c is used. (You can use either uppercase or lowercase characters to specify device types.)
- The **/etc/hosts** file on the host machine contains the hostname and the IP address of the remote machine.
- The environment variable **DISPLAY** is set to *Hostname*:0.0, where *Hostname* is either the host name or the IP address of the machine that will display the graphics.

For example, suppose that you want to run the ANSYS program remotely from another UNIX system for local display of X11 graphics on your workstation monitor. You would perform these steps:

1. Open a window on your workstation and issue the following command to authorize remote hosts to access the display:

   ```
   /usr/bin/X11/xhost +
   ```

2. Log onto a remote host (via Telnet, login, etc.). Type the following command or commands to tell the remote host to display X11 graphics on your workstation.

   C Shell:

   ```
   setenv DISPLAY Your_Workstation:0.0
   ```

   Bourne or Korn Shell:

   ```
   DISPLAY=Your_Workstation:0.0
   export DISPLAY
   ```

   Your_Workstation is either the host name of the IP address of your workstation.

3. Execute the ANSYS program and X11 graphics will be displayed on your workstation monitor:

   ```
   ansys81 -d x11 -g
   ```

8.5.5. HP Graphics Drivers

The X11 and 3-D OpenGL graphics drivers are supported on the HP workstations. You must install the OpenGL libraries on the system to use the OpenGL graphics driver.

CRX and HCRX graphics devices can use only the X11 graphics driver, unless you have installed the PowerShade software on the machine.

If you are running HP CDE, set the color Use option to **BLACK AND WHITE**. You can do so using the HP Style Manager - Color Option.

8.5.6. Producing GraphicDisplays on an HP PaintJet Printer

You can produce hard copy outputs from within the ANSYS program on a PaintJet printer when running on an HP workstation. To do so, issue this command:

```
/pcopy,key
```

To produce a hard copy from within the DISPLAY program, use this command:

```
term,copy,key
```

Possible values for *key* are:

0	Turn hard copy option off.
1	Copy each successive display, placing them in a bitmap file named **file.pjet.*xx***.
now	Copy the current display, placing it in a bitmap file named **file.pjet.*xx***.

The *xx* is a two-digit integer between 00 and 99.

You can send the bitmap file resulting from either of the commands shown above to a PaintJet printer. To print the **file.pjet.*xx*** file, use the HP-UX command **pcltrans**. The format for this command is as follows:

```
pcltrans -C -p file.pjet.xx > /dev/paintjet
```

The value **/dev/paintjet** is the device name for the printer. If the printer is connected to a spooler, use the following command:

```
pcltrans -C -p file.pjet.xx  |  lp -oraw
```

The last example assumes that the PaintJet is the default output device.

Notes

- The **-P** option expands the plot to fit on the default paper size of the plotter.

- You may need to use the **-k** option of **pcltrans** to remove the black background on plots created using the X11 graphics driver.

- If the environment variable **SB_X_SHARED_CMAP** is set to true, the **/PCOPY** command may not produce correct plots. To avoid this problem, unset this variable before running either the ANSYS program or the DISPLAY program when **/PCOPY** will be used.

- When using an HCRX 24 or CRX 24 graphics board, you must set the ANSYS environment variable **ANS_SNGLBUF** to 1 to produce graphics displays on the HP PaintJet printer.

8.5.7. PostScript Hard-Copy Option

When you are using the PostScript Hard-Copy option on a CRX 24 or HCRX 24 graphics board, set the environment variable **ANS_SNGLBUF = 1** to get a higher quality image. This variable disables double buffering. Therefore, set it only before you use the Hard-Copy option.

8.5.8. IBM RS/6000 Graphics Drivers

Both X11 and 3-D graphics drivers are supported on the IBM RS/6000 workstations in the AIX windowing environment. The 3-D driver incorporates the Silicon Graphics licensed software, OpenGL.

8.5.9. Silicon Graphics Drivers

Both X11 and SGI OpenGL graphics drivers are supported on the Silicon Graphics (SGI) workstations.

8.5.10. Sun UltraSPARC Graphics Drivers (32 and 64 bit versions)

If ANSYS is not invoked from the launcher or the **ansys81** script, each ANSYS user's **.cshrc** file must contain the following environment variable definitions in order to use the Solaris graphics drivers:

- For the X11 and 3-D OpenGL graphics drivers, the required environment variable definitions are:

    For 32 bit:

    ```
    setenv OPENWINHOME path/openwin
    setenv LD_LIBRARY_PATH
    /ansys_inc/v81/ansys/syslib/usparc:/ansys_inc/v81/ansys/lib/usparc:/usr/lib
    ```

    For 64 bit:

    ```
    setenv OPENWINHOME path/openwin
    setenv LD_LIBRARY_PATH
    /ansys_inc/v81/ansys/syslib/sun64:/ansys_inc/v81/ansys/lib/sun64:/usr/lib
    ```

    Note — You must enter the **setenv LD_LIBRARY_PATH** definition on a continuous line without a carriage return.

8.6. Creating Graphics Displays

You can create many types of graphics displays: geometry displays (nodes, elements, keypoints, etc.), results displays (temperature or stress contours, etc.), and graphs (stress-strain curves, time-history displays, etc.). Creating any display is a two-step process:

1. You use graphics *specification* functions to establish specifications (such as the viewing direction, number and color controls, etc.) for your display.

2. You use graphics *action* functions to actually produce the display.

You can perform both types of graphics functions either by using menu functions in the GUI or by typing in commands directly.

8.6.1. GUI-Driven Graphics Functions

When running the ANSYS program interactively, most users will prefer to use the GUI. As you use the GUI functions, you execute commands without actually seeing or editing them. (The program will record all underlying executed commands in your **Jobname.LOG** file.) You can access graphics specification functions via **Utility Menu> PlotCtrls**. Graphics action functions reside under **Utility Menu> Plot**.

8.6.2. Command-Driven Graphics Functions

As an alternative to using the GUI functions, you can type ANSYS commands directly in the Input Window. In general, you enter the graphics specifications using the graphics "slash" commands (for example, **/WINDOW**, **/PNUM**, etc.). Graphics action commands are usually either prefixed with **PL** (**PLNSOL**, **PLVAR**, etc.) or are suffixed with **PLOT** (**EPLOT**, **KPLOT**, etc.).

8.6.3. Immediate Mode Graphics

By default in the GUI, your model will immediately be displayed in the Graphics Window as you create new entities (such as areas, keypoints, nodes, elements, local coordinate systems, boundary conditions, etc.). This is called *immediate mode* graphics. Anything drawn immediately in this way, however, will be destroyed if you

bring up a menu or dialog box on top of it. Or, if you iconify the GUI, the immediate mode graphics image will not be shown when you restore the GUI icon.

An immediate image will also be automatically scaled to fit nicely within the Graphics Window - a feature called *automatic scaling*. Periodically, though, you may need to issue an explicit plot function because you have created new entities which lie "outside" the boundaries of the scaled image already in the Graphics Window and are thus not captured with immediate mode graphics. The plot function will rescale and redraw the image.

To obtain a more "permanent" image, you need to execute one of the plot functions (such as **Utility Menu> Plot> Volumes**) or a graphics action command (such as **VPLOT**). An image generated in this way will not be destroyed by menu pop-ups or by iconifying the GUI. Also note that symbols (such as keypoint or node numbers, local coordinate systems, boundary conditions, etc.) are also shown immediately but will not be present on a "permanent" display unless you first "turn on" the appropriate symbol using the functions under **Utility Menu> PlotCtrls** or the appropriate graphics specification command.

If you prefer *not* to see things immediately as you define them, you can use the **IMMED** command (**Utility Menu> PlotCtrls> Erase Options> Immediate Display**) to turn off immediate mode. When you run the ANSYS program interactively *without* using the GUI, immediate mode is off by default.

8.6.4. Replotting the Current Display

The **/REPLOT** command (**Utility Menu> Plot> Replot**) re-executes the last display action command that was executed. However, the program can execute that command only if it is valid in the current ANSYS routine. For instance, if you issue a **PLNSOL** command in POST1, then exit that routine and replot while at the Begin level, no contour display will be formed. To save time, you may want to define an abbreviation for the **/REPLOT** command so that it is available on the Toolbar as a "quick pick."

8.6.5. Erasing the Current Display

You can clear the current graphics display by issuing the **ERASE** command (**Utility Menu> PlotCtrls> Erase Options> Erase Screen**). (GUI menus will not be erased, however.)

8.6.6. Aborting a Display in Progress

If you have initiated a display and decide to terminate it before it is completed, invoke your system "break." (Typically, this means moving the mouse pointer to the Output Window and typing Ctrl+C. However, the specific procedure varies from system to system.) **You must execute this break while the display is visibly in progress, or else your entire ANSYS session will terminate.**

8.7. Multi-Plotting Techniques

The multi-plotting capabilities within ANSYS enable you to display both multiple entities within a window and multiple windows with varying entity types. Defining each window's composition is a four-step process:

1. Define the window layout.
2. Choose the entities you want each window to display.
3. If you are displaying elements or graphs, choose the type of element or graph display used for plots.
4. Display the entities you selected.

8.7.1. Defining the Window Layout

You need to define how many windows you want the ANSYS program to use for plotting and how those windows appear on your screen. You have the following layout options:

- One window

- Two windows (left and right of the screen, or top and bottom)

- Three windows (two at the top of the screen and one at the bottom, or one window at the top and two windows at the bottom)

- Four windows (two at the top of the screen and two at the bottom)

To define the window layout, issue the **/WINDOW** command (**Utility Menu> PlotCtrls> MultiWindow Layout**). If you choose the GUI path, the program displays a dialog box, in which you click on the layout you prefer. That dialog box also contains a **Display upon OK/Apply** field, where you also can specify what the ANSYS program displays next. Choices for this field are **Multi-Plots**, **Replot**, and **No redisplay**. When you finish specifying your layout design, click on **Apply** or **OK**.

8.7.2. Choosing What Entities Each Window Displays

Once you have designed your window layout, you choose what entities each window will display. To do so, use either of the following:

       Command(s): /GTYPE,*WN,Label,KEY*
       GUI: Utility Menu> PlotCtrls> Multi-Plot Controls

If you use the GUI path, a dialog box appears. In its **Window to edit** field, click on either **All window** or a specific window number (default is window 1). In the **Display type** field, choose either **Entity plots** or **Graph plots**. Then, click on **OK**. If you choose **Entity plots**, another dialog box appears, listing the types of entities available for display. (You also can choose the type of plots via the **/GCMD** command, as described below.) All entity types *except* **GRPH** are on by default; to turn an entity type off, click on it.

If you use the **/GTYPE** command, for the *WN* argument, either specify ALL to have all windows display the selected entities or choose a specific window number (default is window 1). For *Label*, specify any of these entity types:

- NODE (nodes)

- ELEM (elements)

- KEYP (keypoints)

- LINE (lines)

- AREA (areas)

- VOLU (volumes)

- GRPH (graph displays)

When the GRPH entity type is activated, you can display only x-y graphs, and you cannot use the **/GCMD** command to issue other commands (such as **/TYPE**) that affect displays. (For more information about **/GCMD**, see the *ANSYS Commands Reference* and Section 8.7.3: Choosing the Display Used for Plots of this manual) If the GRPH type is off, you can display any combination of the other solid model or finite element entity types, and you can use **/GCMD** to issue other display control commands.

To turn an entity type on via the **/GTYPE** command, use a *KEY* value of 1. To turn an entity type off, specify a *KEY* of 0.

8.7.3. Choosing the Display Used for Plots

When you are displaying either the ELEM or GRPH entity type, you can control the type of element or graph display used for plots. To do so, use either of the following:

> **Command(s): /GCMD**,*WN,Lab1,...Lab12*
> **GUI: Utility Menu> PlotCtrls> Multi-Plot Controls**

You can specify ALL to have all windows use the selected display type, or you can apply that display type only to a specific window (default is window 1). The *Lab1* through *Lab12* values shown above are labels for commands such as **/TYPE** and **PLNSOL**,S,X. (For the *Lab* arguments, you can specify only commands that have *WN* (window) arguments.)

Issuing the **/GCMD** command is the same as choosing the GUI path shown above, then choosing either **Entity plots** or **Graph plots** for the **Display Type** field.

Following are two command-based examples of selecting a type of element or graph display.

- To display a **PLNSOL**,S,X command in window 1 when the ELEM entity type is activated, issue the command **/GCMD**,1,PLNS,S,X.

- To change from an element display to a von Mises display, issue the command **/GCMD**,1,PLNS,S,EQV.

8.7.4. Displaying Selected Entities

To display the entities you selected, issue the **GPLOT** command (**Utility Menu> PlotCtrls> Multi-Plots** or **Utility Menu> Plot> Replot**).

Chapter 9: General Graphics Specifications

9.1. Overview

Many graphics features apply to any kind of ANSYS graphics display. These general graphics specifications affect such features as multiple ANSYS windows, viewing directions, zooming and panning your image, etc.

9.2. Using the GUI to Control Displays

The most convenient way to create and control your displays is by using the functions available under **Utility Menu> Plot** and **Utility Menu> PlotCtrls**. Alternatively, you can use graphics action and control commands, as described elsewhere in this manual and below.

You can exercise the features this chapter describes for any kind of ANSYS display, whether they are geometry displays, results displays, or graphs.

9.3. Multiple ANSYS Windows, Superimposed Displays

An *ANSYS window* is a rectangular portion of your terminal screen which lies inside the main Graphics Window. ANSYS windows are defined in screen coordinates (X_s, Y_s). You can define up to five different windows, which can be placed anywhere within the Graphics Window, and which can overlap. Each window can have different graphics *specification* settings. However, graphics *action* commands will apply to *every* active window.

9.3.1. Defining ANSYS Windows

To define the size and placement of an ANSYS window, use either method shown below. You can use convenience labels in this command to size and place windows in the top half, bottom half, right top quadrant, etc. of the Graphics Window.

> **Command(s): /WINDOW**
> **GUI: Utility Menu> PlotCtrls> Window Controls> Window Layout**

9.3.2. Activating and Deactivating ANSYS Windows

You can activate and deactivate existing ANSYS windows by entering ON or OFF in the *XMIN* field on the **/WINDOW** command (**Utility Menu> PlotCtrls> Window Controls> Window On or Off**).

9.3.3. Deleting ANSYS Windows

To delete a window, either enter DELE in the *XMIN* field on the **/WINDOW** command (**Utility Menu> PlotCtrls> Window Controls> Delete Window**).

9.3.4. Copying Display Specifications Between Windows

Use the *NCOPY* field on the **/WINDOW** command (**Utility Menu> PlotCtrls> Window Controls> Copy Window Specs**) to copy a set of display specifications (**/VIEW**, **/DIST**, etc.) from one window to another window.

9.3.5. Superimposing (Overlaying) Multiple Displays

If you want to display dissimilar items in separate ANSYS windows, you must issue a sequence of different action commands as you activate and deactivate appropriate windows, while protecting the displays in your deactivated

windows from being erased. The key to this operation is the **/NOERASE** command (**Utility Menu> PlotCtrls> Erase Options> Erase Between Plots**), which prevents the normal screen erase from occurring as new displays are created. Once your multiple display has been created, you can return to normal erasing mode by issuing the **/ERASE** command.

9.3.6. Removing Frame Borders

The FRAME label on the **/PLOPTS** command enables you to turn all your ANSYS window border lines on and off.

9.4. Changing the Viewing Angle, Zooming, and Panning

Using these display specifications is similar to using a camera. The following sketch illustrates the concepts of *focus point*, *viewpoint*, and *viewing distance*, discussed below.

Figure 9.1 Focus Point, Viewpoint, and Viewing Distance
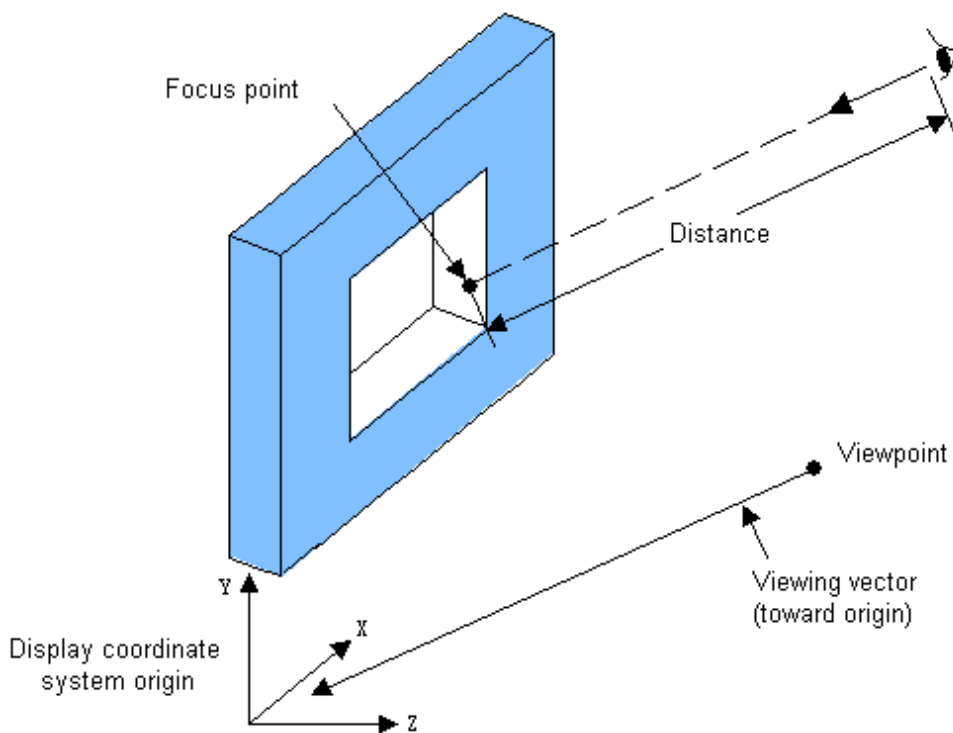
9.4.1. Changing the Viewing Direction

The viewing direction is established by a vector directed from the *viewpoint* to the display coordinate system origin. You use the **/VIEW** command to define the position of the viewpoint in the display coordinate system.

>   **Command(s): /VIEW**
>   **GUI: Utility Menu> PlotCtrls> Pan, Zoom, Rotate**
>   **Utility Menu> PlotCtrls> View Settings> Viewing Direction**

You can also specify **/VIEW**,WN,WP to align the view perpendicular to the current working plane.

Use the following shortcut to pan, zoom, and rotate a graphics display: Press the CONTROL key and hold it down. You are now in Dynamic Manipulation Mode. Notice that the cursor assumes a different shape. Still holding the CONTROL key down, use the mouse buttons to manipulate your view of the display. When you want to leave Dynamic Manipulation Mode, simply release the CONTROL key.

You can also remap your mouse buttons to match the operation (in dynamic mode only) of other programs. The command **/UIS**,BORD,*LEFT*,*MIDDLE*,*RIGHT* can be used. See the **/UIS** command for more information on dynamic mode mouse button remapping.

> *Note* — If you are a Windows ANSYS user performing dynamic manipulation (panning, zooming, rotating), do not use the 256-color setting, which is the default on many systems and which slows down computer performance. To change the color setting, select the Start button in the bottom left-hand corner of the terminal screen and choose **Settings> Control Panel> Display> Settings**. Change the Color Palette drop-down list to True Color, or, at least, the 650536 value. Increase resolution to the maximum value allowed for that setting. Also note that even though you can now run 3-D graphics without a 3-D card, it is highly recommended that you use a 3-D accelerated card to improve dynamic rotation and other plotting speed.

9.4.2. Rotating the Display About a Specified Axis

To rotate the graphics display about the screen axes or about the global Cartesian axes, use any of the following. (The right-hand rule defines positive angular rotation about any axis.)

> **Command(s): /ANGLE**, **/XFRM**
> **GUI: Utility Menu> PlotCtrls> Pan, Zoom, Rotate**
> **Utility Menu> PlotCtrls> View Settings> Angle of Rotation**
> **Utility Menu> PlotCtrls> View Settings> Rotational Center> By Pick**
> **Utility Menu> PlotCtrls> View Settings> Rotational Center> By Location**
> **Utility Menu> PlotCtrls> View Settings> Rotational Center> Reset to Focus Point**

9.4.3. Determining the Model Coordinate System Reference Orientation

The **/VUP** command (**Utility Menu> PlotCtrls> View Settings> Viewing Direction**) determines the "starting" orientation of your display. For instance, with the viewpoint and rotation at their default settings, **/VUP**,WN,X orients the display such that the positive X axis is vertical pointing upward, Y is horizontal pointing to the left of the screen, and Z points out of the screen.

9.4.4. Translating (or Panning) the Display

The *focus point* is that point on your model that appears at the center of your ANSYS windows. You can define or redefine the focus point (in terms of the global Cartesian coordinate system) as follows:

> **Command(s): /FOCUS**
> **GUI: Utility Menu> PlotCtrls> Pan, Zoom, Rotate**
> **Utility Menu> PlotCtrls> View Settings> Focus Point**

This same command also allows you to translate the focus point along the screen axes or along the global Cartesian axes.

9.4.5. Magnifying (Zooming in on) the Image

The *viewing distance* represents the distance between the observer and the focus point, and determines the magnification of your image. Smaller viewing distances magnify the image (zoom in), and larger distances shrink the image (zoom out). To change the viewing distance:

> **Command(s): /DIST**
> **GUI: Utility Menu> PlotCtrls> Pan, Zoom, Rotate**
> **Utility Menu> PlotCtrls> View Settings> Magnification**

9.4.6. Using the Control Key to Pan, Zoom, and Rotate - Dynamic Manipulation Mode

Press the CONTROL key and hold it down to enter Dynamic Manipulation Mode. Notice that the cursor assumes a different shape. You can now use your mouse buttons to pan, zoom, and rotate the graphics display. When you want to leave Dynamic Manipulation Mode, simply release the CONTROL key.

9.4.7. Resetting Automatic Scaling and Focus

Anytime that you change the viewing distance or focus point, your explicitly-defined settings become "frozen." That is, automatic scaling or centering of the image are turned off for subsequent displays. ("Frozen" parameters are preceded with an asterisk in the legend column of the display.) To restore automatic scaling and focus, use one of the methods shown below:

> **Command(s): /AUTO**
> **GUI: Utility Menu> PlotCtrls> Pan, Zoom, Rotate**
> **Utility Menu> PlotCtrls> View Settings> Automatic Fit Mode**

9.4.8. Freezing Scale (Distance) and Focus

By default, your display will be automatically scaled and centered such that the image of your model will just fill your ANSYS windows. If you want to "freeze" these automatically-generated scale and focus settings, use one of these methods:

> **Command(s): /USER**
> **GUI: Utility Menu> PlotCtrls> View Settings> Automatic Fit Mode**

9.5. Controlling Miscellaneous Text and Symbols

You can control the display of different symbols and text entries in your graphics window. These items can help to clarify the way your data is displayed. Although many of these items are controlled by commands, the GUI provides an interfaces to many of the commands to allow the selection and placement of the items you desire.

9.5.1. Using Legends in Your Displays

You can use legends to help define and clarify the data in your display. The Window Options Dialog Box is the "master" legend control. It controls whether or not the legend is displayed, the type of legend display, and in some cases, the content of your legend. The position of the Triad is also controlled from this dialog box. See Figure 9.2: "The Window Options Dialog Box" below.

The **INFO** pull down window provides control for the type of legend. It allows you to turn legend displays on and off, and also to access either the Auto Legend or the Multi-Legend display. The on and off settings control the display of all legend items, for all types of legends.

The Legend On and Auto legend selections control the display of the documentation column. The documentation column display places all of your legend data along the right side of the graphics window and resizes your model area appropriately. Legend On displays the documentation data at all times, while Auto Legend displays the appropriate data, only when it is applicable.

The Multi Legend provides placement options for your text and contour scales within the model area of your graphics window. The Multi Legend options are discussed below.

Figure 9.2 The Window Options Dialog Box
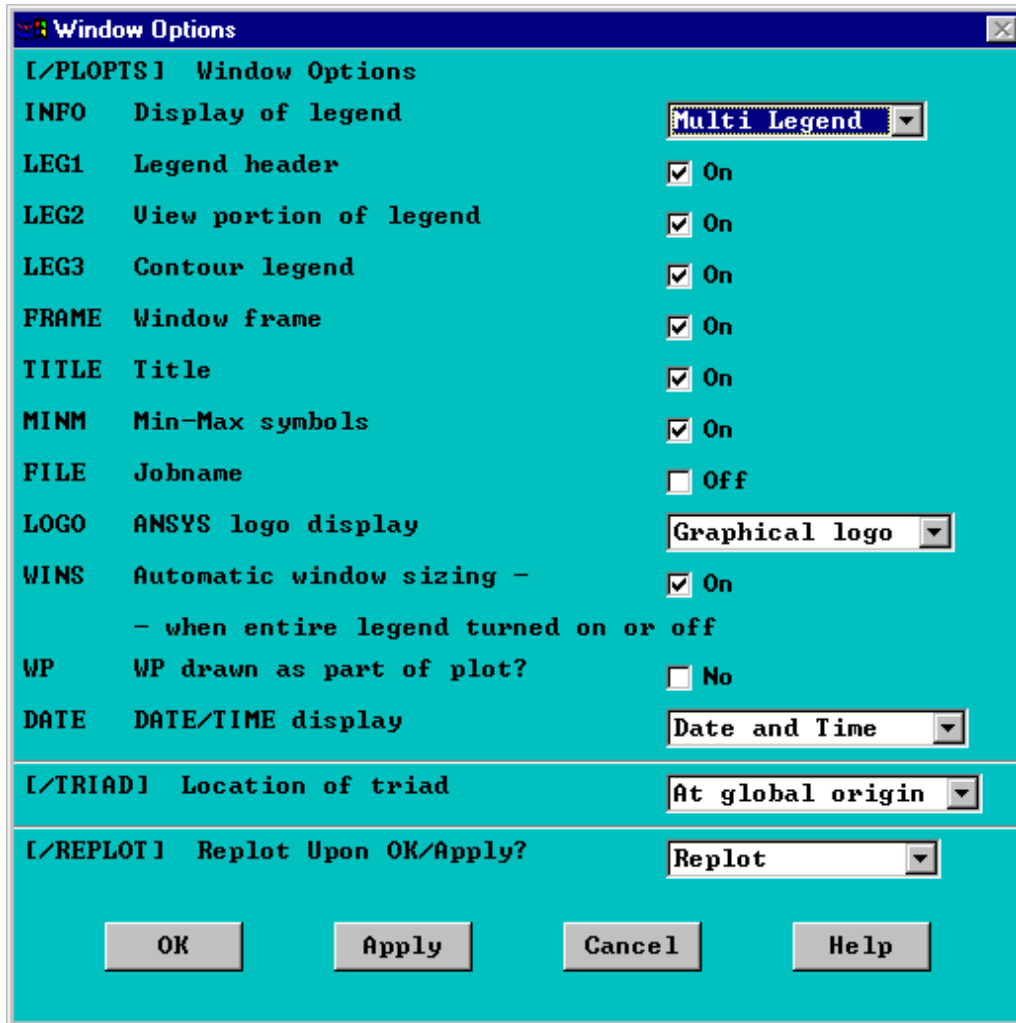
The default, legend setting is the user-defined "Multi-Legend." (**Utility Menu> Plot Ctrls> Window Controls> Window Options> MultiLegend** - **/PLOPTS**,INFO,3).

9.5.1.1. Controlling the Content of Your Legends

The window options dialog box shown above in Figure 9.2: "The Window Options Dialog Box" controls the type of legend, along with the content of the documentation column. You control the content of the Multi Legend via dialog boxes found at **Utility Menu> Plot Ctrls> Style> Multi Legend Options**. The Text Legend dialog box shown in Figure 9.3: "The Multi Legend Text Legend" provides control of the content and placement of the various text items available for the Multi Legend option. This dialog box corresponds to the controls and priorities listed in the **/UDOC** command.

Figure 9.3 The Multi Legend Text Legend
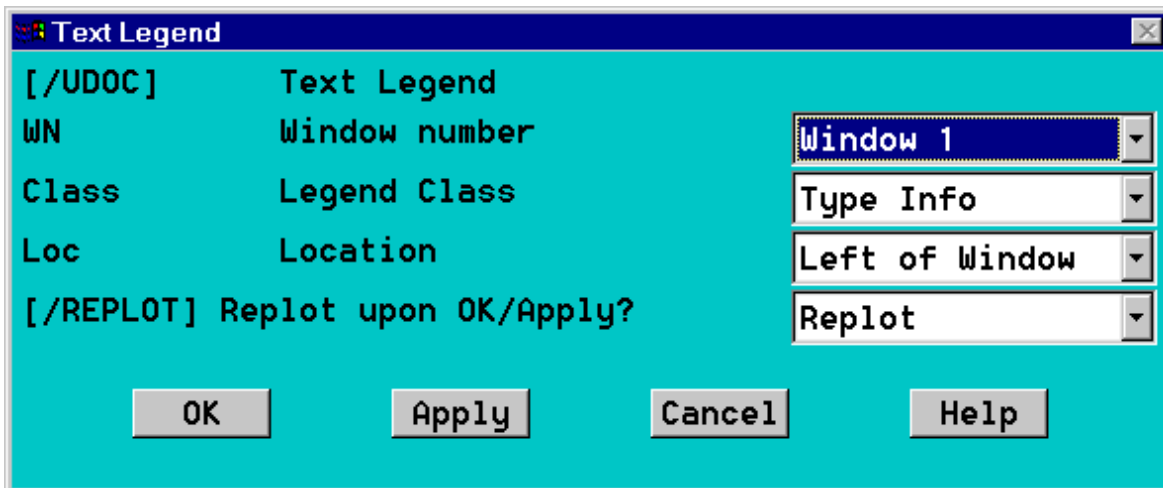
9.5.1.2. Controlling the Placement of Your Contour Legend

The Multi Legend setting allows you to place your contour scales along the four sides of the graphics window. You access this control via **Utility Menu> Plot Ctrls> Style> MultiLegend Options> Contour Legend**. The Contour Legend Dialog box is shown in Figure 9.4: "The Multi Legend Contour Legend". This dialog box corresponds to the controls and priorities listed in the **/UDOC** command.

Figure 9.4 The Multi Legend Contour Legend
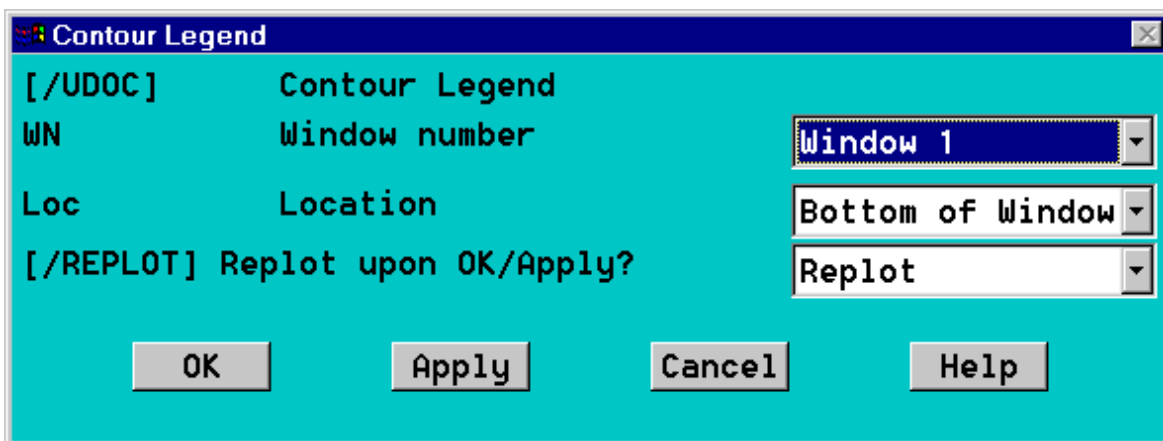
Note — The settings in the Window Options dialog box will in many cases take precedence over your Multi Legend Options settings. See the command documentation for **/UDOC** and **/PLOPTS** for a complete discussion of these dependencies.

9.5.2. Controlling Entity Fonts

You can change the appearance of the fonts that are used to produce the numbers and characters that are shown on your displays. Through the ANSYS GUI, choose the DISPLAY Program or **Utility Menu> PlotCtrls> Font Controls**, or issue either the **/DEVICE**,FONT,*KEY* or **/DEVDISP**,FONT,*KEY* command. Each of these commands requires *Val1* through *Val6* as arguments. These arguments allow you to indicate the family name of the font that you wish to use (e.g., Courier), the weight of the font (e.g., medium), font size, and other attributes which define font selection. (See the *ANSYS Commands Reference* for more information about the requirements of the **/DEVICE**,FONT,*KEY* and **/DEVDISP**,FONT,*KEY* commands.)

9.5.3. Controlling the Location of the Global XYZ Triad

The **/TRIAD** command (**Utility Menu> PlotCtrls> Window Controls> Window Options**) enables you to change the location of the global triad symbol on your display. (The actual mathematical position of the global origin will not change.)

9.5.4. Turning Triad Symbols On and Off

Use the **/TRIAD** command to turn the global triad on and off. Use the **/PSYMB** command (**Utility Menu> PlotCtrls> Symbols**) to control the local, nodal, and element coordinate system triads. Use one of the following to control the working plane triad:

      Command(s): WPSTYL
      GUI: Utility Menu> List> Status> Working Plane
      Utility Menu> WorkPlane> Display Working Plane
      Utility Menu> WorkPlane> Display Working Plane> Offset WP by Increments
      Utility Menu> WorkPlane> Display Working Plane> Show WP Status
      Utility Menu> WorkPlane> Display Working Plane> WP settings

9.5.5. Changing the Style of the Working Plane Grid

You can display the working plane grid as a triad only, grid only, or both triad and grid. Use **WPSTYL** to change from one style to another. There are two methods of turning the working plane on for displays:

- Issuing **WPSTYL** with no arguments toggles the working plane grid, asterisk, and triad on and off immediately, as an "overlay" image on the existing display.

- **/PLOPTS**,WP,ON specifies that the working plane be turned on for *subsequent* displays. In this case, the working plane is drawn as *part of* the display (not just an overlaid image as in **WPSTYL**). For this reason, this method is best used in combination with a hidden-line technique for viewing the location of the working plane with respect to a 3-D model. **WPSTYL** and its GUI equivalents control whether the working plane is displayed as a triad only, grid only, or both.

9.5.6. Turning the ANSYS Logo On and Off

By issuing **/PLOPTS**,VERS,1, you cause the ANSYS logo to appear in the upper right corner of the screen (along with the version number).

9.6. Miscellaneous Graphics Specifications

ANSYS includes a number of miscellaneous graphics commands that let you manipulate your graphics environment.

9.6.1. Reviewing Graphics Control Specifications

Issuing the **/PSTATUS** command (**Utility Menu> List> Status> Graphics> General**) lists the current graphics control specifications. To see the graphics specifications for one window only, specify the window number instead of **General**.

9.6.2. Restoring Defaults for Graphics Slash Commands

Use the **/RESET** command (**Utility Menu> PlotCtrls> Reset Plot Ctrls**) to restore the default settings of **/WINDOW**, **/TYPE**, **/VIEW**, and other graphics "slash" commands.

9.6.3. Saving the Display Specifications on a File

Choose the **/GSAVE** command (**Utility Menu> PlotCtrls> Save Plot Ctrls**) to write a copy of your graphics "slash" command settings on an ASCII text file.

9.6.4. Recalling Display Specifications from a File

You can read graphics "slash" commands from an ASCII text file, using the **/GRESUME** command (**Utility Menu> PlotCtrls> Restore Plot Ctrls**), or by issuing **/INPUT,Filename** (**Utility Menu> File> Read Input from**) where **Filename** is the file of graphics specifications.

9.6.5. Pausing the ANSYS Program

If you prepare an input file for demonstration or presentation purposes, you might find it useful to pause the program after creating a display, to allow the display to be viewed for a reasonable length of time. You can do so by adding **/WAIT** commands to your input stream after the display action commands. The **/WAIT** command has no GUI equivalent.

9.7. 3-D Input Device Support

ANSYS provides integrated support for the Spaceball and Spacemouse 3-D input devices. These devices detect slight fingertip pressures and resolve them into X, Y, and Z translations, rotation components, and movements of your 3-D images. This provides smooth, dynamic, interactive, simultaneous six-degree-of-freedom control of 3-D graphical images or objects. These devices are designed to be used in conjunction with the mouse, not in place of it.

The requisite developer's kit software had been included in the applicable ANSYS code, and specific ANSYS drivers are available at the respective web sites.

http://www.spacemouse.com/

http://www.labtec.com

If problems are encountered, you should try loading different drivers for the devices, either older drivers, or drivers from similar operating systems. Legacy drivers, for older models of these devices are also available. Please contact the appropriate manufacture if you have any questions, or require any additional information on these devices.

Chapter 10: PowerGraphics

10.1. Two Methods of Graphics Displays

There are two methods available for graphics displays:

- The Full Model display method. You invoke this method using **/GRAPHICS**,FULL command (**Utility Menu> PlotCtrls>Style> Hidden-Line Options**).

- The Power Graphics display method. You invoke this method using the **/GRAPHICS**,POWER command (**Utility Menu> PlotCtrls> Style> Hidden-Line Options**).

The PowerGraphics method is the default when the ANSYS GUI is on. This method is valid for all element types except for circuit elements. The Full Model method can be used for all element types. The display method you choose depends upon the size of your model and the type of elements used in the model. If your model contains circuit elements, for example, you select the Full Model method. If you select the PowerGraphics method for a model containing circuit elements, ANSYS automatically uses the Full Model method instead. On the other hand, if you are creating a large model containing element types that are supported by PowerGraphics, the Power-Graphics method offers significantly faster performance than the Full Model method.

10.2. Characteristics of PowerGraphics

- Displays for large models are plotted at a much greater speed than with the Full Model method.

- PowerGraphics plots quadratic (curved) surfaces for midside node elements.

- This method can display discontinuous results due to material type and real constant discontinuities.

- Shell element results are displayed at both top and bottom layers, simultaneously.

- You can use the *Query* picking option to query subgrid results for some elements in the Graphical User Interface.

- PowerGraphics is not available for circuit elements.

- When requested results data are not supported by PowerGraphics, the results are output using the Full Model method.

- Results averaging occurs using only the data at the model surface.

- PowerGraphics supports only the results coordinate system for plotting results data. (The elements-based coordinate system is not supported.)

10.3. When to Use PowerGraphics

Using the PowerGraphics display method has distinct advantages, since graphics displays are plotted at a much faster rate of speed than with the Full Model method. In addition, PowerGraphics produces more realistic results at material type and real constant discontinuities in the model. See the description of the **/GRAPHICS** command (in the *ANSYS Commands Reference*) for more information.

10.4. Activating and Deactivating PowerGraphics

There are two ways to activate and deactivate the PowerGraphics display method: Through the Graphical User Interface (GUI), and through the **/GRAPHICS** command.

- PowerGraphics is the default method when the ANSYS GUI is active. You can switch to the Full Model method, however, by taking one of the following actions:

    1. Click on the POWRGRPH button in the Toolbar of the Graphical User Interface. This selection opens a dialog box which allows you to turn PowerGraphics off or on.

    2. Deactivate or activate PowerGraphics by selecting **Utility Menu> PlotCtrls> Style> Hidden-Line Options**

- You can deactivate PowerGraphics by issuing the command **/GRAPHICS**,FULL, or you can activate PowerGraphics by issuing the command **/GRAPHICS**,POWER.

    Note — Issuing the **/PMETH**,ON command activates PowerGraphics unless you issued a prior **/GRAPH-ICS**,FULL command. Similarly, **/PMETH**,OFF deactivates PowerGraphics unless **/GRAPHICS**,POWER has been previously issued.

10.5. How to Use PowerGraphics

When the PowerGraphics method for graphics displays is active, it is used for element, area, volume, line, and result displays and result data listings. PowerGraphics does not support the graphics display or listing for circuit elements; for such cases, ANSYS automatically activates the Full Model graphics method and uses it for that display or listing. See the **/GRAPHICS** command description for more information.

10.6. What to Expect from a PowerGraphics Plot

Since PowerGraphics plots or listings are given for the exterior surface of the model, you can expect to see differences in these results, compared to those given when using the Full Model method. The averaging calculations for PowerGraphics include results for only the model surface. The averaging calculations, plots, and listings for the Full Model method include results for the entire model (interior and exterior surfaces). Therefore, the PowerGraphics and Full Model methods display results values differently for nodal results (but not for element results).

PowerGraphics makes the **EPLOT**, **APLOT**, **VPLOT**, **LPLOT**, **PLDISP**, **PLNSOL**, and **PRNSOL** commands behave differently than with the Full Model method. For details, see these commands' descriptions in the *ANSYS Commands Reference*.

10.6.1. Viewing Your Element Model

The subgrid approach used by PowerGraphics allows you to control the amount of displayed element curvature. You can plot varying degrees of curvature in your model by specifying the number of facets to be used for element display. Facets are piecewise linear approximations of the actual curve represented by the element face or edge. You specify the number of facets per element edge using one of the following:

> **Command(s): /EFACET**
> **GUI: Main Menu> General Postproc> Options for Outp**
> **Utility Menu> List> Results> Options**
> **Utility Menu> PlotCtrls> Style> Size and Shape**

The more facets you specify, the smoother the representation of the element surface for PowerGraphics plots.

The subgrid approach affects both the display of geometric curvature and the display and printout of results quantities (displacements, stresses, etc.). However, when you use PowerGraphics in POST1 for derived quantities on solid elements, the maximum value on the plot and the maximum value in the printout may not agree. PowerGraphics displays do not average at geometric discontinuities. The printouts in PowerGraphics will, however,

provide averaging information at geometric discontinuities if the models do not contain shell elements. Carefully inspect the data you obtain at geometric discontinuities.

10.6.2. Printing and Plotting Node and Element Results

You can list displacements, stresses, and strains at all node locations (both corner and midside nodes), using the **PRNSOL** command (**Utility Menu> List> Results> Nodal Solution**). For shell elements, you can list results and plot them at the top/bottom and middle layer locations. Likewise, these nodal values can be contoured for display purposes using the **PLNSOL** command (**Utility Menu> Plot> Results> Contour Plot> Nodal Solution**). The number of facets per element edge that you specify determines contour resolutions.

Note that results values for shell elements are displayed simultaneously for the top and bottom layers.

When viewing nodal results using PowerGraphics (**PRNSOL**, **PLNSOL**, or the GUI **Query** function), you can average results in various ways. To choose how results are averaged, use the **AVRES** command (**Main Menu> General Postproc> Options for Outp** or **Utility Menu> List> Results> Options**). (**AVRES** has no effect on the Degree of Freedom solution values (UX, UY, TEMP, etc.). You can average results at all boundaries (default), or at all boundaries except where real constant and/or material discontinuities exist. *Results are not averaged at geometric discontinuities.*

> *Note* — In Full Graphics mode, it is possible to deselect an individual node, select all elements (including the element that contains that node), and then perform postprocessing calculations on those elements and have that unselected node not be considered in those calculations. However, if PowerGraphics is active postprocessing always displays based on selected elements.

Plotting and printing of element results are similar to that for the Full Model graphics method; you use the **PLESOL** or **PRESOL** command, or one of the following GUI paths:

GUI:

> **Main Menu> General Postproc> Plot Results> Contour Plot> Element Solu**
> **Utility Menu> Plot> Results> Contour Plot> Elem Solution**
> **Main Menu> General Postproc> List Results> Element Solution**
> **Utility Menu> List> Results> Element Solution**

The program unaverages nodal results and sorts them by element number. Averaging results does not affect element results plots. Results are for all nodal locations on the model surface. If you issued the **/EFACET**,1 command, the results for the midside nodes are not listed.

PowerGraphics does not support safety factor calculations.

> **Caution:** In unusual cases, your model may contain element types having different results data sets. If so, be sure to unselect those element types which do not have the data set you are reviewing. This prevents zero values from being averaged with valid results. For example, if your model contains FLUID30 (Acoustic Fluid) and SOLID45 (Structural Solid) elements, unselect all SOLID45 elements before viewing a pressure gradient.

Chapter 11: Creating Geometry Displays

11.1. Using the GUI to Display Geometry

A *geometry display* is a display of your model's geometric features (keypoints, areas, nodes, elements, loads, etc.). This is the kind of display that you might typically produce during the model-generation and load-definition phases of your analysis. Figure 11.1: "Typical Geometry Display" shows a typical geometry display.

Figure 11.1 Typical Geometry Display


```
1
                                          13:07:16
                                          VOLUMES
                                          TYPE NUM

                                          XV   =1
                                          YV   =2
                                          ZV   =3
                                          DIST=4.941
                                          PRECISE HIDDEN



Dash Pot Lifter
```
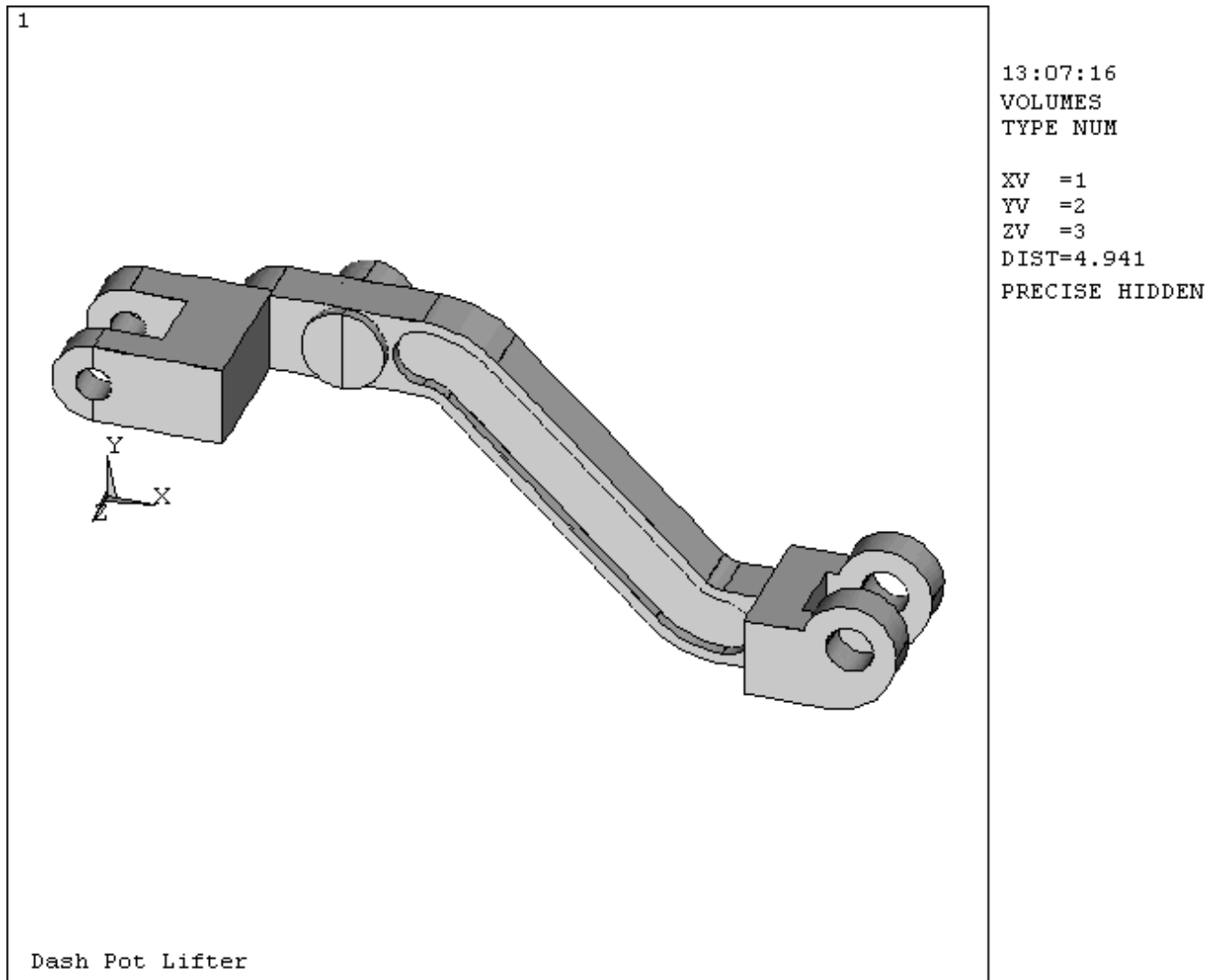
Many users find that the most convenient way to create and control geometry displays is by using the functions available under **Utility Menu> Plot** and **Utility Menu> PlotCtrls**. Alternatively, you can use graphics action and control commands, as described in the following subsections.

11.2. Creating Displays of Solid-Model Entities

The following commands will create displays of solid-model entities:

Table 11.1 Commands for Displaying Solid-Model Entities

Command	GUI Menu Paths	Purpose
APLOT	**Main Menu> Preprocessor> Modeling> Operate> Show Degeneracy> Plot Degen Areas** **Utility Menu> Plot> Areas** **Utility Menu> Plot> Specified Entities> Areas**	Displays a plot of areas
EPLOT	**Utility Menu> Plot> Elements**	Displays a plot of elements
KPLOT	**Utility Menu> Plot> Keypoints** **Utility Menu> Plot> Specified Entities> Keypoints**	Displays a plot of keypoints
LAYPLOT	**Utility Menu> Plot> Layered Elements**	Displays the layer stacking sequence and layer angle orientation of layered element types such as SOLID46 and SHELL91
LPLOT	**Utility Menu> Plot> Lines** **Utility Menu> Plot> Specified Entities> Lines**	Displays a plot of lines
NPLOT	**Utility Menu> Plot> Nodes**	Displays a plot of nodes
/REPLOT	**Utility Menu> Plot> Replot**	Re-executes the last display action executed
VPLOT	**Main Menu> Modeling> Preprocessor> Operate> Show Degeneracy> Plot Degen Volus**	Displays a plot of degenerated volumes

The controls you establish before you invoke these actions can also cause your displays to contain other information, such as lower-order entity numbers (for instance, node numbers associated with selected elements), loads, etc.

11.3. Changing the Specifications for Your Geometry Displays

In addition to the features listed below, also see Chapter 8, "Getting Started with Graphics" for general graphics specifications that apply to any type of display, including geometry displays.

11.3.1. Changing the Style of Your Display

The following sections describe a number of ways to change the way your models are displayed.

11.3.1.1. Displaying Line and Shell Elements as Solids

If your model consists of line elements (such as beams and pipes) or shell elements, you can use the following to display many of them as solids:

> **Command(s): /ESHAPE**
> **GUI: Utility Menu> PlotCrls> Style> Size and Shape**

The ANSYS program uses a rectangular cross section for beams and shells, and uses circular cross sections for pipes. The element real constants are used to proportion the cross section.

You can also use the **/ESHAPE** command to show the orientation of reinforcing (rebar) in SOLID65 elements (see Figure 11.2: "Element Plot of SOLID65 Concrete Elements"). For the rebar to be visible, you must enable vector mode using the **/DEVICE** command (**Utility Menu> PlotCtrls> Device Options**). You must also activate a basic plot type using the **/TYPE** command (**Utility Menu> PlotCtrls> Style> Hidden-Line Options**). To view the rebar, issue these commands in the following order:

```
/ESHAPE,1
/TYPE,,BASIC
/DEVICE,VECTOR.ON
EPLOT
```

Figure 11.2 Element Plot of SOLID65 Concrete Elements
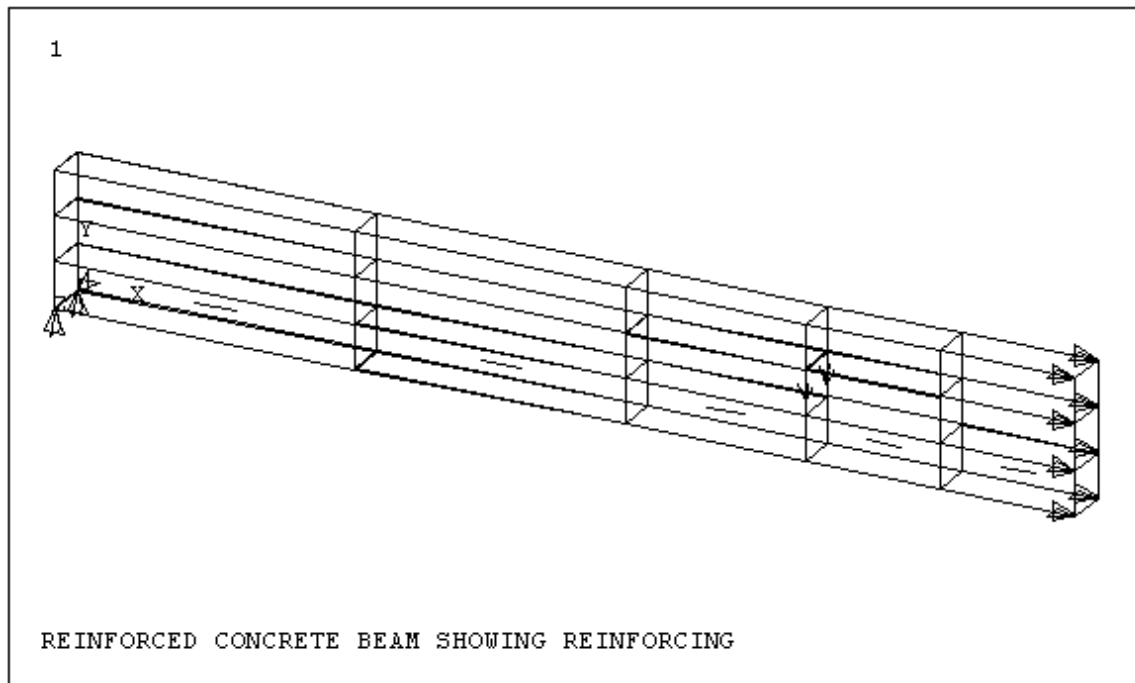
11.3.1.2. Displaying Only the Edges of an Object

While working with displays, you might want to see only the edges of an object; that is, you might want to remove element outlines from the interior of the object. To see only the edges of non-contour displays (**EPLOT**), issue **/EDGE**, ,1 (**Utility Menu> PlotCtrls> Style> Edge Options**). On contour displays (**PLESOL**, **PLETAB**, **PLNSOL**, **PLTRAC**), edges are displayed by default (**/EDGE**, ,0).

11.3.1.3. Displaying the Interior Element Edges of an Object

While working with displays, you might prefer to see the interior element edges, or detail, of an object. If you are working with non-contour displays (**EPLOT**), the interior element edges are displayed by default (**/EDGE**, ,0). To see the interior element edges of contour displays (**PLESOL**, **PLETAB**, **PLNSOL**, **PLTRAC**), issue **/EDGE**, ,1.

An edge, as used in the above context, is the common line between adjacent faces that are not coplanar. The *ANGLE* field on the **/EDGE** command allows you to specify the "degree of coplanarity" at which an edge should be displayed. That is, if *ANGLE* = 45° (which is the default value), an edge is displayed only if the two adjacent faces deviate from coplanarity by more than 45°. If *ANGLE* = 0°, even the slightest deviation from coplanarity causes the edge to be displayed. The default value of 45° is particularly helpful in displaying a cylindrical shell model as a smooth cylinder rather than as a "faceted" cylinder.

11.3.1.4. Using Dashed Element Outlines

You can switch the style of element outlines from solid line to dashed line by using the **/GLINE** command (**Utility Menu> PlotCtrls> Style> Edge Options**). This command allows you to remove element outlines entirely.

11.3.1.5. Shrinking Entities for Clarity

The **/SHRINK** command (**Utility Menu> PlotCtrls> Style> Size and Shape**) shrinks displayed elements, lines, areas, and volumes by a specified percentage so that adjacent entities are separated for clarity. ANSYS ignores a request to shrink the display when the edge option is active.

11.3.1.6. Changing the Display Aspect Ratio

You can artificially distort your display's geometry in a particular direction with the **/RATIO** command (**Utility Menu> PlotCtrls> Style> Size and Shape**). This can be useful for displaying details within a long, skinny object more clearly.

11.3.1.7. Changing the Number of Facets

Area and volume raster displays are made up of numerous small facets (or polygons). Occasionally, you might want to obtain a more precise representation of your areas or volumes by increasing the number of facets used to create these displays. To switch between two different facet densities, use either of the following:
> **Command(s): /FACET**
> **GUI: Utility Menu> PlotCtrls> Style> Solid Model Facets**

11.3.1.8. Changing Facets for PowerGraphics Displays

When PowerGraphics is enabled, you can display varying degrees of curvature in your model by specifying the number of facets per element edge to be used for element display. Facets are piecewise linear approximations of the actual curve represented by the element face or edge. The greater the number of facets, the smoother the representation of the element surface for element plots.

To specify the number of facets per edge, use one of the following:
> **Command(s): /EFACET**
> **GUI: Utility Menu> PlotCtrls> Style> Size and Shape**
> **Utility Menu> List> Results> Options**
> **Main Menu> General Postproc> Options for Outp**

11.3.1.9. Changing Hidden-Line Options

By default, raster displays will be created as Z-buffered displays. See the description of the **/TYPE** command in the *ANSYS Commands Reference* for other "hidden-line" options. All non-Z-buffered hidden-line options produce the same results in vector displays. For area, volume, and p-element Z-buffered displays, you can further specify the type of surface shading (the "smoothness" of the object) using the **/SHADE** command (**Utility Menu> PlotCtrls> Style> Hidden-Line Options**). Also, you can use the **/GFILE** command to set the resolution of Z-buffered displays that are written to graphics files.

11.3.1.10. Section, Slice, or Capped Displays

To view the interior of a 3-D solid element model, you can use *section* displays, *slice* displays, or *capped* displays. (These are all special versions of hidden-line displays controlled by the **/TYPE** command.) A section display produces an image of a 2-D planar section that is defined by the intersection between your model and the cutting plane (see below for a discussion of cutting planes). A slice display is similar to a section display except the edge lines of the remaining 3-D model are also shown. A capped display produces an image of a 3-D portion of your model with a portion of the model display "cut off" by the cutting plane.

11.3.1.11. Specifying the Cutting Plane

Three types of graphics displays - section, slice, and capped - require a cutting plane. Specify the cutting plane via the **/CPLANE** command (**Utility Menu> PlotCtrls> Style> Hidden-Line Options**), and define the plane as either:

- Normal to the viewing direction and passing through the focus point (default)
- The working plane.

11.3.1.12. Vector Versus Raster Mode

The **/DEVICE** command (or **/SHOW** command) allows you to toggle between vector and raster mode. By default, raster mode is active; that is, polygons are filled with color when they are displayed. This affects area, volume, and element displays, as well as the geometry in postprocessing displays. Vector mode produces "wireframe" displays, which show only the outlines of entities, and which usually take less time to form than do raster displays. To display wireframe outlines for solid model entities only (areas and volumes) when your graphics session is otherwise in raster mode, specify the WIRE option of **/FACET**.

11.3.1.13. Perspective Displays

By default, ANSYS creates a non-perspective display of your model. To cause a perspective display to be formed, use the **/VCONE** command (**Utility Menu> PlotCtrls> View Settings> Perspective View**) to define a view cone angle. (The larger the view cone angle, the more pronounced the perspective effect will be.)

11.3.2. Applying Styles to Enhance the Model Appearance

Often, you will want to highlight portions of your model in order to provide a clearer representation of its structure or to highlight certain areas. You can use the following techniques (found under **Utility Menu> PlotCtrls> Style**) to enhance and clarify your model.

11.3.2.1. Applying Textures to Selected Items

You can use textures (**Utility Menu> PlotCtrls> Style> Texturing**) to add realistic effects and differentiate between various items in your model. You must be using a 3-D, Open GL display device, with the appropriate graphics driver loaded. You can apply textures to numbered entities by specifying them on the command line, or you can use graphical picking to select the desired items in your graphics window. Textures are controlled via the **/TXTRE** command.

Textures can affect the speed of many of your display operations. You can increase the speed by temporarily turning the textures off (**Utility Menu> PlotCtrls> Style> Texturing> Display Texturing**). This menu selection toggles your textures on and off. When textures are toggled off, all of the texture information is retained so that it can be reapplied when texturing is toggled back on.

The **/TXTRE** command can be used to apply bitmaps on 2-D devices. Other applications of this command require 3-D capability.

Some 3-D effects will not display properly unless the triangle strip display method is disabled. Tri-stripping provides faster resolution of 3-D displays, and is on by default. You can control tri-stripping via the TRIS option of the **/DV3D** command. Be sure to reapply the TRIS option after you obtain a satisfactory output.

11.3.2.2. Creating Translucent Displays

On some 2-D and 3-D devices, you can create see-through, translucent images by using the **/TRLCY** command (**Utility Menu> PlotCtrls> Style> Translucency**). You can specify the entities to be made translucent either by picking, or by entering the appropriate entity numbers in a fill-in box. The level of translucency can range from opaque to fully transparent.

On 2-D devices, ANSYS displays only the visible faces of the selected items. Using a small value for the **/SHRINK** command (**Utility Menu> PlotCtrls> Style> Size and Shape**) will force the hardware to plot the hidden faces and produce the desired effect.

11.3.2.3. Changing Light-Source Shading

Light-source shading will enhance raster displays on 2-D and 3-D devices having at least eight color planes (2^8 = 256 colors). To specify the number of color planes necessary for light-source shading, use one of these methods:

> **Command(s): /SHOW**
> **GUI: Utility Menu> PlotCtrls> Device Options**

On some 3-D devices, you can adjust the intensity of ambient and directional light, change the light direction, and modify the directional light reflectance factor, using the **/LIGHT** command (**Utility Menu> PlotCtrls> Style> Light Source**). You can also change the light direction for 2-D devices with **/LIGHT** when the Z-buffering hidden-line option is used.

11.3.2.4. Adding Background Shading and Textures

Background treatments help to contrast and highlight your model display, while providing a more pleasing output. Background options are available at **Utility Menu> PlotCtrls> Style> Background**. There are four options that toggle the background on and off and control whether a color, a texture, or a user-specified file is used for the background. The available colors are defined in the **/COLOR** command, and the progression of the gradients (up and down or left and right) can also be specified.

The available textures are defined in the **/TXTRE** command. Depending on the pixel size of your user-specified file, it will either be tiled, fill the entire background, or only a portion will be shown. The texture and file backgrounds place a greater load on graphics speed than the color gradients.

External bitmap files can also be used for your background. You can import a sample from another source to create any desired background. The ***.bmp**, ***.png** and ***.jpg** formats are supported for the PC. Unix systems support ***.png** and ***.jpg**, along with native XWD format. Your imported bitmaps occupy the number of pixels they were generated at, and cannot be resized in the graphics window. Depending on the size (pixels) of the file, the background will be tiled to produce full coverage. Use an external graphics program to obtain the proper size before you import bitmap files.

The default background for the ANSYS graphics window is blue shading with a gradient progression from top to bottom (**/COLOR**,PBAK,ON,1,BLUE). You can modify this using the methods above, or disable it using the **/UIS**,(ON or OFF) option.

11.3.2.5. Using the Create Best Quality Image Capability

Once you have applied the various style attributes to your model, you will want to coordinate their presentation to yield an image that not only conveys the information properly, but also provides a realistic representation. Often, this is a trial-and error procedure, where you apply the various attributes, and then replot to see how the model looks. You can use the "Create Best Quality Image" feature to optimize the use of these effects. You access this feature via **Utility Menu> PlotCtrls> Best Quality Image> Create Best Quality**.

The "Create Best Quality Image" function is a complex macro that takes into consideration the three dimensional nature of your model, the way the various parts of your model are defined, and how the model's attributes should be displayed. It goes through the model's style attributes, and provides the optimal settings. This dialog box, along with a description of each of the controls, is shown in the following description:

Figure 11.3 Create Best Quality Image Function Box
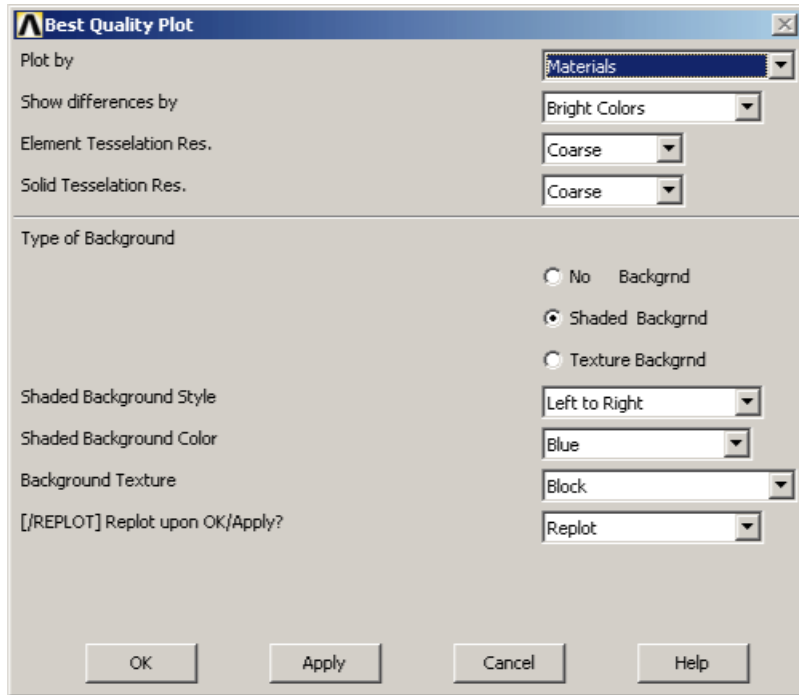
ANSYS Function
Activates the Best Quality Macro. Running this macro optimizes your color palette, light source, translucency and background shading, providing a simple method to arrive at an optimized model representation.

Plot by
Lets you choose the model attribute upon which to base the graphical optimization. The choices are:

- `Materials`: This choice will optimize the model representation according to the defined materials in your model.
- `Type`: This choice will optimize the model representation according to the various element types you have defined in your model.
- `Real`: This choice will optimize the model representation according to the various real constants you have defined for your model.
- `Items (Entities)`: This choice will use the defined areas, volumes and elements to provide the basis for optimization.

Show differences by
The model colors will be applied according to the **Show differences by** selections. A pull-down menu shows the choices for two different color schemes. You can also show the differences according to the textures and translucency styles you have already applied to the model. To modify the color scheme go to **Utility Menu> PlotCtrls> Best Quality Image> Modify Colors**.

Element Tesselation Resolution	Click on this pull-down menu to display three choices for the resolution of your element displays. The "Normal" resolution choice provides an optimized mix of speed and quality, while the "Coarse" or "Fine" selections provide maximum speed or quality, respectively.
Solid Tessellation Resolution	Click on this pull-down menu to display three choices for the resolution of your solid displays. The "Normal" resolution choice provides an optimized mix of speed and quality, while the "Coarse" or "Fine" selections provide maximum speed or quality, respectively.
Type of Background	The background choices are either blank, shaded, or textured. The shaded or textured backgrounds will correspond to the colors found in the **/COLOR** command or the textures found in the **/TXTRE** command. You specify these items in the pull-down menus below the background selections.
Shaded Background Style	This pull-down menu provides four choices for the shading progression of background color.
Shaded Background Color	This pull-down menu allows you to specify the background color according to the color choices listed in the **/COLOR** command.
Background Texture	This pull-down menu allows you to specify the background texture according to the texture choices listed in the **/TXTRE** command.
Replot Upon OK/Apply	Controls the application of replot after applying changes.

Note — The Best Quality Image macro modifies the color map. This can affect the color display on subsequent plots. Once you have captured or plotted the image you should reset the color map by activating either the "Reset to Previous" or "Reset to Global" functions found in the initial Best Quality image menu. (**Utility Menu> PlotCtrls> Best Quality Image**) You must issue a replot for these functions.

11.3.3. Controlling Numbers and Colors

In ANSYS, item numbers and colors are usually related. By default, entities will not be numbered. Numbering (and associated coloring, on appropriate devices) can be turned on and off using the following procedures.

11.3.3.1. Turning Item Numbers On and Off

You can use the **/PNUM** command (**Utility Menu> PlotCtrls> Numbering**) to turn numbering on and off for these items:

- Nodes
- Elements
- Element coordinate systems
- Material types
- Real types
- Element types
- Element locations (for reordered elements)
- Contour values (integer only; on element displays)
- Solid-modeling entities (keypoints, lines, areas, and volumes).

Numbers will not be shown in face hidden-line or precise hidden-line displays.

11.3.3.2. Choosing a Format for the Graphical Display of Numbers

You can select the format in which you want floating point numbers to be displayed by issuing the **/GFORMAT** command **Utility Menu> PlotCtrls> Style> Floating Point Format**. This command lets you indicate the width of the fields in which numbers are displayed and the number of digits that are displayed for a FORTRAN format type. Other commands that let you tailor the appearance of the display include **/PNUM**, **/PBC**, **/PBF**, and **/PSF**.

11.3.3.3. Controlling Number and Color Options

Once you have turned numbering on for an item, you can then use the **/NUMBER** command (which uses the same GUI path as **/PNUM**) to choose among the four possible "on-off" combinations of numbering and coloring (for instance, show colors and numbers (default); show colors, but not numbers; do not show colors, but show numbers; show neither colors nor numbers).

11.3.3.4. Controlling Color Values

You control the correspondence between specific items or numbers and their associated colors using the **/COLOR** command (**Utility Menu> PlotCtrls> Style> Colors>**color option). You can also change the overall color map (edit an existing or store a new color map on a file). See the **/CMAP** command for more information on this capability.

The CMAP utility allows you to change the assignment options for the colors you use, and to save different assignment protocols in separate files that you can load later. This is especially useful for generating specialized contour plots and intricate component and assembly structures.

When you use the CMAP utility, you should close any other ANSYS window, especially the Annotation, Pan Zoom Rotate, Working Plane and Picker windows.

11.3.4. Displaying Loads and Other Special Symbols

The following sections describe how to manipulate loads and other special symbols.

11.3.4.1. Turning Load Symbols and Contours On and Off

To turn load symbols on or off for degree of freedom constraints and concentrated loads, use the **/PBC** command (**Utility Menu> PlotCtrls> Symbols**). **/PBC** controls both solid-model and finite-element load symbols.

For surface loads symbols or contours, use the **/PSF** command (**Utility Menu> PlotCtrls> Symbols**). **/PSF** activates "immediate" display of surface loads on finite elements, but does not activate "immediate" surface load display on solid model entities.)

For body force load contours, use **/PBF** (**Utility Menu> PlotCtrls> Symbols**). **/PBF** applies to finite-element loads only; body force symbols do not appear in solid model displays. **/PBF** does not produce an "immediate" display.

You can also use the **/PBF** command to display your current density magnitude as a vector instead of a contour. **/PBF**, JS, 2 will display the current density magnitude as vector arrows along the surface. The length of the arrows is proportional to the current density magnitude.

You will typically use the above commands to turn load symbols on for visual verification when you apply the loads in SOLUTION (or PREP7). The ANSYS program automatically turns these symbols off when you enter POST1.

See Chapter 12, "Creating Geometric Results Displays" for more information on controlling postprocessing displays.

11.3.4.2. Displaying Boundary Condition Values Next to a Symbol

You can display load symbols by using the **/PBC** command. (See Section 11.3.4.5: Turning Other Symbols On and Off for information on turning other symbols on and off.) This command also provides an option that lets you display the boundary condition values next to the symbols. Some of the boundary condition values that are associated with this command include reaction force (RFOR), reaction moment (RMOM), displacement (U), and current flow (AMPS).

Since your applied forces/moments can differ by orders of magnitude from your derived forces/moments, you can use the FBCS option of the **/PSYMB** command to determine the basis of the Force Boundary Conditions Scaling of your display.

See the *ANSYS Commands Reference* for more information about the various boundary values that are supported.

11.3.4.3. Displaying Boundary Condition Symbols for Hidden Surfaces

When there are hidden surfaces, 2-D drivers will display your boundary condition symbols, yielding a confusing display. In some instances, however, you may desire to see them. You can use the **/HBC** command to control BC symbol display. The default setting is to NOT display boundary condition symbols on the hidden surfaces (**/HBC**, WN, OFF). You can set the display ON or OFF individually for each window of your display. 3-D are not controlled by this command. This function is accessed from the **Utility Menu> PlotCtrls> Style> Hidden Line Options** area of the GUI.

11.3.4.4. Scaling Vector Load Symbols

/VSCALE (**Utility Menu> PlotCtrls> Style> Vector Arrow Scaling**) allows you to adjust the scale of vector item symbols (such as the arrows representing concentrated forces, etc.). This same command also allows you to choose a "uniform scaling" option, in which all items' vector symbols have the same length, regardless of their relative magnitudes.

11.3.4.5. Turning Other Symbols On and Off

You can turn symbols for master degrees of freedom, coupled nodes, and nodes in constraint equations on and off with the **/PBC** command. Use the **/PSYMB** command (**Utility Menu> PlotCtrls> Symbols**) to turn symbols on and off for local, nodal, and element coordinate systems, line directions, keypoints/nodes, and layer orientation (for layered elements).

Chapter 12: Creating Geometric Results Displays

12.1. Using the GUI to Display Geometric Results

In a *geometric results display*, you can review your solution results in a postprocessing display of your model's elements. The choice of geometric results displays includes displaced shapes, results contours (including line-element "contours," such as moment diagrams), and vector (arrow) results (such as thermal flux vector displays). These displays are available only within the general postprocessor, POST1. Figure 12.1: "Contour Results Plot" illustrates a typical geometric results display.

Figure 12.1 Contour Results Plot
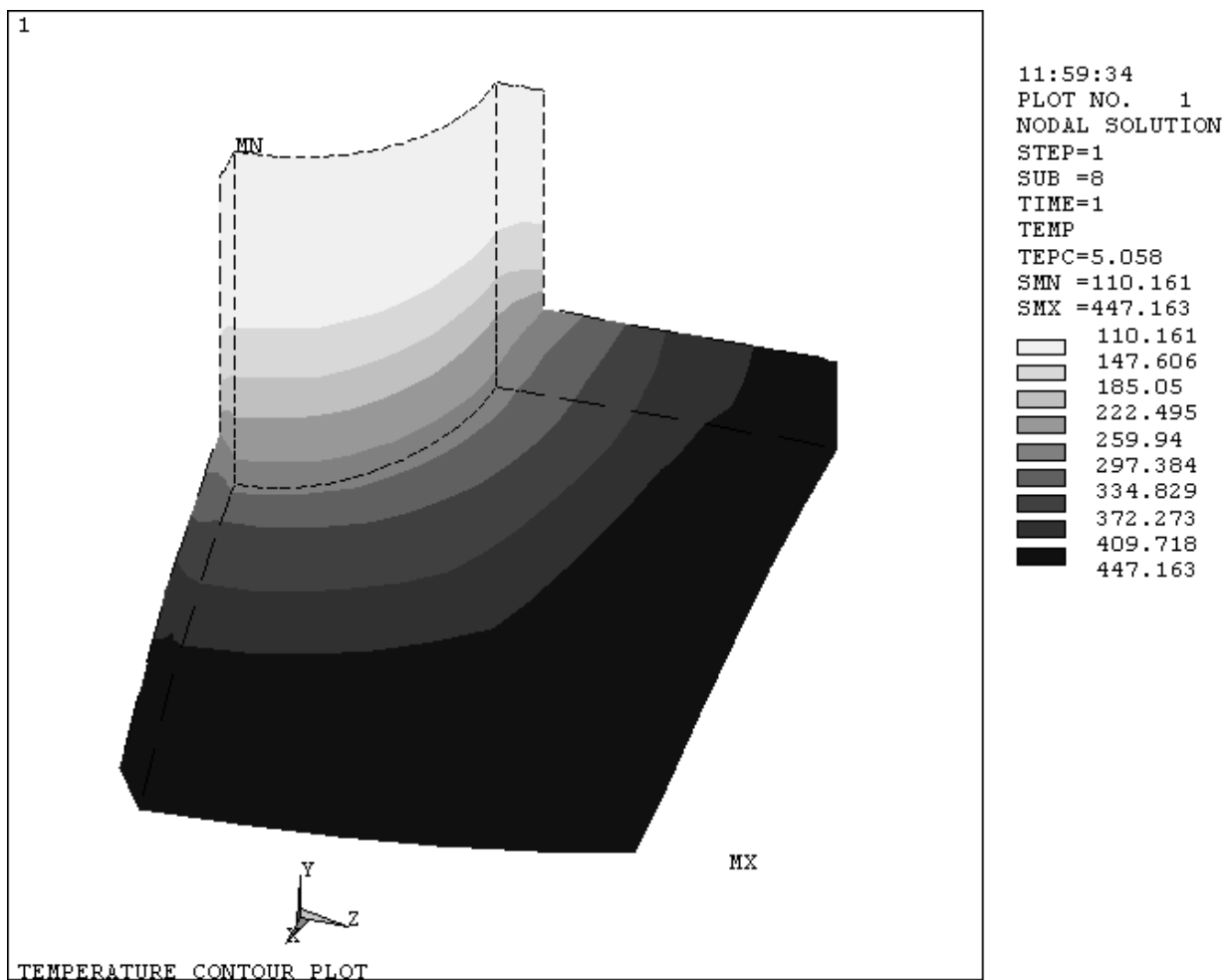
The most convenient way to create and control geometric results displays is by using the functions available under **Utility Menu> Plot** and **Utility Menu> PlotCtrls**. Alternatively, you can use graphics action and control commands, as described in the following subsections.

12.2. Options for Creating Geometric Results Displays

The following commands create geometric results displays in POST1:

Table 12.1 Commands for Creating Geometric Results Displays

Command	GUI Menu Path	Purpose
PLDISP	**Main Menu> General Postproc> Plot Results> Deformed Shape** **Utility Menu> Plot> Results> Deformed Shape**	Display displaced shapes
PLESOL	**Main Menu> General Postproc> Plot Results> Contour Plot> Element Solu** **Utility Menu> Plot> Results> Contour Plot> Elem Solution**	Display contours of results, discontinuous across element boundaries
PLETAB	**Main Menu> General Postproc> Element Table> Plot Elem Table** **Main Menu> General Postproc> Plot Results> Contour Plot> Elem Table** **Utility Menu> Plot> Results> Contour Plot> Elem Table Data**	Display contours of element table data
PLLS	**Main Menu> General Postproc> Plot Results> Contour Plot> Line Elem Res**	Display element table items along line elements and 2-D axisymmetric shell elements
PLNSOL	**Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu** **Utility Menu> Plot> Results> Contour Plot> Nodal Solution**	Display continuous results contours
PLTRAC	**Main Menu> General Postproc> Plot Results> Flow Tra** **Utility Menu> Plot> Results> Flow Trace** **Main Menu> General Postproc> Plot Results> Particle Trace** **Utility Menu> PlotCtrls> Animate> Particle Flow**	Display particle flow or charged particle trace
PLVECT	**Main Menu> General Postproc> Plot Results> Vector Plot> Predefined** **Main Menu> General Postproc> Plot Results> Vector Plot> User-defined** **Utility Menu> Plot> Results> Vector Plot**	Display solution results as vectors
/REPLOT	**Utility Menu> Plot> Replot**	Re-executes the last display action that executed

In Figure 12.2: "A Typical ANSYS Results Plot", a typical geometric results display (in this example, created with a **PLNSOL** command) illustrates the kinds of information included in such displays.

Figure 12.2 A Typical ANSYS Results Plot

Window number

Location of maximum (MX) and minimum (MN) values of displayed item

Structural models are displayed in their displaced configuration

1

MX

MN
Y
Z X

Global triad

ANSYS 5.6
MAY 24 1999
14:59:00
PLOT NO.   1
NODAL SOLUTION
STEP=1
SUB =1
TIME=1
SX   (AVG)
RSYS=0
DMX =0.013267
SMN =-16448
SMNB=-17551
SMX =20444
SMXB=23086
-16448
-12349
-8249
-4150
-51.27
4048
8147
12246
16345
20444

Legend Column (see below)

Typical ANSYS Results Plot ◄──────── Title

ANSYS 5.6 ──────── ANSYS revision number
MAY 24 1999 ──────── Date of display
14:59:00 ──────── Time of display
PLOT NO.   1 ──────── Sequential plot number on plot file (optional)
NODAL SOLUTION ──────── Type of plot
STEP=1 ──────── Load step number (optional)
SUB =1 ──────── Substep number (optional)
TIME=1 ──────── Time value (optional)
SX   (AVG) ──────── Label of plotted item (SX)
RSYS=0 ──────── Active results coordinate system
DMX =0.013267 ──────── Maximum displacement
SMN =-16448 ──────── Minimum value of plotted item
SMNB=-17551 ──────── Minimum bounded value of plotted item (based on error estimate)
SMX =20444 ──────── Maximum value of plotted item
SMXB=23086 ──────── Maximum bounded value of plotted item (based on error estimate)

-16448
-12349
-8249
-4150
-51.27
4048
8147
12246
16345
20444
──────── Contour legend
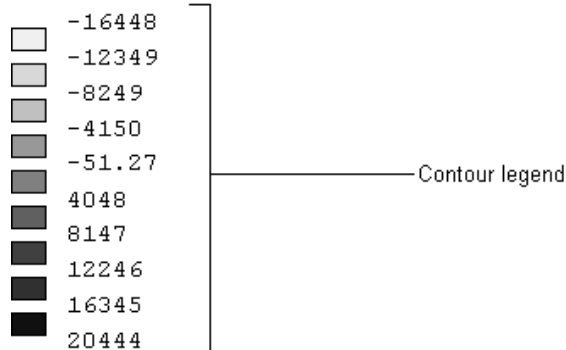
12.3. Changing the Specifications for POST1 Results Displays

Besides reading about the features listed below, also see Chapter 8, "Getting Started with Graphics" for general graphics specifications that you can apply to any kind of display, including geometric results displays.

12.3.1. Controlling Displaced Shape Displays

You can control displaced shape displays in two ways:

- *By superimposing undisplaced and displaced shapes.* A display of a structure's displaced shape will often be more meaningful if you can compare the displaced configuration against the original configuration. You can do this by using the *KUND* argument on the **PLDISP** command.

- *By multiplying displacements for distortion displays.* In most small-deformation structural analyses, the displaced shape is hard to distinguish from the undisplaced shape. The program automatically multiplies the displacements in your results display, so that their effect will be more readily apparent. You can adjust this multiplication factor, using the **/DSCALE** command (**Utility Menu> PlotCtrls> Style> Displacement Scaling**). The program interprets exactly zero values of this multiplier (*DMULT* = 0) as the default setting, which causes the displacements to be scaled automatically to a readily discernible value. Thus, to obtain "zero" displacements (that is, an undistorted display) you must set *DMULT* = OFF.

12.3.2. Controlling Vector Symbols in Your Results Display

You have two options for controlling vector symbols:

- *Displaying nodal or reaction force symbols.* You can add arrow symbols representing nodal and reaction forces (and moments) to your results display with the **/PBC** command (**Utility Menu> PlotCtrls> Symbols**).

- Vector length scaling. You can control the lengths of vector symbols (such as are displayed by **PLVECT** or **/PBC**) with either of the following:
    Command(s): /VSCALE
    GUI: Utility Menu> PlotCtrls> Style> Vector Arrow Scaling

12.3.3. Controlling Contour Displays

When light-source shading is on, the colors shown in the contour legend will not exactly match the contour colors used in the shaded model display. You can manipulate contour displays in the following ways:

- *Labeling contours.* In both vector and raster mode, your contours will always be automatically color-coded. In vector mode, you can add alphabetic contour labels (and a contour legend), using the **/CLABEL** command (**Utility Menu> PlotCtrls> Style> Contours> Contour Labeling**). In raster mode, **/CLABEL** will add (or remove) the contour legend.

- *Controlling the contour legend.* Sometimes, lengthy text in the legend column can cause part of the contour legend to be truncated. You can make more room available for the contour legend by issuing **/PLOPTS**,LEG1,0 (**Utility Menu> PlotCtrls> Window Controls> Window Options**). To remove the contour legend from the legend column, issue **/PLOPTS**,LEG3,0.

- *Changing the number of contour labels.* In vector mode, if you apply contour labels, they will, by default, appear in every element crossed by a contour line. You can use **/CLABEL** to control the number of alphabetic contour labels per element.

- *Changing contour colors.* To change the contour colors used in your display, create a new color-map file and read the new color-map file using one of the following:
    Command(s): /CMAP

To restore color to contours that are grayed out, issue the command **/NUMBER**,0.

· *Changing isosurface colors.* Using the ISURF label in the **/COLOR** command (**Utility Menu> PlotCtrls> Style> Colors>**color type) enables you to change isosurface colors.

· *"Inverting" (or reversing) the contour colors.* By default, the ANSYS program displays the algebraically greatest results values with a bright red contour color, and the algebraically lowest values, with a blue contour color. In some cases, you may want to invert this order. You can create a reversed color-map file by using the CREATE option of the **/CMAP** command. You can then read that reversed color map file into the database, also with the **/CMAP** command.

· *Changing the contour interval.* To change the contour interval on your results display, either issue the **/CVAL** command (**Utility Menu> PlotCtrls> Style> Contours> Nonuniform Contours**) or the **/CONTOUR** command (**Utility Menu> PlotCtrls> Style> Contours> Uniform Contours**).

These commands change the range of values displayed in contour displays. **/CONTOUR** produces uniform contour intervals, while **/CVAL** produces specified contour values (which need not be uniform). If you issue both commands, the program uses the last one specified. For related information, see Section Section 12.3.4: Changing the Number of Contours.

· *Topographic contour displays.* You can transform "flat" contour results displays into "3-D" topographic displays with the **/SSCALE** command (**Utility Menu> PlotCtrls> Style> Contours> Contour Style**).

· *Displaying numerical results values.* To display results values at each node in a contour display, issue **/PNUM**,SVAL,1 (choose **Utility Menu> PlotCtrls> Numbering**).

· *Turning "MN" and "MX" symbols on and off.* The MN and MX symbols identify the locations of the minimum and maximum contour values. The MINM label on the **/PLOPTS** command enables you to turn these symbols on and off.

· *Producing 3-D isosurface, particle gradient, or gradient triad displays.* Isosurfaces, particle clouds, and gradient triads are tools that can help you visualize the state of response within a 3-D solid body. By issuing the **/CTYPE** command (**Utility Menu> PlotCtrls> Style> Contours> Contour Style**), you can change your contour displays to one of these three styles of display.

12.3.4. Changing the Number of Contours

By default, the ANSYS program displays nine contours. To decrease (but not increase) the number of contours, you can either issue the **/CVAL** command (**Utility Menu> PlotCtrls> Style> Contours> Nonuniform Contours**). To change (increase or decrease) the number of contours, you can issue the **/CONTOUR** command (**Utility Menu> PlotCtrls> Style> Contours> Uniform Contours**). However, one or more of the following factors can prevent ANSYS from displaying more than nine contours:

· The device name.

· Whether the display is directed to the screen or to a file.

· The display mode (vector or raster).

· The number of color planes.

Any of these factors can override the number of contours you specify via **/CONTOUR**. You control these factors using either the **/SHOW** command (**Utility Menu> PlotCtrls> Device Options**). In any case, the maximum number of contours available is 128.

The paragraphs below explain how device name, display mode, etc. limit the number of contours available to you:

Driver	Contour Display
The X11 driver (screen display) and raster mode	You can display a maximum of nine contours, no matter how many contours the **/CONTOUR** command specifies.
The X11 driver (screen display) and raster mode	You can display more than nine contours, but the number of contours displayed will be rounded down to the next lowest multiple of nine. For example, if you specify 20 contours, the program displays only 18 contours. In addition, if you specify more than nine contours, contour colors will not be unique (that is, you might have two or more adjacent contour lines with the same color).
The X11C driver (screen display) in either vector or raster mode	If eight graphic planes are available, you can specify any number of contours, up to 128. If your display device does not support eight graphic planes, you are limited to displaying nine contours. If another process has used some of the colors, making fewer than eight graphic planes available, you cannot display more than nine contours. (To verify how many graphic planes are available, issue the **/PSTATUS** command after a plot command.) To make more graphic planes available, you must exit from the ANSYS program, re-enter, and then issue the **/SHOW**,X11C-FORC to force selection of the full set of eight graphic planes.
Plotting to an ANSYS neutral graphics file	Nine contours are the maximum, unless you specify the contour range (using $VMIN$ and $VMAX$ in the **/CONTOUR** command), or unless you explicitly set $NCPL$ to 8 on the **/SHOW** command).

Note — If the current ANSYS graphics are *not* displayed as Multi-Plots (**Utility Menu> Plot> Multi-Plots**), then the following is true:

If the current device is a 3-D device [**/SHOW**,3D], the model contours in all active windows will be the same, even if separate **/CONTOUR** commands are issued for each active window.

For efficiency, ANSYS 3-D graphics logic maintains a single data structure (segment), which contains precisely one set of contours. The program displays the same segment in all windows. The view settings of each window constitute the only differences in the contour plots in the active windows.

12.4. Q-Slice Techniques

Q-slicing is a technique you can use to query the interior or your model via slice planes. To implement Q-slicing, change the hidden surface type to Q-slice using either of these methods:

> **Command(s): /TYPE**,1,8
> **GUI: Utility Menu> PlotCtrls> Style> Hidden-Line Options**

By default, the slice plane is perpendicular to the view and is positioned at the focus point. You can set the slice plane via the GUI path shown above or by using the **/CPLANE**,1 command.

To position the working plane, you can use either of these methods:

- Choose **Utility Menu> WorkPlane> Align WP with> Keypoints**.

- Click on the dynamic mode button in the Offset WP menu. To access this menu, choose **Utility Menu> WorkPlane> Offset WP by Increments**.

You can animate Q-slices. To do so, choose either of these GUI paths:

GUI:

> **Utility Menu> PlotCtrls> Animate> Q-Slice Contours**
> **Utility Menu> PlotCtrls> Animate> Q-Slice Vectors**

12.5. Isosurface Techniques

Isosurface displays are surfaces of constant values (for example, stress). To obtain an isosurface display of von Mises stress, perform these steps:

1. Issue the command **/CTYPE**,1 (**Utility Menu> PlotCtrls> Style> Contours> Contour Style**).

2. Issue the command **PLNSOL**,S,EQV (**Main Menu> General Postproc> Plot Results> Contour Plot> Nodal Solu**).

You can animate isosurfaces. To do so, either invoke the **ANISOS** macro (**Utility Menu> PlotCtrls> Animate> Isosurfaces**).

12.6. Controlling Particle Flow or Charged Particle Trace Displays

You can produce graphic displays of how a particle travels in a flowing fluid or how a charged particle travels in an electric or magnetic field. See The General Postprocessor (POST1) for more information on graphic displays and see Chapter 15, "Animation" for information on particle trace animation. See the *ANSYS, Inc. Theory Reference* for simplifying assumptions on electromagnetic particle tracing.

To produce particle flow or charged particle trace displays, use either of the following:
> **Command(s): PLTRAC**
> **GUI: Main Menu> General Postproc> Plot Results> Plot Flow Tra**
> **Utility Menu> Plot> Results> Flow Trace**
> **Main Menu> General Postproc> Plot Results> Particle Trace**
> **Utility Menu> PlotCtrls> Animate> Particle Flow**

Such displays require you to select the trace points by number or by picking. To select points, use either method shown below:
> **Command(s): TRPOIN**
> **GUI: Main Menu> General Postproc> Plot Results> Flow Trace> Defi Trace Pt**

You can list or delete these points using the commands shown below:

Command(s):
> **TRPLIS**

GUI:
> **Main Menu> General Postproc> Plot Results> Flow Trace> List Trace Pt**

Command(s):
> **TRPDEL**

GUI:
> **Main Menu> General Postproc> Plot Results> Flow Trace> Dele Trace Pt**

Use either of the following to animate the particle flow or charged particle trace to a specified elapsed time.
> **Command(s): TRTIME**
> **GUI: Main Menu> General Postproc> Plot Results> Flow Trace> Time Interval**

Chapter 13: Creating Graphs

13.1. Using the GUI to Create and Control Graphs

If you want to review your material property curves, trace the time-history response of your system, or examine the relationship between any two items in your analysis, you can often do so most effectively using a graph. ANSYS graphs can be either 2-D (X-Y) or 3-D (X-Y-Z, where Z must always be *TIME*). Figure 13.1: "Typical ANSYS Graphs" illustrates some typical graphs.

Figure 13.1 Typical ANSYS Graphs
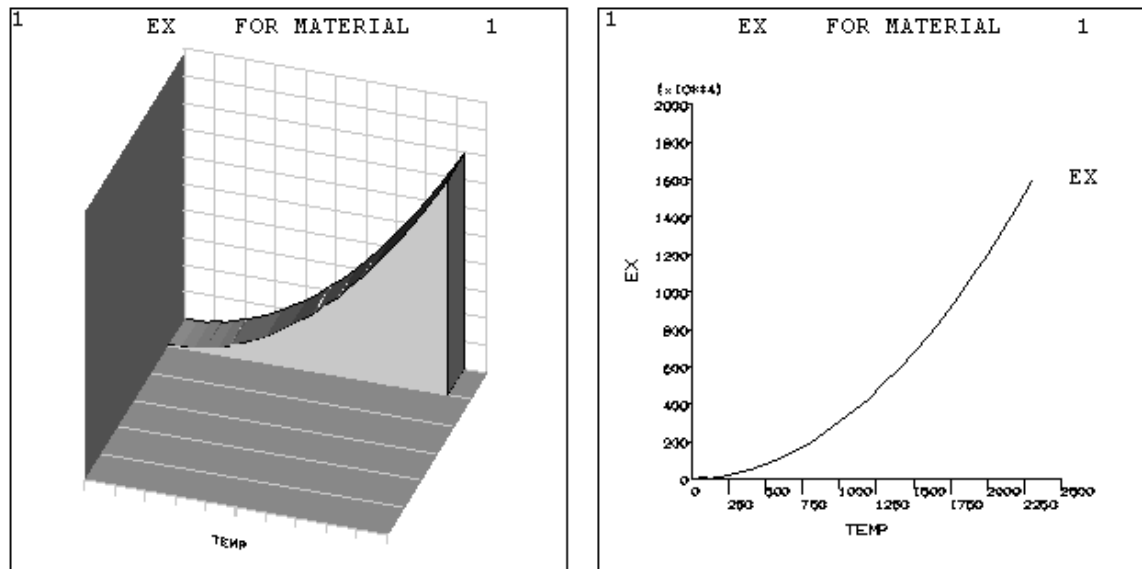
The most convenient way to create and control graph displays is by using the functions available under **Utility Menu> Plot** and **Utility Menu> PlotCtrls**. Alternatively, you can use graphics action and control commands, as described in the following subsections.

13.2. Graph Display Actions

The commands listed below create graphs anywhere in the ANSYS program (including the BEGIN level):

To display linear material properties (those defined with the **MP** family of commands) as a function of temperature, use the following:
>   **Command(s): MPPLOT**
>   **GUI: Utility Menu> Plot> Materials**

To display nonlinear data curves (stress-strain, B-H curve, etc. defined with the **TB** family of commands), use one of the following:
>   **Command(s): TBPLOT**
>   **GUI: Utility Menu> Plot> Data Tables**

To display column vectors of array parameters, use one of the following:
>   **Command(s): *VPLOT**
>   **GUI: Utility Menu> Plot> Array Parameters**

The commands listed below create graphs in POST1 only:

To display a stress item associated with a particular location and event versus loading number (for use in fatigue analyses), use one of the following:

> **Command(s): FSPLOT**
> **GUI: Main Menu> General Postproc> Fatigue> Store Stresses> Plot Stresses**

To calculate and graph path items versus path length, choose one of these methods:

> **Command(s): PLPATH**
> **GUI: Main Menu> General Postproc> Path Operations> Plot Path Item**
> **Main Menu> General Postproc> Plot Results> Plot Path Item**
> **Utility Menu> Plot> Results> Path Plot**

To calculate and graph the membrane and membrane plus linearized stresses along a path, use one of these methods:

> **Command(s): PLSECT**
> **GUI: Main Menu> General Postproc> Path Operations> Linearized Strs**
> **Main Menu> General Postproc> Plot Results> Plot Path Item> Lineariz Strs**

The **PLVAR** command (**Main Menu> TimeHist Postpro> Graph Variables**) graphs any predefined variable as a function of *TIME* (or, for harmonic response analyses, frequency) or some other variable that you define. This command is available in the time-history postprocessor, POST26. A similar **PLVAROPT** command (**Main Menu> Design Opt> Graphs/Tables**) is available in OPT, the design optimization processor.

Issue the **/REPLOT** command (**Utility Menu> Plot> Replot**) to re-execute the last display action command that was executed.

13.3. Changing the Specifications for Graph Displays

In addition to reading about the features listed below, also see Chapter 8, "Getting Started with Graphics" for general graphics specifications that apply to any type of display, including graphs.

13.3.1. Changing the Type, Style, and Color of Your Graph Display

You can alter the appearance of your graph display as follows:

Turning axis divisions (tick marks) on or off. You can control this feature using the AXDV label on the **/GROPT** command (**Utility Menu> PlotCtrls> Style> Graphs**).

Turning axis scale numbers on or off. The AXNM label on the **/GROPT** command controls whether or not your axis scale numbers appear.

Changing the size of axis scale numbers. You can enlarge or reduce the axis scale numbers, using the AXNSC label (and the *KEY* field) on the **/GROPT** command.

Changing the number of significant digits used in axis scale numbers. Axis values will, by default, display four significant digits before the decimal point, and three significant digits after the decimal point. You can change these values with the DIG1 and DIG2 labels on the **/GROPT** command.

Switching between log and linear scales. By default, your graphs will use linear scales. You can switch to log scales on the X and Y axes, using the LOGX and LOGY labels on the **/GROPT** command. (X and Y axes can be switched independently of each other; Z is always linear.)

Establishing separate Y-axis scales for different curves. If you want to graph two or more different items on one display, you might find that the numerical values of the different graphed items differ so significantly that no meaningful information can be obtained from some of the curves. An example would be a time-history graph

of an applied force (with magnitude $\sim10^3$) superimposed over a time-history graph of a resulting deflection (with magnitude $\sim10^{-1}$). The deflection curve would appear to be a straight line if plotted to the same scale as the applied force.

To solve this problem, use different Y-axis scales for each curve. You can activate such a feature with the **/GRTYP** command (**Utility Menu> PlotCtrls> Style> Graphs**). **/GRTYP**,2 displays up to three separate 2-D curves, while **/GRTYP**,3 displays up to six separate 3-D curves. You must also make sure that automatic Y-axis scaling is set to its default value of ON (**/GROPT**,ASCAL,ON) for this feature to work.

Uniform scaling of separate Y axes. If you want to label separate Y-axes distinctly, but want all of them to use the same Y axis scale, you must turn automatic Y-axis scaling off (**/GROPT**,ASCAL,OFF).

Creating "data slice" graph curves (curves that have Z-direction "thickness"). Separately-scaled curves can be separated and given Z-direction thickness with the **/GRTYP**,3 command. (To see this effect, you must change your display's viewing angle and distance - for instance, via **/VIEW**,1,2,2,3 and **/DIST**,1,.88 (**Utility Menu> PlotCtrls> Pan, Zoom, Rotate**). The color-fill option must also be set on via the **/GROPT**,FILL,ON command.)

Setting line thickness for axes, grid lines or graph curve lines. You can accentuate graph items by increasing their line thickness, using the AXIS, GRID, and CURVE labels in the **/GTHK** command (**Utility Menu> Plot Ctrls> Style> Graphs**).

Turning the grid on or off (in the XY plane). You can add a grid to your graph displays, using the **/GRID** command (**Utility Menu> Plot Ctrls> Style> Graphs**). If you add a grid, it can be either a full grid (horizontal *and* vertical grid lines) or a partial grid (horizontal *or* vertical grid lines).

Producing a dashed tolerance curve about the displayed curve. You might want to indicate a range of data spread, tolerance, or uncertainty on your graph curves. You can do so using the **SPREAD** command (**Main Menu> TimeHist Postpro> Settings> Graph**).

Color-filling areas under curves. You can enhance the visual impact of your graph curves by using the FILL label on the **/GROPT** command to fill the areas under the curves with color.

Changing the color of curves (and color-filled areas under curves). The CURVE label on the **/COLOR** command (**Utility Menu> PlotCtrls> Style> Colors> `color type`**) allows you to control the color of each curve in your graph.

Filling the areas under curves with grids. If you have turned on the color-fill option and have also turned on the grid option, then you can cause the grid to appear in the color-filled areas under curves by issuing **/GROPT**,CGRID,ON.

Coloring the XY, XZ, and/or YZ grid planes. The GRBAK label on the **/COLOR** command allows you to control the color of the XY, YZ, and ZX planes.

Coloring the window background. The WBAK label on the **/COLOR** command enables you to control the background color of each window in your display.

13.3.2. Labeling Your Graph

Labeling the axes. You can label the X and Y axes using the **/AXLAB** command.
> **Command(s): /AXLAB**
> **GUI: Utility Menu> PlotCtrls> Style> Graphs**

Labeling the curves. For POST26 plotted-variable graphs, the labels applied to your curves are established when you choose one of the following:

> **Command(s): NSOL**, **ESOL**
> **GUI: Main Menu> TimeHist Postpro> Define Variables**
> **Main Menu> TimeHist Postpro> Elec&Mag> Circuit> Define Variables**

For all other types of graphs, including array parameter (***VPLOT**) curves, the default label will be the item or parameter name specified in the display action command. For these curves, you can use the **/GCOLUMN** command (**Utility Menu> PlotCtrls> Style> Graphs**) to change the curve labels. The **/GCOLUMN** command allows any text or character string to be used as a curve label.

Adding user-defined graphics and text. You can add extra graphics and text to your displays using the annotation functions by choosing **Utility Menu> PlotCtrls> Annotation**. See Chapter 14, "Annotation" of this manual for additional details.

13.3.3. Defining X and Y Variables and Their Ranges

The following subsections detail how to define X and Y variables and their ranges.

13.3.3.1. Defining the X Variable

In POST26 plotted-variable graphs, by default, the program uses *TIME* (or, for harmonic response analyses, frequency) for the X variable. *TIME* does not always have to represent chronological time. In setting up a time-independent analysis, you can arbitrarily define *TIME* to be equal to the value of some other item of interest (such as input pressure). To define a different parameter (other than *TIME*) against which the Y variable is to be displayed, use the **NSOL**, **ESOL**, and **XVAR** commands or their GUI equivalents.

13.3.3.2. Defining the Part of the Complex Variable to Be Displayed

When plotting harmonic-response results in POST26, you need to decide what part of the complex variable (amplitude, phase angle, real part, or imaginary part) to display in your graph. Make your choice using the **PLCPLX** command (**Main Menu> TimeHist Postpro> Settings> Graph**).

13.3.3.3. Defining the Y Variable

The various graphics "action" commands define the Y variable. Sometimes, these commands refer to labels that have been defined in other commands. For instance, **PLPATH** uses labels defined in the **PDEF**, **PVECT**, **PCALC**, **PDOT**, and **PCROSS** commands. **PLVAR** also uses labels defined in the **NSOL** and **ESOL** commands. **PLSECT**, **FSPLOT**, and ***VPLOT**, on the other hand, identify the Y variable directly. (For the GUI equivalents to these commands, see their descriptions in the *ANSYS Commands Reference*.)

13.3.3.4. Setting the X Range

The **/XRANGE** command (**Utility Menu> PlotCtrls> Style> Graph**) enables you to graph only a portion of the full range of X-variable data. This command allows you to "zoom" in or out on a particular segment of your curve.

13.3.3.5. Defining the TIME (or, For Harmonic Response Analyses, Frequency) Range

The **PLTIME** command (**Main Menu> TimeHist Postpro> Settings> Graph**) enables you to establish a range of *TIME* for graph displays. ANSYS always displays *TIME* in the Z-axis direction. If **XVAR** = 1, *TIME* is *also* displayed in the X-axis direction. **PLTIME** or its equivalent then also sets the abscissa scale range. (A range established by **/XRANGE** takes precedence over one defined by **PLTIME**.)

13.3.3.6. Setting the Y Range

By default, your graph will contain the full range of available Y-variable data. Use the **/YRANGE** command (**Utility Menu> PlotCtrls> Style> Graph**) to define a smaller or larger range. The NUM argument allows you to selectively define different ranges for different curves (providing you have established separate Y-axis scales).

Chapter 14: Annotation

14.1. Annotation Overview

A common step in the analysis process is presenting model and results data with additional notations applied - dimensions, comments, highlights, or other text or artwork. You can enhance your standard ANSYS display with a variety of annotation primitives including text, dimensions, polygons, symbols, and even pie charts. (You cannot use the "!" and "$" characters in either 2-D or 3-D ANSYS text annotation.)

ANSYS annotation functions are available for both 2-D and 3-D devices. 3-D annotation can be applied, even if a 2-D device is installed, or a 2-D driver (Win32 or X11) is being used. However, the best results will be obtained when a quality 3-D graphics card is installed, and the appropriate 3-D or Open GL device driver is loaded. The following sections describe 2-D and 3-D annotation capabilities.

14.2. 2-D Annotation

2-D text and graphics annotations are formed as a 2-D overlay on the graphics screen. Because this overlay exists as an imaginary plane, when you transform your model (by changing the scaling, focus, viewing angle, magnification, etc.), your carefully-constructed annotation will not move with the model. Because of this, 2-D annotation should be used primarily for finalized output (reports and printouts) and for representations of the model's state at various stages in the analysis. 3-D annotations will remain anchored to a specific location on the model, and are discussed later in this chapter.

You access 2-D annotation functions through the GUI at **Utility Menu> PlotCtrls> Annotation> Create 2D Annotation**. Every annotation function performed from the GUI places one or more underlying ANSYS command(s) in the log file. This allows you to accurately reproduce the display if the log file is later submitted for batch input. Annotation commands that might appear in such a session log include **/ANNOT**, **/ANUM**, **/TLABEL**, **/LINE**, **/LARC**, **/LSYMBOL**, **/POLYGON**, **/PMORE**, **/PCIRCLE**, **/PWEDGE**, **/TSPEC**, **/PSPEC**, and **/LSPEC**.

The following annotation primitives are available from the 2-D annotation dialog box:

- Text
- Lines
- Rectangles
- Circles
- Arcs
- Polygons
- Wedges
- Arrows
- Dimensions
- Pies
- Symbols

An Options setting is also available. You use the Options setting to copy, move, resize or delete existing annotations.

14.3. Creating Annotations for ANSYS Models

When you choose **Utility Menu> PlotCtrls> Annotation> Create 2D Annotation**, the text annotation dialog box shown below appears. Text annotation can be applied either as stroke text (line-draw characters created within ANSYS) or as bitmap fonts. Bitmap fonts are available on most systems, with the number and type varying

from system to system. Bitmap fonts must be enabled (**Utility Menu> PlotCtrls> Annotation> Enable Bitmap Font**) before the annotation is created. You cannot use the "!" and "$" characters in ANSYS text annotation.

Figure 14.1 Stroke Text Annotation Dialog Box

The fields and buttons presented in the annotation dialog box change when you reset the annotation entity type. For example, if you reset the annotation entity to arcs, the dialog box shown in Figure 14.1: "Stroke Text Annotation Dialog Box", changes to display the options available for annotation arcs (arc color, solid or dashed lines, and arc width). Regardless of which annotation entity you choose, the annotation dialog box always displays four action buttons:

Undo - Erases the last annotation entity created.

Refresh - Redisplays the annotation, which is useful after move and delete operations.

Close - Closes the annotation dialog box.

Help - Displays online help for the dialog of the currently selected annotation entity.

Once you create annotations, you can control their display by selecting **Utility Menu> PlotCtrls> Annotation> Display Annotation**. Accessing this menu pick toggles annotation display on and off.

14.4. 3-D Annotation

3-D text and graphics annotations are assigned XYZ coordinates and exist in 3-D space. When you apply 3-D annotation, you choose from one of the following anchor locations:

- Nodes
- Elements
- Key Points
- Lines
- Areas
- Volumes
- All
- At XYZ
- On View

Because 3-D annotation is applied in relation to the XYZ coordinates of the anchor, you can transform your model, and the annotation will maintain the spatial relationship with the model. This works within reason, and there are instances where changing the perspective or the size of the model will change the apparent relationship between the annotation and the model. The overall 3-D dimensions of your model are defined by a bounding box. If portions of your model's bounding box lie outside of the visible area of your graphics window (if you are zoomed in on a specific area of your model), it can affect the placement of your 3-D annotations. Zooming out will usually overcome this problem. Unlike 2-D annotation, 3-D annotation is valid for the global Cartesian (**CSYS**,0) coordinate system only.

3-D annotation functions are accessed through the GUI at **Utility Menu> PlotCtrls> Annotation> Create 3D Annotation**. Every annotation function performed from the GUI places one or more underlying ANSYS command(s) in the log file. This allows you to accurately reproduce the display if the log file is later submitted for batch input.

The following annotation primitives are available from the 3-D annotation dialog box:

- Text
- Lines
- Areas
- Symbols
- Arrows

An Options setting is also available. You use the Options setting to copy, move, resize or delete existing annotations.

14.5. 3-D Query Annotation

With Query Annotation, you can retrieve model information directly from the database and apply it to the model. The ANSYS Model and Results Query Pickers provide a "Generate 3-D Anno" check box that enables the annotation function. You can obtain basic model information, results data and even simple geometric/loading information (force per unit area, angle between lines, etc.) by graphically picking the desired items. Like standard 3-D Annotation, you use the Options setting to copy, move, resize or delete 3-D Query Annotations. As with standard 3-D Annotation, 3-D Query Annotation is valid for the Cartesian (**CSYS**,0) coordinate system only. See Chapter 5, "Graphical Picking" in the *ANSYS Operations Guide* for more information on Query Annotation.

Chapter 15: Animation

15.1. Animation Overview

Animation is a valuable tool for graphically interpreting many analytical results, including nonlinear or time-dependent behavior. The ANSYS program provides tools that enable you to animate any type of display.

Many workstations, PCs, and some terminals having local segment memory support animation. However, some hardware platforms do not support online animation well (or at all). An alternative to online animation is to capture a sequence of images offline, frame by frame, on film or videotape.

15.2. Creating Animated Displays Within ANSYS

The easiest way to perform animation in ANSYS is to use the functions available under **Utility Menu> PlotCtrls> Animate**. These GUI functions allow you to achieve "push-button animation" effects in ANSYS. The GUI functions internally execute ANSYS animation commands, which you can type in directly if you prefer. Procedures for using commands are discussed next. See Chapter 16, "External Graphics" for information on viewing animated sequences in the stand-alone DISPLAY program.

15.3. Using the Basic Animation Commands

You can display several frames in rapid succession to achieve an animation effect, via these commands:
>**Command(s): /SEG**, **ANIM**
>**GUI: Utility Menu> PlotCtrls> Redirect Plots> Delete Segments**
>**Utility Menu> PlotCtrls> Redirect Plots> Segment Status**
>**Utility Menu> PlotCtrls> Redirect Plots> To Segment Memory**
>(UNIX)
>**Utility Menu> PlotCtrls> Redirect Plots> To Animation File**
>(Windows)
>**Utility Menu> PlotCtrls> Animate> Replay Animation**
>**Utility Menu> PlotCtrls> Animate> Replay Animation**

The **/SEG** command allows you to store graphics data in the terminal's local "segment" (graphics operation) or "pixmap" (screen dot) memory (which may or may not be available, depending on the type of graphics device you are using). The storage occurs at the same time that a graphics action command produces a display. You can then use the **ANIM** command to display the stored frames in a sequence. A typical command stream for animation would look like this:

```
/SEG,DELE    ! Deletes all currently stored segments
/SEG,MULTI       ! Stores subsequent displays in segment memory
...              ! Plot-creation commands to generate a sequence of images
...              ! (See below for options)
/SEG,OFF         ! Turns off the frame-capture function
ANIM,15          ! Cycles through the stored sequence 15 times
```

To create the series of frames for your animation sequence, you can either issue a frame-by-frame series of graphics action commands, or you can invoke a predefined ANSYS macro to automatically generate the sequence. The predefined macros are **ANCNTR**, **ANCUT**, **ANDATA**, **ANDSCL**, **ANFLOW**, **ANISOS**, **ANMODE**, **ANTIME**, and **ANDYNA**.

The available amount of local segment or pixmap memory, and the memory requirements of each frame limit the number of frames you can include in an animated sequence. On most workstations and PCs, the amount of memory required depends on the number of pixels (for example, screen dots) in each frame. On X-window

devices, reducing the size of your graphics window reduces the number of pixels, yielding a longer achievable animation run.

Although you can create animations of multiple ANSYS window schemes, animations created with OpenGL display lists (**/DV3D**, ANIM, 0) do not retain the windowing scheme information. You CAN save multiple windows via the X11/WIN32 drivers, or via the OpenGL driver with **/DV3D**, ANIM, KEY in effect (where KEY is not zero

15.4. Using One-Step Animation Macros

A better alternative to the basic animation commands discussed above is to use specialized "one-step" animation macros.

- **ANDATA** (**Utility Menu> PlotCtrls> Animate> Over Results**) produces a sequential contour animation over a range of results data. This macro allows you to create an animation sequence based on the last plot action command (e.g. **PLDISP**).

- **ANDSCL** (**Utility Menu> PlotCtrls> Animate> Deformed Shape**) produces an animated sequence of a deformed shape in the POST1 postprocessor. Before you use the **ANDSCL** macro, you must execute a display command that contains deformation (such as the **PLDISP** command).

- **ANCNTR** (**Utility Menu> PlotCtrls> Animate> Deformed Results**) produces an animated sequence of a contoured deformed shape in POST. Before using the macro, you need to execute a display command that contains contouring, deformation, or both (such as **PLNSOL**,S,EQV).

- **ANMODE** (**Utility Menu> PlotCtrls> Animate> Mode Shape**) produces an animated sequence of a deformed mode shape in POST1. Before using **ANMODE**, you must execute a command that contains deformation.

- **ANTIME** (**Utility Menu> PlotCtrls> Animate> Over Time**) produces an animated sequence of a contoured deformed shape varying over time in POST1. Before using this macro, you must execute a display command that contains deformation, contouring, or both *and* you must have a solution containing time variance.

- **ANISOS** (**Utility Menu> PlotCtrls> Animate> Isosurfaces**) produces an animated sequence of an isosurface of contoured deformed shape in POST1. Before using **ANISOS**, you must execute a display command that contains contouring.

- **ANCUT** (**Utility Menu> PlotCtrls> Animate> Q-Slice Contours** or **Utility Menu> PlotCtrls> Animate> Q-Slice Vectors**) produces an animated sequence of a cutting plane through a contoured deformed shape in POST1. Before using this macro, you need to execute a display command that contains contouring.

- **ANFLOW** (**Utility Menu> PlotCtrls> Animate> Particle Flow**) produces an animated sequence of particle flow or charged particle motion. Before using this macro, you need to execute a command that produces particle flow trace on an element display (i.e., **PLTRAC**).

- **ANHARM** (**Utility Menu> PlotCtrls> Animate> Time-harmonic**) produces a time-transient animation of time-harmonic results of the last plot action command (e.g. **PLNSOL**,B,SUM). The animation converts the complex solution variables (real and imaginary sets) into time varying results over one period.

ANDYNA, while still supported by ANSYS, has been replaced by the **ANDATA** macro.

15.5. Capturing Animated Display Sequences Off-Line

In this procedure, you produce graphics images one at a time, photographing or video-recording them frame by frame. Among this technique's advantages is the fact that when you capture an animated sequence one frame at a time, there is generally no limit on its complexity, and performance does not degrade with increasing numbers of entities.

In general, producing high-quality graphics video recordings is a job for multimedia experts with specialized equipment. Capturing a sequence of individual frames on video requires three separate pieces of equipment:

- A device that produces a television-style video signal (accomplished through the use of an add-in board, a separate encoder, or a scan converter).

- A frame controller to control the video recorder as it captures the individual frames. The frame controller receives both the television video signal and a computer input (such as serial RS-232), and sends instructions to capture the frames.

- A frame-controllable video recorder (which differs considerably from a home VCR).

In addition to specialized hardware requirements, some custom software is also needed for video recording. The **/SYS** command in ANSYS provides the programming interface between the ANSYS program and these special systems, allowing video system commands to be integrated into your ANSYS session.

Another hardware solution for animation is capturing single frames onto film, using a device known as a film recorder. As with video frame-capture equipment, images are saved onto film under software control. The best of these devices can be expensive, and custom programming may be involved in using them.

A relatively low-cost approach to film recording involves the use of a stationary camera shooting individual frames from a graphics display. These frames are then processed as the individual frames of a film. The resources of photographic technicians are often required to turn still images into acceptable-quality moving film.

15.6. The Stand Alone ANIMATE Program

When you create animations in UNIX, they are stored as ANIM files. This format is not supported outside of ANSYS. You can use the ANIMATE Program (**ANIMATE.exe**) to conveniently play back your ANIM files on the PC. The ANIMATE program runs on the PC, even if you do not have ANSYS installed. You can also use the ANIMATE program to convert your ANIM files to an AVI format. The AVI animation file format is supported by a number of Windows applications, including Windows Media Player. ANIMATE is especially useful for creating portable files that can be exchanged via the internet, since the AVI file format is significantly smaller than the ANIM format.

The ANIMATE program is included with all ANSYS products (UNIX and Windows). The program is located in the **bin\intel** directory of your ANSYS installation CD, and requires no license keys or passwords to install. It provides better frame-speed and window-size control than the standard Windows Media Player, and is small enough (250kb) to be transported or e-mailed with your other analysis files.

15.6.1. Installing the ANIMATE Program

In order to install the ANIMATE program on a PC system that is not running ANSYS, you must ensure that the proper Dynamic Link Library (DLL) files are present, and that the Windows registry has been modified to recognize those DLLs. To do this, copy the files **DSGStreamN.dll**, **ATL.dll** and **ANIMATE.exe** from your installation disk into the same directory on the PC where you want to run ANIMATE. You will find the DLL files on your installation disk and in the **\bin\intel** directory for Windows. The **ANIMATE.exe** file is also located in the **\bin\intel** directory.

Once you have all three files in a common directory, run the **regsvr32.exe** file, from within that directory, for each of the DLL files. You do this by running the file using the specific path, which in most cases is **C:\winnt\system32** (check to ensure that this is the path for your system).

For example, while in the directory containing the DLL files run:

C:\winnt\system32\regsvr32 DSGStreamN.dll

C:\winnt\system32\regsvr32 ATL.dll

15.6.2. Running the ANIMATE Program

In order to convert your UNIX animations, the ANIM files must be transferred to the Windows file system. This can be done using FTP protocols, or with SAMBA or some other file system transfer utility. Once the ANIM files are accessible, they can be opened directly.

The controls provided for the ANIMATE program are nearly identical to those found in the ANSYS animation controller. When you start the program, the panel shown below is displayed.

Figure 15.1 The ANIMATE Program Display
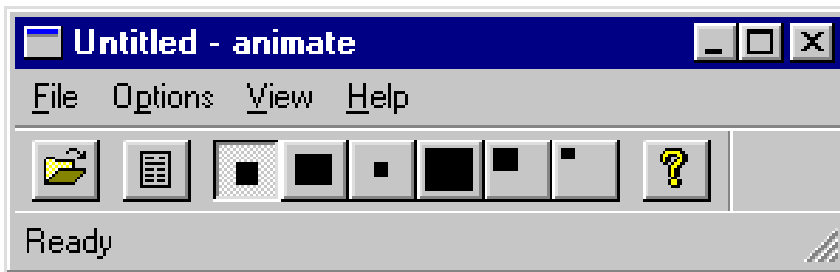
You can access the following operations from the initial program display:

- **File**: Allows you to open AVI or ANIM files and to save these files in True Color AVI or 256 Color format.
- **Options**: Allows you to pop up the ANIMATION CONTROLLER, and to choose from six different screen sizes for the animation window.
- **View**: Allows you to toggle (ON or OFF) the display of the TOOLBAR (icons at the top of the screen) and the display of the STATUS BAR (read out at the bottom).
- **Help**: Displays information about the program.

Once you load an animation file, you can use the ANIMATION CONTROLLER for a number of playback options. The ANIMATION CONTROLLER is shown below:

Figure 15.2 The Animation Controller
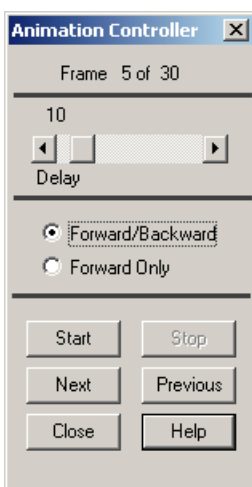
You can access the following operations from the ANIMATION CONTROLLER panel:

Animation Delay   Use the slider to adjust the speed of animation (that is, how quickly the animation progresses from one frame to the next). The higher the delay setting, the slower the animation speed.

Forward/Backward - Forward Only	Allows you to loop the file either by following a forward run with a reverse run, or by playing it to the end and restarting it.
Action Buttons	• **Start - Stop - Next - Previous**: Allows you to play the animation continuously, or to view it frame by frame.
	• **Cancel**: Dismisses the controller panel.

Note — Although you can create animations of multiple ANSYS window schemes, animations created with OpenGL display lists (**/DV3D**, ANIM, 0) do not retain the windowing scheme information. You CAN save multiple windows via the X11/WIN32 drivers, or via the OpenGL driver with **/DV3D**, ANIM, KEY in effect (where KEY is not zero).

15.7. Animation in the Windows Environment

The ANSYS and DISPLAY programs on Windows platforms use the Microsoft standard AVI file format to store animation frames (video only) of ANSYS graphics. The ANSYS program's support for AVI files is described next.

15.7.1. How ANSYS Supports AVI Files

In ANSYS, animation capabilities are split among the options in the **Utility Menu> PlotCtrls** GUI path and the animation macros described earlier in this chapter. If you are animating a deformed shape or different mode shapes of your analysis, the program stores the animation frames in a file called **Jobname.AVI**, where **Jobname** is the jobname for the current ANSYS session. After completing this step, ANSYS starts Media Player (located under Accessories). This application has a control panel that closely resembles the controls of a videocassette player.

If you wish to animate contours, ANSYS displays a dialog box from which you can choose animation options. After you supply this data, ANSYS generates the frames and Media Player displays them.

The Replay animation option starts Media Player. If you have stored an animation sequence during the current ANSYS session, the file name associated with it is supplied to Media Player automatically.

You can animate other quantities, or do animation in other parts of ANSYS, via the **/SEG** command. You can access this command directly through the ANSYS Input Window or **Utility Menu> PlotCtrls> Redirect Plots**.

AVI files cannot be created directly in batch mode. If you are working in batch mode, you must save multiple images to a single **Jobname.GRPH** file using the **/SHOW** command. After the batch run, you can open the resulting **Jobname.GRPH** file in the DISPLAY program and then create an AVI file.

15.7.2. How DISPLAY Supports AVI Files

If you have stored a series of graphics in an ANSYS graphics file, you can create an animation file of these in the DISPLAY program.

Start the DISPLAY program and choose **Display> Animate> Create** on the menu bar. The following dialog box appears.

Figure 15.3 ANSYS DISPLAY Program and the Create Animation Sequence Dialog Box
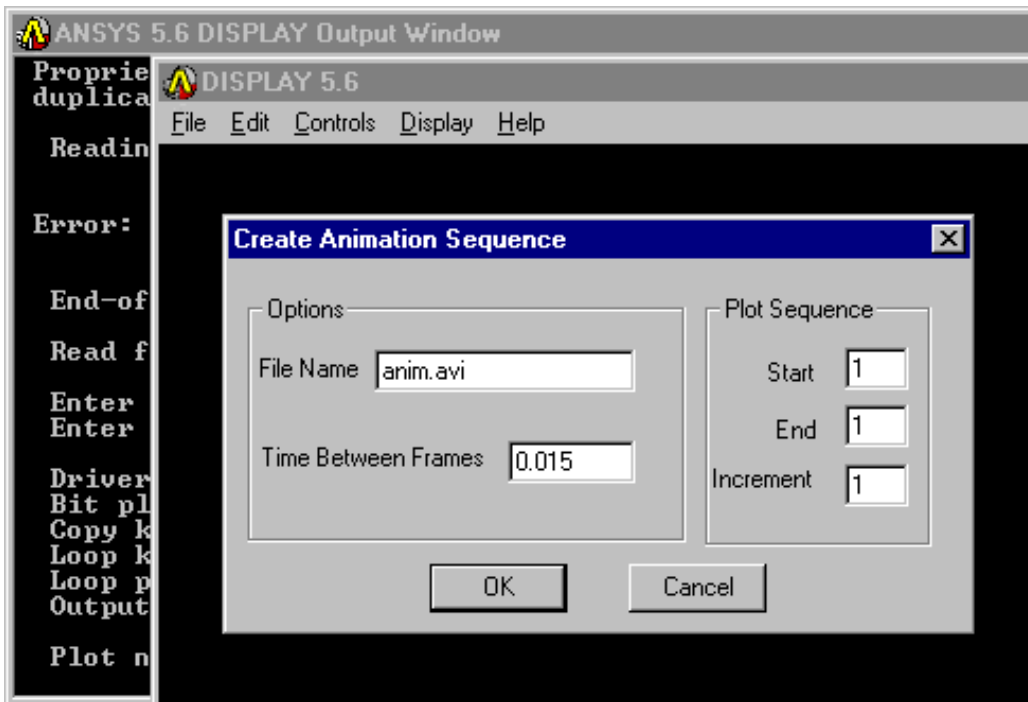
Specify the plots to be used during the animation in the File Name box and the delay time in seconds in the Time Between Frames box. For example, if your **Jobname.GRPH** file contains 20 plots and you wish to use every other plot in your animation, select 1 (for the beginning plot), 20 (for the end plot), and 2 (for the increment). The Create function stores your animated sequence in the default file **ANIM.AVI**.

To replay your animation, use the Playback option, which starts Media Player.

> *Note* — If you are doing animation from an AVI type program on an ANSYS animation file, make sure that the graphics window size of the AVI setting is set to "Original Size." To check the setting for window size, click on the AVI icon and click on SETTINGS. You can change the window size here, if necessary.

15.7.3. Other Things You Can Do with AVI Files

While you are in Media Player, you can use Media Player's OLE (Object Linking and Embedding) to export your ANSYS animation to other applications. You do this through the "Copy" option under the Edit Menu. Then, you can embed the animation in another OLE-compliant application. For example, you can embed ANSYS animation objects in Microsoft Word or Microsoft Excel.

Once an object is embedded on an application, you can just double-click on the object to start playing back your ANSYS animation sequence. To share your compound document with others, give them the **Jobname.AVI** file you created in ANSYS or DISPLAY plus a copy of the file containing the embedded animation sequence.

Chapter 16: External Graphics

16.1. External Graphics Overview

Other chapters in this document discuss how to create and control graphics that you view directly in ANSYS. While in ANSYS, you can export the contents of the graphics window (full screen options are also available for some platforms), either to a printer or to a graphics file.

You can also generate a neutral graphics file (***.GRPH**) and use the stand-alone DISPLAY program. The DISPLAY program can be used to view static or animated screen images, or to convert your file into the appropriate format for printing, plotting, or exporting to word processing or desktop publishing programs.

16.1.1. Printing Graphics in Windows

In ANSYS, you can obtain hard copy output by choosing **Utility Menu> PlotCtrls> Hard Copy**. You then choose to print the contents of the graphics window, or to create an exportable graphics file. When the **To Printer** option is selected, the Windows printer dialog box for the designated printer is displayed. Printing options, including page layout, output resolution and document handling can be modified from this, system-controlled panel.

Printer spooling options are commonly used to free up the processor more quickly (especially in Z-buffered mode). In Type 4 or Polygon mode, spooling may cause some elements to not plot, or to be improperly placed. Select the Print Directly to Printer option when these types of printing problems are encountered.

16.1.2. Exporting Graphics in Windows

Selecting **Utility Menu> PlotCtrls> Hard Copy** and then the **To File** option displays the Graphics Hard Copy dialog box. This box provides a number of popular file export formats (BMP, EPS, JPEG, TIFF and PNG) along with limited page layout and configuration options. These formats allow you to export your output window (or full screen) contents into a large number of commercially available desk top publishing or presentation software applications. ANSYS JPEG software is based in part on the work of the Independent JPEG Group, Copyright 1998, Thomas G. Lane.

The PNG format is an extremely capable, portable file format that is gaining widespread acceptance in many computer applications, including the web. It is a lossless, true-color image format that minimizes the distortion, mottling and pallet limitations found on other formats, while still retaining excellent compression performance. ANSYS creates PNG files with the assistance of the following LIBPNG and ZLIB packages:

LIBPNG version 1.0.5 - October 1999

Copyright 1995, 1996, Guy Eric Schalnat, Group 42, Inc.

Copyright 1996, 1997 Andreas Dilger

Copyright 1998, 1999 Glenn Randers-Pehrson

ZLIB Version 1.1.3

Copyright 1995 - 1998 Jean Loup Gailly and Mark Adler.

You can also create exportable graphics by selecting **Utility Menu> PlotCtrls> Redirect Plots**. In addition to the page layout and configuration options found in the Graphics Hard Copy dialog box, this method provides additional position, resolution and scaling options. When you use this option, the output functions for most of

these formats are controlled from within ANSYS, instead of by the operating system. This selection also provides HPGL (plotter) and VRML 3-D rendering output, along with screen dump and animation options.

You can also export Windows Metafiles directly from ANSYS (Windows systems only) by selecting **Utility Menu> PlotCtrls> Write Metafile**. The subsequent dialog boxes provide limited page layout and configuration options.

Because most of the export methods listed above use the system's video output to generate the file format, they will not work unless you are in interactive mode. If you are not in interactive mode, and you want to export any of the above graphics types, use the **/SHOW** command. This method allows you to generate output files when you run your analysis from a batch file. The Courier and Helvetica font files used for text output within the JPEG, PNG and TIFF batch outputs are not accessed from your operating system. Permission to use these files is granted by Adobe Systems, Inc. and Digital Equipment Corp. in association with these functions only.

16.1.3. Printing Graphics in UNIX

You can print to a post script printer from within ANSYS. Selecting **Utility Menu> PlotCtrls> Hard Copy** displays the PS Hard Copy dialog box. Print options, including page layout, reverse video and grey scale are available from this dialog box.

The Reverse Video option affects only the background of the display. The black background provided in the ANSYS graphics window is often unsuitable for printing. Selecting Reverse Video will present the graphics on a white background. Contour colors and other colors selected from the ANSYS palette are unaffected by this option.

16.1.4. Exporting Graphics in UNIX

The PS Hard Copy dialog box (**Utility Menu> PlotCtrls> Hard Copy**) also provides a number of file export formats (EPS, TIFF and JPEG) along with limited page layout options. These files can be used in various word processing and desktop publishing applications.

As in Windows, you must use the **/SHOW** command in order to generate file exports during batch runs.

To obtain additional export formats, choose **Utility Menu> PlotCtrls> Redirect Plots**. You can select from GRPH, PSCR, HPGL, HPGL2, JPEG, TIFF and VRML. These formats are suitable for a wide range of applications outside of the ANSYS program. Of special interest is the **.GRPH** file, a neutral graphics file that uses the ANSYS plotting instructions to recreate the file in applications other than ANSYS.

16.2. Creating a Neutral Graphics File

The key to external graphics is the *neutral graphics file*. This is an ASCII text file containing the instructions required to produce a graphics display. You can view the displays stored on this file, using the DISPLAY program and the appropriate 2-D graphics driver, on any supported hardware platform. The neutral graphics file is not a bitmap format but an ASCII text format, which means the resolution of a display produced by the DISPLAY program usually will be better than that produced using the **Utility Menu> PlotCtrls> Hard Copy**.

To route your graphics displays to a neutral graphics file having any valid filename, use one of the choices shown below. (In batch mode, by default, the ANSYS program will assign this filename to **Jobname.GRPH**)

> **Command(s): /SHOW**
> **GUI: Utility Menu> PlotCtrls> Device Options**
> **Utility Menu> PlotCtrls> Redirect Plots> To GRPH File**
> **Utility Menu> PlotCtrls> Redirect Plots> To Screen**

Each subsequent graphics action command that you issue writes a separate display to this file. (Thus, a neutral graphics file can contain more than one display, with each display being sequentially numbered, beginning with

1.) You can use the ANSYS animation macros, which automatically generate a series of graphics action commands for animation purposes, to create multiple displays on your neutral graphics file. If you wish, you can reissue the **/SHOW** command with a graphics device name to direct subsequent displays to your terminal screen. This way, you can toggle back and forth between the screen and a file (which is appended, not overwritten) as many times as you wish.

16.3. Viewing and Translating Neutral Graphics Files with the DISPLAY Program

Once you have created a neutral graphics file, you can use the stand-alone DISPLAY program to view static or animated screen images, or to translate your file into the appropriate format for printing, plotting, or exporting to word processing and desktop publishing programs. The DISPLAY program creates images directly by using information from a .GRPH file created in a previous ANSYS session.

DISPLAY supports all UNIX screen devices and printers that the ANSYS program supports. It also supports Windows-compatible screen devices and printers and the following hard copy formats:

- Hewlett-Packard Graphics Language (HPGL*x*)
- PostScript (version 1.0 minimally conforming)
- Metafile Format (WMF or EMF)
- Interleaf ASCII Format (OPS Version 4.0)
- ASCII Text Dump

16.3.1. Getting Started with the DISPLAY Program

The DISPLAY program runs independently from the ANSYS program. From Windows, click on the **Start** button and choose **Programs>ANSYS 8.1> Display Utility**. From a UNIX prompt, issue the command **display81** or **xdisplay81** for a GUI similar to the Windows DISPLAY program. You can specify any or all of the following commands options:

-j	Jobname
-d	Device_Type
-s	Read / Noread

These options function exactly as they do in the ANSYS program. The DISPLAY program does not support the memory (-m), database (-db), batch (-b), ANSYS menu (-g), language (-l), product (-p), version (-v), and parameter specification options.

DISPLAY does support the redirection of standard input and output. For example, in the C (csh) shell, the following statement is valid:

```
display81  -d  X11  -j  demo  <demo.dat>&  demo.out  &
```

To streamline your use of the DISPLAY program during presentations and demonstrations, you might want to create a **start81.dsp** file containing any valid DISPLAY commands that you would want to execute automatically at start-up. (Use an external text editor to create start81.DSP.)

The ANSYS program reads the first **start81.dsp** file it finds in the following search paths:

- the working directory

- your home directory
- the ANSYS apdl directory

If you are running DISPLAY on a Windows System, instead of using a **start81.dsp** file, you can simply select a **file.GRPH** file from File Manager, drag it to the DISPLAY window, and drop it there.

16.3.2. Viewing Static Images on a Terminal Screen

Use the following procedure to view static displays on a screen with the DISPLAY program.

> *Note* — The commands discussed in this section, unless otherwise noted, are DISPLAY commands, *not* ANSYS commands.

1. Set up your DISPLAY session with the **/SHOWDISP** and (if desired) **/CMAP** or **NOCOLOR** commands. (You can include these commands in a **start81.dsp** file.)

2. Using the **FILEDISP** command, direct the DISPLAY program to read the desired neutral graphics file. If you are using the DISPLAY and ANSYS programs simultaneously, make sure the neutral graphics file is first *closed in ANSYS*. That is, issue **/SHOW**,CLOSE (in ANSYS) before reading the file in DISPLAY.

3. Specify terminal options with the **TERM** command. For screen display, you might be interested in setting the **TERM**,LOOP options (the number of loops, *NLOOP*, and the amount of time to pause between displays, *PAUSE*).

4. Issue the **PLOT** command to cause specified displays to be formed. Recall that your graphics file can contain several different displays. You can call up specific displays by number, or you can instruct the program to display ALL plots found on your file.

5. Issue **FINISH** to exit the DISPLAY program.

16.3.3. Viewing Animated Sequences on a Screen

The procedure for creating an animated display in the DISPLAY program is similar to that used in the ANSYS program. By executing **/SEG** and **ANIM** commands, you can display several frames in rapid succession to achieve an "animation" effect. (As in the ANSYS program, with the DISPLAY program you cannot use all hardware platforms to produce online animation.)

For the DISPLAY program, the *Aviname* and *DELAY* arguments of the **/SEG** command are ignored.

GUI menu paths to the **/SEG** and **ANIM** commands are:

GUI:

    Utility Menu> PlotCtrls> Redirect Plots> Delete Segments
    Utility Menu> PlotCtrls> Redirect Plots> Segment Status
    Utility Menu> PlotCtrls> Redirect Plots> To Segment Memory (UNIX)
    Utility Menu> PlotCtrls> Redirect Plots> To Animation File (Windows)

The same comments regarding memory requirements for ANSYS animation also apply for the DISPLAY program. A typical command stream for animation would look like this:

```
/SEG,DELE           ! Deletes all currently stored segments
/SEG,MULTI          ! Stores subsequent displays in segment memory
PLOT,4,8,1          ! Plots #4 - #8 (5 frames total) are stored in segment
                    !    memory  (Use PLOT,ALL to include every plot)
/SEG,OFF            ! Turn off the frame-capture function
ANIM,10             ! Cycles through the five frames 10 times
```

16.3.4. Capturing Animated Sequences Offline

You can use the DISPLAY program to capture animation offline (on film or videotape) in much the same way as you would do for the ANSYS program. See Chapter 15, "Animation" for a general discussion of this technique.

16.3.5. Exporting Files to Desktop Publishing or Word Processing Programs

You can use the DISPLAY program to translate ANSYS graphics files into Hewlett Packard Graphics Language (HPGL), Encapsulated PostScript (EPS), or some other external format, for possible use in outside desktop publishing and word processing programs. The Window's version of DISPLAY is also capable of exporting Metafile Graphics (WMF or EMF), in addition to the formats listed above. See your program's documentation for the particular format requirements.

16.3.5.1. Exporting Files on a UNIX System

To create such exportable graphics files, perform these tasks:

1. Using the DISPLAY program, issue the **FILEDISP** command to direct the program to read the desired filename. With the **/SHOWDISP** command, identify which graphics format you desire (HPGL, POSTSCRIPT, INTERLEAF, etc.). The HPGL format includes *color* HPGL capability, and the POSTSCRIPT format includes Encapsulated Postscript by default.

2. Still in the DISPLAY program, create one plot per file by typing **PLOT**,1, **PLOT**,2, ... etc. (or **PLOT**,ALL). The DISPLAY program will automatically assign an output filename.

3. Exit the DISPLAY program (using the **FINISH** command) and enter the word processing program. Create a graphics box in the document at the size of your choice. (Square boxes are recommended to avoid clipping the ANSYS image.)

4. Identify the appropriate file (for instance, **PSCRnn.GRPH**, **HPGLnn.GRPH**, etc.) and retrieve the image into the box. HPGL files will produce a screen image (bitmapped). You also can set up an EPS file to include a TIFF (Tagged Image File Format) bitmap or an Encapsulated PostScript Interchange Format (EPSI) bitmap for screen previewing. To do so, use the **PSCR**,TIFF and **PSCR**,EPSI command options within the DISPLAY program.

16.3.5.2. Exporting Files on a Windows System

The Windows version of DISPLAY provides direct METAFILE graphics export, in addition to the formats listed above. The system must be running in 32 bit mode (XP or 2000, running win32 or win32C). If the system is operating in Z-buffered mode, it will automatically switch to polygon mode when the file export is requested. A Windows metafile is created and saved in the specified directory by performing the following tasks:

1. From the FILE menu, select Export ANSYS Graphics.

2. Select Metafile.

3. To invert the colors of the graphic, choose the Controls option. The Standard Color option exports the ANSYS display file with a black background and the colors designated in the ANSYS program. The Invert WHITE/BLACK option will provide a white background, but still retain the original colors. Choose OK to continue.

4. Select the name, location and type of metafile to be exported. The option for WMF or EMF is provided. Each file export is assigned the default filename **file00.emf** (or wmf), with the "00" field incrementing for each subsequent file export. Older Windows products do not support EMF files.

Windows Metafiles can be exported directly from ANSYS (Windows systems only) by selecting **Utility Menu> PlotCtrls> Write Metafile**. The subsequent dialog boxes provide the same options listed above.

Windows Metafiles cannot be obtained directly from the UNIX platform (ANSYS or DISPLAY). The DISPLAY utility for Windows must be installed, and the ANSYS graphics files must be exported to the Windows file system in order to be converted. DISPLAY for Windows is shipped with all ANSYS products. Contact your ASD for installation instructions.

16.3.6. Editing the Neutral Graphics File with the UNIX GUI

The UNIX GUI provides an edit dialog box to create a new neutral graphics file from a subset of plots in an existing file. You access this dialog box by choosing **File > Edit Plot Sequence**.

16.4. Obtaining Hardcopy Plots

The DISPLAY program can generate hard copy through a number of external printer and plotter drivers, or by means of built-in terminal hardcopy capability.

16.4.1. Activating the Hardcopy Capability of Your Terminal on UNIX Systems

If your terminal has built-in hardcopy capability, you can execute **/PCOPY** (for HP work stations only) or **TERM**,COPY,*NCOPY* (in the DISPLAY program) to activate it. This option is available only during interactive sessions, with the **/SHOWDISP** specification active for terminals having hard copy capability.

16.4.2. Obtaining Hardcopy on External Devices Using the DISPLAY Program

The DISPLAY program supports a variety of printers and plotters via the HPGL, INTERLEAF, and POSTSCRIPT graphics drivers. To activate one of these drivers, first issue the appropriate **/SHOWDISP** command (such as **/SHOWDISP**,HPGL), and then set various driver options (using the **HPGL**, or **PSCR** commands, as appropriate). If you are using a pen plotter, the **TRANS** command will read the current neutral graphics file and will create a compressed and more efficient version of the file. Do not apply **TRANS** to files containing raster-mode hidden line displays, although **TRANS** will not adversely affect vector-mode hidden line displays.) Subsequent **PLOT** commands will create graphics files formatted for the desired device. The UNIX **xdisplay81** GUI also provides a Postscript only Print dialog box.

16.4.3. Printing Graphics Displays on a Windows-Supported Printer

To produce hard copy versions of ANSYS graphics displays, use the ANSYS Hard Copy menu and the Print menu in the DISPLAY program.

When you are printing to a local (not shared) printer, follow these steps:

1. Activate the Print Manager.
2. Select the Properties Menu for the desired printer.
3. Select the Details Menu.
4. Select the option Print directly to ports.

Chapter 17: The Report Generator

The report generator allows you to capture graphical and numerical data at any time throughout the analysis process and then assemble an HTML-based report using the captured data.

To capture data, you can use the report generator interactively or in batch mode. To assemble a report using the captured data, you can use any of these tools:

- The report generator itself (either interactively or in batch mode)
- A third-party (external) HTML editor
- Third-party (external) presentation software.

Using the report generator is a straightforward process, as follows:

1. Start the report generator and specify a directory to store your data and report(s).
2. Capture data (images, animations, tables and listings) that you want to include in your report.
3. Assemble your report using the captured data.

The remainder of this chapter describes the report generator in more detail. The following pages cover these topics:

- Starting the Report Generator
- Capturing an Image
- Capturing Animation
- Capturing a Data Table
- Capturing a Listing
- Assembling a Report
- Setting Report Generator Defaults.

17.1. Starting the Report Generator

To start the report generator, select **Utility Menu> File> Report Generator**. *Result:* The **ANSYS Report Generation** window appears, as shown:

Figure 17.1 Report Generator GUI
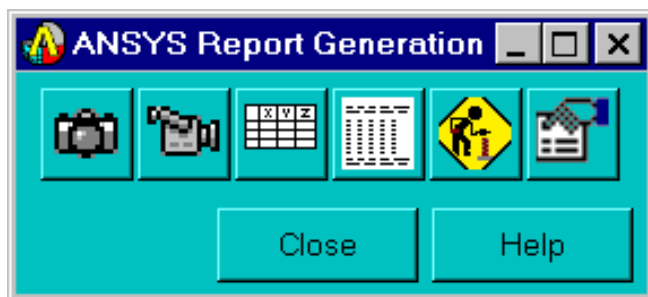
The buttons activate the following functions (from left to right): **Image Capture**, **Animation Capture**, **Table Capture**, **Listing Capture**, **Report Assembler**, and **Settings**. When using the

report generator, move the mouse pointer over any button to see a description of that button's function.

17.1.1. Specifying a Location for Captured Data and Reports

Your captured data, and any reports that you assemble, reside in a directory that you specify when you start the report generator. The default directory is **jobname_report**.

If the directory that you specify does not exist, the report generator creates it (after prompting you to approve the new directory). If the directory already exists, you have the option to *append* (add) captured data to any existing data in the directory or to *overwrite* (erase) the contents of the directory and start over.

When you specify a directory for your data and reports, the report generator writes this command to the ANSYS log file:

```
~eui,'ansys::report::setdirectory directory'
```

17.1.2. Understanding the Behavior of the ANSYS Graphics Window

The report generator constrains image size to accommodate most printers and paper sizes. When you start the report generator, it resizes the **ANSYS Graphics** window to obtain the optimum image size.

Note — After starting the report generator, do not adjust the size of the **ANSYS Graphics** window; otherwise, unpredictable results may occur.

To further facilitate printing, the image foreground changes to black and the background changes to white. (You can modify the report generator's settings to prevent the default color change. For more information, see Section 17.7: Setting Report Generator Defaults.)

When you close the report generator, it restores the **ANSYS Graphics** window's original size and color scheme.

17.1.3. A Note About the Graphics File Format

The report generator uses the Portable Network Graphics (PNG) format to store images. PNG files are small and suffer from little or no color loss. Fast becoming a standard, the PNG format enjoys support by many popular software products including Microsoft Internet Explorer™, Netscape Navigator™, Microsoft PowerPoint™ and Microsoft Word™.

17.2. Capturing an Image

This section describes how to capture and store a still image, either interactively or within a batch run.

The report generator saves images to the **images** subdirectory of your specified report directory. The name of each file is **image$n$.png**, where n is a sequential numeric identifier beginning at 1 and incrementing as you capture additional images.

17.2.1. Interactive

Follow these steps to capture a still image using the report generator GUI:

1. Click on the **Image Capture** button.

    Result: The **Image Capture** dialog appears.

2. Specify a caption for the captured image (for example, "Pentagonal Prism").

    The caption can contain APDL parameters in the format **%parm%**. (Specify "%%" if you want to display the "%" character in your caption.)

3. Click on the **OK** button.

    Result: The report generator issues this report command to the ANSYS program and saves the image to your report directory:

    ```
    ~eui,'ansys::report::imagecapture "caption"'
    ```

17.2.2. Batch

The following line must appear near the beginning of your batch code *before* any report commands:

```
~eui,'package require ansys'
```

To capture a still image via a batch run, insert this report command at the point in the run where you want to capture an image:

```
~eui,'ansys::report::imagecapture "caption"'
```

17.3. Capturing Animation

This section describes how to capture and store an animation sequence, either interactively or within a batch run. (Animation capture is possible only in postprocessing after issuing a **SET** command.)

The report generator saves all individual image files comprising an animation sequence to a subdirectory (of your specified report directory) named **animseq_n**, where n is a sequential numeric identifier beginning at 1 and incrementing as you capture additional animations. The functions for accessing the animation reside in **ansysAnimations.js**, a JavaScript file in the report directory.

17.3.1. Interactive

Follow these steps to capture an animation sequence using the report generator GUI:

1. Click on the **Animation Capture** button.

    Result: The **Animation Capture** dialog appears.

2. Specify a caption for the captured animation (for example, "Prism Deformed Shape Animation Result").

    The caption can contain APDL parameters in the format **%parm%**. (Specify "%%" if you want to display the "%" character in your caption.)

3. Specify the type of animation sequence to capture (such as mode shape, deformed shape, etc.) as applicable.

4. Click on the **OK** button.

    Result: The report generator issues this report command to the ANSYS program:

    ```
    ~eui,'ansys::report::animcapture "caption"'
    ```

    Also, the animation settings window associated with the animation type you selected (for example, **Animate Mode Shape** or **Animate Deformed Shape**) appears.

5. Modify the animation settings or accept the default settings, then click on the **OK** button.

Result: The report generator saves the animation sequence.

17.3.2. Batch

The following line must appear near the beginning of your batch code *before* any report commands:

```
~eui,'package require ansys'
```

To capture an animation sequence via a batch run via a batch run, insert this report command at the point in the run where you want to capture an animation:

```
~eui,'ansys::report::animcapture "caption"'
```

Follow the report command with an ANSYS animation command (for example, **ANTIME** or **ANDATA**).

17.4. Capturing a Data Table

This section describes how to capture and store a data table, either interactively or within a batch run.

The report generator appends captured table data to **ansysTables.js**, a file in your specified report directory containing the JavaScript functions for accessing your table data. (The file contains code to generate HTML as well as comments that hold the table information in a tab-delimited form, allowing you to paste the table data into software other than an HTML document.) The report generator assigns the name **table_*n*** to each captured table, where *n* is a sequential numeric identifier beginning at 1 and incrementing as you capture additional tables.

17.4.1. Interactive

Follow these steps to capture a data table using the report generator GUI:

1. Click on the **Table Capture** button.

    Result: The **Table Capture** dialog appears.

2. Specify a caption for the captured animation (for example, "Prism Material Properties Table").

    The caption can contain APDL parameters in the format **%*parm*%**. (Specify "%%" if you want to display the "%" character in your caption.)

3. Select a predefined table type from the list. (The report generator filters the list of available table types based on the current analysis.)

    If you select the **"Material properties"** table type, specify the currently defined materials via the **Material Selection** field.

    > *Note* — ANSYS does not display a material property which has no value associated with it.

    If you select the **Custom Table** option, specify the table size (that is, the number of columns and rows) via the **Custom Table Size** field.

4. Click on the **OK** button.

    Result: The report generator saves your *captured* table data. However, if you have selected the **Custom Table** option, see Section 17.4.1.1: Creating a Custom Table.

17.4.1.1. Creating a Custom Table

If you are creating a custom table, the **Custom Table Definition** dialog appears after you click on the **Table Capture** dialog's **OK** button. The dialog contains an empty table of the dimensions that you specified. For example, assuming that you specified a table of three columns and four rows, the dialog looks like this:

Figure 17.2 Custom Table Definition
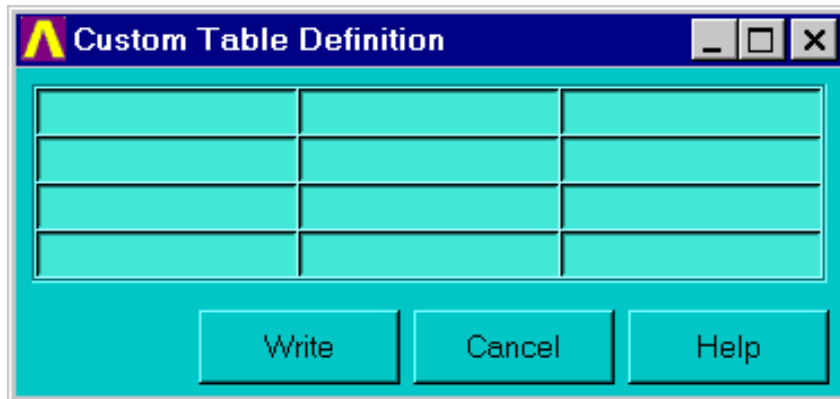
Type your custom information, which can include the following valid entries, into each cell:

Valid entry type	Example
Text	`Maximum Deflection`
Text with HTML tags	`Maximum Stress [Kg/mm<SUP>2</SUP>]`
An ANSYS *GET command	`*get,,node,10,u,y`
An ANSYS *GET command with HTML tags	`{<B><I>} {*get,,node,10,u,y} {</I></B>}` *Important:* The "{" and "}" characters are necessary for parsing purposes.

After typing entries into each cell, click on the **Write** button to save your custom table.

17.4.2. Batch

The following line must appear near the beginning of your batch code *before* any report commands:

```
~eui,'package require ansys'
```

To capture a data table via a batch run, insert this report command at the point in the run where you want to capture the table:

```
~eui,'ansys::report::tablecapture "caption" tableID materialID materialName'
```

The `tableID` value is a table identifier representing one of the following predefined table types:

Table ID	Description
1	Creates a table of the finite element entities used in the analysis
2	Creates a table of properties for the requested material ID used in the analysis
3	Creates a table of the loads applied in the analysis
4	Reaction Forces
5	Reaction Moments

Table ID	Description
6	Max Displacements
7	Directional Stress
8	Shear Stress
9	Principal Stress
10	Equivalent Stress and Stress Intensity
11	Thermal Gradients
12	Thermal Flux
13	Thermal Temperatures
14	Natural Frequencies
15	Rotation
16	Temperature
17	Pressure
18	Electric Potential
19	Fluid Velocity
20	Current
21	Electromotive Force Drop
22	Turbulent Kinetic Energy
23	Turbulent Energy Dissipation
24	Component Total Strain
25	Shear Total Strain
26	Principal Total Strain
27	Total Strain Intensity and Total Equivalent Strain
28	Component Elastic Strain
29	Shear Elastic Strain
30	Principal Elastic Strain
31	Elastic Strain Intensity and Elastic Equivalent Strain
32	Component Plastic Strain
33	Shear Plastic Strain
34	Principal Elastic Strain
35	Plastic Strain Intensity and Plastic Equivalent Strain
36	Component Creep Strain
37	Shear Creep Strain
38	Principal Creep Strain
39	Creep Strain Intensity and Creep Equivalent Strain
40	Component Thermal Strain
41	Shear Thermal Strain
42	Principal Thermal Strain
43	Thermal Strain Intensity and Thermal Equivalent Strain
44	Component Pressure Gradient and Sum
45	Component Electric Field and Sum
46	Component Electric Flux Density and Sum

Table ID	Description
47	Component Magnetic Field Intensity and Sum
48	Component Magnetic Flux Density and Sum

If `tableID` is 2 (Material Properties), two additional arguments are required:

- *materialID* -- Corresponds to an ANSYS material ID

- *materialName* -- The name of the material for the current table.

17.5. Capturing a Listing

This section describes how to capture the results of an ANSYS command, either interactively or within a batch run.

The report generator appends listing data to **ansysListings.js**, a file in your specified report directory containing the JavaScript functions for accessing the listing. (The file contains code to generate HTML as well as comments that hold the list information, allowing you to paste the listing into software other than an HTML document.) The report generator assigns the name **listing_*n*** to each captured listing, where *n* is a sequential numeric identifier beginning at 1 and incrementing as you capture additional listings.

If you intend to use a captured listing in an HTML report (assembled using either the report generator or a third-party HTML tool), be aware that HTML sizes the text smaller if its width is greater than 132 columns; however, all text associated with the listing may still not fit on a printed page.

17.5.1. Interactive

Follow these steps to capture a listing using the report generator GUI:

1. Click on the **Listing Capture** button.

   Result: The **Listing Data Capture** dialog appears.

2. Specify a caption for the listing (for example, "Prism Model Area Listing").

3. Specify the ANSYS command to issue to generate the output text.

4. Click on the **OK** button.

   Result: The report generator issues this report command to the ANSYS program and saves the listing:

   ```
   ~eui,'ansys::report::datacapture "caption" ansysCommand'
   ```

17.5.2. Batch

The following line must appear near the beginning of your batch code *before* any report commands:

```
~eui,'package require ansys'
```

To capture a listing via a batch run, insert this report command at the point in the run where you want to capture a listing:

```
~eui,'ansys::report::datacapture "caption" ansysCommand'
```

17.6. Assembling a Report

This section describes how to assemble your captured image and text data into a report interactively, within a batch run, or manually using the JavaScript interface.

17.6.1. Interactive Report Assembly

Follow these steps to assemble your report using the report generator GUI:

1. Click on the **HTML Report Assembler** button.

    Result: The **HTML Report Assembler** window appears, as shown:

Figure 17.3 HTML Report Assembler Window
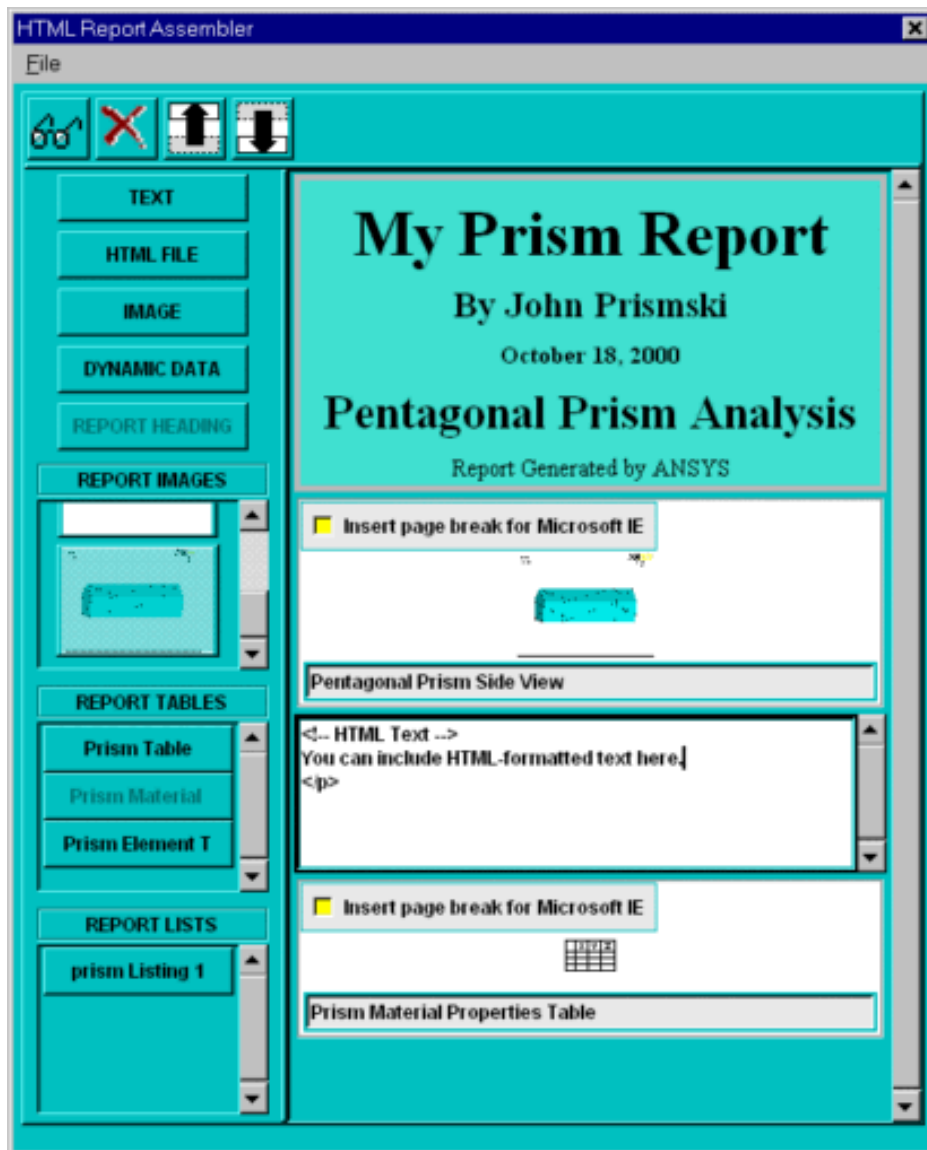
Click on the buttons and thumbnail images in the left panel to add components to your report. The larger panel on the right is your work area, displaying the components that you have chosen to include in the report.

2. Assemble the components of your report.

    Click on the buttons and thumbnail images in the left panel to add your captured images and text, and other components, as follows:

Button or Field	Purpose
TEXT	Inserts a text area allowing you to type HTML-formatted text into your report.
HTML FILE	Inserts a specified *existing* HTML file.
IMAGE	Inserts a specified image file (in PNG, JPG, JPEG or GIF format). The report assembler copies the image to your specified report directory and inserts a thumbnail image in the work-area panel.
DYNAMIC DATA	Inserts a text area allowing you to type ANSYS commands, the results of which appear in your report. The dynamic data becomes part of the HTML code written to the ANSYS log file. The report generator also writes the **ansys::report::interpdynamicdata** command to the log file; the command must process the HTML code in order for the results of the ANSYS command(s) to appear in your report.
REPORT HEADING	Inserts a specified title, author name and subtitle, and includes the current date. The heading component always appears at the top of your report.
REPORT IMAGES	Inserts any or all of your captured images and animations. Move the mouse pointer over a thumbnail to see the caption that you assigned to the corresponding image or animation when you captured it. Click on any thumbnail image to insert the image or animation that it represents into your report.
REPORT TABLES	Inserts a captured data table. A button appearing in this field is labeled with the first 15 characters of the caption that you assigned to the corresponding table when you captured it. Move the mouse pointer over a button to see the caption that you assigned to the corresponding table when you captured it. Click on a button to insert the table that it represents into your report.
REPORT LISTS	Inserts a captured raw-data output listing. A button appearing in this field is labeled with the first 15 characters of the caption that you assigned to the corresponding listing when you captured it. Move the mouse pointer over a thumbnail to see the caption that you assigned to the corresponding listing when you captured it. Click on a button to insert the listing that it represents into your report.

3. Preview and clean up your report, as follows:

To perform this editing task ...	Do this:
Preview your report	Click on this button in the toolbar:
Delete a report component	Select (click on) the component that you want to delete, then click on this button in the toolbar:

To perform this editing task ...	Do this:
Change the order of the report components	Select (click on) the component that you want to move. Click on either of these buttons to move the component up or down, respectively: *Note* — If your report contains a heading, it remains fixed at the top of the report.
Change the caption of a report component	Click on the caption field that you want to change to place the cursor within the text. Type your changes directly into the field.
Prevent Microsoft Internet Explorer (IE) from splitting a component between two pages during printing	Check the box labeled **Insert page break for Microsoft IE** for the appropriate component. This feature is especially useful *after* you have printed an initial draft of your report from within IE and determined how your printed report looks.

4.   Save your work.

Select **File> Save** periodically (or **File> Save and Close** to save your report and close the report assembler window).

17.6.2. Batch Report Assembly

Issue the ***CREATE** command in batch mode to open and append HTML tags to your report file. By default, the report assembler uses the ***CREATE** command to write the report to the ANSYS log file.

17.6.3. Report Assembly Using the JavaScript Interface

For maximum flexibility and usefulness, the report generator employs JavaScript, a coding language for Web browsers supported by Microsoft Internet Explorer™ and Netscape Navigator™. JavaScript allows easy encapsulation of data and access to that data.

The report generator creates JavaScript functions (in form) as you capture image and text data. Therefore, rather than using the report generator's built-in report assembler, you can use the JavaScript functions to manually assemble your HTML-based report.

17.6.3.1. Inserting an Image

The following example JavaScript code creates an image in the HTML report that you are assembling. If the specified image is not available, your report will contain a message indicating the problem.

```
<script LANGUAGE="JavaScript1.2" SRC="ansysImages.js"> </script>
<script>
imgName('imgCaption');
</script>
```

Following is an explanation of the JavaScript code:

<script LANGUAGE="JavaScript1.2" SRC="ansysImages.js"> </script>

   Loads the **ansysImages.js** file to provide the images. You must include this line of code at least once in your HTML document and *before* calling the `imgName` function. Typically, this line appears in the <HEAD> section of an HTML document.

<script>
> The HTML tag to begin JavaScript code.

imgName
> A unique image name as it appears in the **ansysImages.js** file.

imgCaption
> The caption to display for the image. This value is a string and must be enclosed in single quotation marks. It can include HTML tags around the text as well. If not specified, the function uses the default image caption.

</script>
> The HTML tag to end JavaScript code.

17.6.3.2. Inserting an Animation

The following example JavaScript code creates an animation sequence in the HTML report that you are assembling. If the specified animation is not available, your report will contain a message indicating the problem.

```
<script LANGUAGE="JavaScript1.2" SRC="ansysAnimations.js"> </script>
<script>
animName('animCaption',animTime, 'animDirect');
</script>
```

Following is an explanation of the JavaScript code:

<script LANGUAGE="JavaScript1.2" SRC="ansysAnimations.js"> </script>
> Loads the **ansysAnimations.js** file to provide the animation sequences. You must include this line of code at least once in your HTML document and *before* calling the $animName$ function. Typically, this line appears in the <HEAD> section of an HTML document.

<script>
> The HTML tag to begin JavaScript code.

animName
> A unique animation name as it appears in the **ansysAnimations.js** file.

animCaption
> The caption to display for the animation sequence. This value is a string and must be enclosed in single quotation marks. It can include HTML tags around the text as well. If not specified, the function uses the default animation caption.

animTime
> The time delay (in milliseconds) between each display of an individual image in the animation sequence. This value is limited by the capabilities of your computer hardware; therefore, a value lower than 500 typically has little effect on the animation. If not specified, the function assumes the default value of 500.

animDirect
> The direction of play, as follows:

> forward

> After displaying the last individual image of the animation sequence, the function repeats the animation from the beginning (that is, starting with the first image and incrementing).

> back

> After displaying the last individual image of the animation sequence, the function repeats the animation in the opposite direction (that is, starting with the prior image and decrementing). After displaying the first image again, the animation repeats in the forward direction.

If not specified, the function assumes the default value of back.

</script>
    The HTML tag to end JavaScript code.

17.6.3.3. Inserting a Data Table

The following example JavaScript code creates a data table in the HTML report that you are assembling. If the specified table is not available, your report will contain a message indicating the problem.

```
<script LANGUAGE="JavaScript1.2" SRC="ansysTables.js"> </script>
<script>
tableName('tableCaption');
</script>
```

Following is an explanation of the JavaScript code:

<script LANGUAGE="JavaScript1.2" SRC="ansysTables.js"> </script>
    Loads the **ansysTables.js** file to provide the data table. You must include this line of code at least once in your HTML document and *before* calling the `tableName` function. Typically, this line appears in the <HEAD> section of an HTML document.

<script>
    The HTML tag to begin JavaScript code.

`tableName`
    A unique data table name as it appears in the **ansysTables.js** file.

`tableCaption`
    The caption to display for the table. This value is a string and must be enclosed in single quotation marks. It can include HTML tags around the text as well. If not specified, the function uses the default table caption.

</script>
    The HTML tag to end JavaScript code.

17.6.3.4. Inserting a Listing

The following example JavaScript code creates an ANSYS output listing in the HTML report that you are assembling. If the specified listing is not available, your report will contain a message indicating the problem.

```
<script LANGUAGE="JavaScript1.2" SRC="ansysListings.js"> </script>
<script>
listingName('listingCaption');
</script>
```

Following is an explanation of the JavaScript code:

<script LANGUAGE="JavaScript1.2" SRC="ansysListings.js"> </script>
    Loads the **ansysListings.js** file to provide the listing. You must include this line of code at least once in your HTML document and *before* calling the `listingName` function. Typically, this line appears in the <HEAD> section of an HTML document.

<script>
    The HTML tag to begin JavaScript code.

`listingName`
    A unique listing name as it appears in the **ansysListings.js** file.

`listingCaption`
    The caption to display for the listing. This value is a string and must be enclosed in single quotation marks. It can include HTML tags around the text as well. If not specified, the function uses the default listing caption.

</script>
    The HTML tag to end JavaScript code.

17.7. Setting Report Generator Defaults

This section describes how to change settings affecting report generator operation. Click on the **Settings** button to open the **Settings** dialog, as shown:

Figure 17.4 Report Generator Settings Dialog
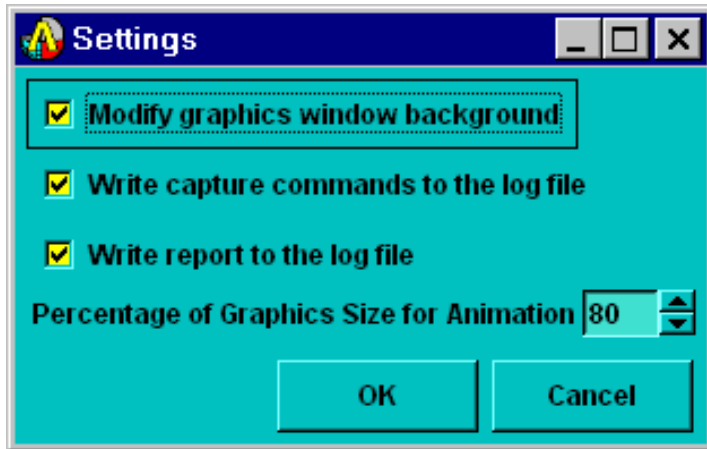
You can control whether or not the report generator:

- Reverses the foreground and background colors in the **ANSYS Graphics** window (and hides the background image) when you start the report generator

- Writes capture commands to the ANSYS log file

- Writes your assembled report to the ANSYS log file.

You can also use the **Settings** dialog to set the percentage value that the report generator uses to reduce image sizes for animations.

By default, all options are activated (that is, all check boxes contain check marks) and the report generator uses an image size of 100 percent of the **ANSYS Graphics** window size for animations. Any changes that you make in the settings window become the new defaults for subsequent report generator sessions until you change them again.

Chapter 18: File Management and Files

18.1. File Management Overview

The ANSYS program uses files extensively for data storage and retrieval, especially while solving an analysis. The files are named **filename.ext**, where **filename** defaults to the jobname, and **ext** is a unique two- to four-character value that identifies the contents of the file. The jobname is a name you can specify while entering the ANSYS program (using the **/FILNAME** command (**Utility Menu> File> Change Jobname**) after entering the program. If you specify no jobname, it defaults to **FILE** (or **file**).

> *Note* — File names (both jobname and extension) may be lowercase on some systems. For example, if the jobname is **bolt**, you may have files at the end of an ANSYS analysis which could include:

bolt.db	Database file
bolt.emat	Element matrices
bolt.err	Error and warning messages
bolt.log	Command input history
bolt.rst	Results file

Table 18.1: "Temporary Files Written by the ANSYS Program" and Table 18.2: "Permanent Files Written by the ANSYS Program" show a list of files written by the ANSYS program. Files that are generated and then deleted sometime before the end of the ANSYS session are called *temporary* files (Table 18.1: "Temporary Files Written by the ANSYS Program"). Files that remain after the ANSYS session are called *permanent* files (Table 18.2: "Permanent Files Written by the ANSYS Program").

18.1.1. Executing the Run Interactive Now or DISPLAY Programs from Windows Explorer

If you are running ANSYS on a Windows system, you can double-click on the following types of files from the Windows Explorer to execute the Run Interactive Now or DISPLAY programs:

- Double-click on a **.db** or **.dbb** file to execute the Run Interactive Now program. When executed in this way, ANSYS will use the **Initial jobname**, **Total Workspace** (**-m**), and **Database** (**-db**) values previously set in the **Interactive** dialog box. To change these settings, access the **Interactive** dialog box (select Interactive from the ANSYS folder), change the settings as desired, and click on Close.

- Double-click on a **.grph** or **.f33** file to execute the DISPLAY program. The first plot that appears in the file will be loaded into DISPLAY automatically.

For more information about the **Interactive** dialog box, see the *ANSYS Operations Guide*. For more information about the DISPLAY program, see Section 16.3: Viewing and Translating Neutral Graphics Files with the DISPLAY Program in the *ANSYS Basic Analysis Guide*.

18.2. Changing the Default File Name

When you activate the ANSYS program, you can change the default jobname from **file** or **FILE** to a name that is more meaningful. To do so, activate the program as follows:

```
ansys81 -j newjobname
```

The value –j (or –J) is an option indicating that a new jobname, **newjobname**, follows. Once this command executes, all ANSYS files produced during this run will have a filename of **newjobname.ext**.

> *Note* — If an ANSYS job is running in the background, *do not execute the ANSYS program interactively* in the same directory unless you use a different jobname.

ANSYS can process blanks in file and directory names. Be sure the file name is enclosed in a pair of single quotes if a blank appears in the file name. Note that many UNIX commands do not support object names with blank spaces.

18.3. Sending Output to Screens, Files, or Both

One of the files commonly referred to throughout the ANSYS documentation set is the output file (**Jobname.OUT**). If you are running on a UNIX system and you want to send ANSYS output only to the screen, open the launcher via the `launcher81` command. Then select the **Preferences** tab and select **Screen Only** for the **Send Output To** option. The output "file" will be your ANSYS output window. If you choose **Screen and File**, then an actual text file called **Jobname.OUT** will also be written in your current working directory.

> *Note* — When you launch ANSYS from the launcher and direct output to both screen and file, ANSYS will not immediately display output in the output window. The I/O buffer must be filled or flushed first. Errors and warnings will flush the I/O buffer. You can also issue certain commands (e.g., **/OUTPUT**, **NLIST**, or **KLIST**) to force flush the I/O buffer.

Windows systems do not support the **Screen and File** option. The default behavior is to print output to the output window. You can redirect your text output to a file by using the **/OUTPUT** command.

18.4. Text Versus Binary Files

Depending on how files are used, the program writes them in text (ASCII) form or binary form. For example, **ERR** and **LOG** files are text, while **DB**, **EMAT**, and **RST** files are binary. In general, files that you may need to read (and edit) are written in text form, and all other files are written in binary form.

Binary files may be of *external* or *internal* type. *External* binary files are transportable between different computer systems. *Internal* binary files are usable only by the computer hardware system on which they were written and cannot be transported. By default, all ANSYS-saved binary files are of external type. You may change them to internal type using either of the following:

> **Command(s): /FTYPE**
> **GUI: Utility Menu> File> ANSYS File Options**

You *cannot* change the database file (**Jobname.DB**) or the results file (**Jobname.R***xx*) to internal type.

Below are some tips for using binary files:

- If you do not intend to transport files between different computer systems, you may save some CPU time by specifying all binary files to be of internal type. Some systems take longer to write external binary files than to write internal binary files.

- When transferring files via FTP (File Transfer Protocol), you must set the BINARY option before doing the transfer.

- Most ANSYS binary files *must* have write permission to be used, even if the data is only being read from the file. However, the database files (**file.DB**) and results files (**file.RST**, **file.RTH**, etc.) can be read-only. When you save a read-only **file.DB**, the existing read-only file is saved to a **file.DBB**. However, you cannot

save the read-only **file.DB** a second time, because it will attempt to write over the **file.DBB**, which ANSYS will not allow.

> **Warning:** Binary files are *not* backward compatible to previous releases of the ANSYS program. For example, you cannot use binary files produced by ANSYS 8.1 with release ANSYS 5.7 or earlier. Attempting to use binary files from later releases with an earlier release can cause serious operating problems in ANSYS. For a list of the files that are upwardly compatible, see Table 18.2: "Permanent Files Written by the ANSYS Program".

18.4.1. ANSYS Binary Files over NFS

You can access ANSYS binary files (for example, **file.TRI**, **file.DB**, **file.RST**) from NFS-mounted disk partitions. However, this usage is discouraged because heavy network traffic may result. Also, network traffic may cause NFS errors, which in turn can cause the ANSYS program to read or write an ANSYS binary file incorrectly.

18.4.2. Files that ANSYS Writes

The following tables list the files that ANSYS writes.

Table 18.1 Temporary Files Written by the ANSYS Program

Identifier	Type	Contents
ANO	Text	Graphics annotation commands [/**ANNOT**]
BAT	Text	Input data copied from batch input file [/**BATCH**]
DO*n*	Text	Do-loop commands for nesting level n
DSCR	Binary	Scratch file (ANTYPE=2, Modal Analysis)
EROT	Binary	Rotated element matrices
LSCR	Binary	Scratch file (ANTYPE=4, Mode Superposition)
LV	Binary	Scratch file from substructure generation pass with more than one load vector.
LNxx	Binary	Scratch files for the sparse solver (x = 1-32)
MASS	Binary	Compressed mass matrix for modal analysis (subspace method)
MMX	Binary	Work matrix for modal analysis (subspace method)
PAGE	Binary	Page file for ANSYS virtual memory (database space)
PCS	Text	Scratch file for PCG solver
PC*n*	Binary	Scratch file for PCG solver (n = 1 to 10)
SCR	Binary	Scratch file for Jacobi Conjugate Gradient solver
SSCR	Binary	Scratch file from substructure generation pass

Many of the permanent ANSYS files are upwardly compatible. Files that generally can be used by future releases of ANSYS have a **Y** in the **Upward** column.

Table 18.2 Permanent Files Written by the ANSYS Program

Identifier	Type	Upward	Contents
ANF	Text	-	ANSYS Neutral Format file, written by default by ANSYS after a Connection product import[1]
BDB	Binary	-	Database for best design (optimization) [**OPKEEP**]

Identifier	Type	Upward	Contents
BFIN	Text	-	Interpolated body forces written as **BF** commands [**BFINT**]
BRFL	Binary	-	FLOTRAN results file for best design (optimization) [**OPKEEP**]
BRMG	Binary	-	Magnetic results file for best design (optimization) [**OPKEEP**]
BRST	Binary	-	Structural results file for best design (optimization) [**OPKEEP**]
BRTH	Binary	-	Thermal results file for best design (optimization) [**OPKEEP**]
CBDO	Text	-	Interpolated DOF data written as **D** Commands [**CBDOF**]
CDB	Text	**Y**	Text database file [**CDWRITE**]
CMAP	Text	-	Color map file
CMD	Text	**Y**	Commands written by ***CFWRITE**
CMS	Binary	**Y**	Component Mode Synthesis file.
DB	Binary	**Y**	Database file [**SAVE**, **/EXIT**]
DBB	Binary	**Y**	Copy of database file created when a nonlinear analysis terminates abnormally (used for traditional restart)
DBE	Binary	-	Database file from **VMESH** failure in batch mode
DBG	Text	-	FLOTRAN "debug" file (contains solution information)
DSUB	Binary	**Y**	Superelement DOF solution from use pass
ELEM	Text	**Y**	Element definitions [**EWRITE**]
EMAT	Binary	**Y**	Element matrices
ERR	Text	-	Error and warning messages
ESAV	Binary	**Y**	Element saved data **ESAV** files created by nonlinear analyses may not be upwardly compatible.
FATG	Text	-	Fatigue data [**FTWRITE**]
FULL	Binary	-	Assembled global stiffness and mass matrices
GRPH	Text	**Y**	Neutral graphics file
IGES	Text	**Y**	IGES file from ANSYS solid model data [**IGESOUT**]
LDHI	Text	**Y**	Loading and boundary conditions of load steps (used for multiframe restart)
LGW	Text	**Y**	Database command log file [**LGWRITE**]
L*nn*	Binary	**Y**	Load case file (where nn = load case number) [**LCWRITE**]
LOG	Text	**Y**	Command input history
LOOP	Text	-	Optimization looping file
MCF	Text	**Y**	Modal coordinates from harmonic or transient analysis
MCOM	Text	**Y**	Mode combination commands from spectrum analysis
MODE	Binary	**Y**	Modal matrices (modal or buckling analysis)
MP	Text	**Y**	Material property definitions [**MPWRITE**]
NLH	Text	**Y**	Nonlinear diagnostics file that tracks results or contact quantities throughout the solution (**NLHIST**).
NODE	Text	**Y**	Node definitions [**NWRITE**]
OPO	Text	-	ANSYS output for last optimization loop
OPT	Text	-	Optimization data
OSAV	Binary	-	Copy of **ESAV** file from last converged substep

ANSYS Basic Analysis Procedures Guide . ANSYS Release 8.1 . 001972 . © SAS IP, Inc.

Identifier	Type	Upward	Contents
OUT	Text	-	ANSYS output file
PARM	Text	**Y**	Parameter definitions [**PARSAV**]
PFL	Text	-	FLOTRAN printout file
PSD	Binary	-	**PSD** file (modal covariance matrices, etc.)
RCN	Binary	**Y**	Results file for initial contact state
RDB	Binary	**Y**	State of the database at the start of the first substep of the first load step (used for multiframe restart)
RDF	Text	-	FLOTRAN residual file [**FLDATA**,OUTP]
RDSP	Binary	-	Reduced displacements
REDM	Binary	-	Reduced structure matrix
RFL	Binary	**Y**	FLOTRAN results file
RFRQ	Binary	-	Reduced complex displacements
RMG	Binary	**Y**	Results file from magnetic field analysis
R*nnn*	Binary	**Y**	Element saved records, solution commands, and status for a single substep of a load step (used for multiframe restart)
RST	Binary	**Y**	Results file from structural and coupled-field analyses
RSW	Text	-	FLOTRAN "wall" results file
RTH	Binary	**Y**	Results file from thermal analysis
RUN	Text	-	FLOTRAN run data
SELD	Binary	**Y**	Superelement load vector data from generation pass
S*nn*	Text	**Y**	Load step files (where nn = load step number) [**LSWRITE**]
SORD	Text	-	Superelement name and number from use pass
STAT	Text	-	Status of an ANSYS batch run
SUB	Binary	**Y**	Superelement matrix file from generation pass
TB	Text	**Y**	Hyperelastic material constants [***MOONEY**]
TRI	Binary	-	Triangularized stiffness matrix
USUB	Binary	**Y**	Renamed **DSUB** File for input to substructure expansion pass
XBC	Text	-	FLOTRAN boundary condition data (ANSYS to FLOTRAN)
XGM	Text	-	FLOTRAN geometry data (ANSYS to FLOTRAN)
XIC	Text	-	FLOTRAN initial condition data (ANSYS to FLOTRAN)

1. For more information about the files produced by an ANSYS Connection import, see the *ANSYS Connection User's Guide*.

18.4.3. File Compression

Many file compression utilities exist for UNIX (*e.g.*, **compress**, **gzip**) and Windows (*e.g.*, **PKzip**, **WinZip**). ANSYS cannot read compressed files. However, you can compress ANSYS models to save space when archiving, so long as you uncompress the models before trying to read them into ANSYS.

18.5. Reading Your Own Files into the ANSYS Program

In many situations, you will need to read in your own files while using the ANSYS program. The file may be a text file of ANSYS commands or a binary file of ANSYS data.

Use the **/INPUT** command (**Utility Menu> File> Read Input from**) to read in a text file containing ANSYS commands. For instance, you can read in the log file (**Jobname.LOG**) from a previous ANSYS session. For example, the following command causes the ANSYS program to read the file **MATERIAL.INP** from the current directory.

```
/INPUT,MATERIAL,INP
```

Table 18.3: "Commands for Reading in Text Files" lists other commands that you can use to read in text files.

Table 18.3 Commands for Reading in Text Files

Command	GUI Menu Path	Purpose
*USE	**Utility Menu> Macro> Execute Data Block**	Reads in macros
PARRES	**Utility Menu> Parameters> Restore Parameters**	Reads in parameters (**Jobname.PARM**) files
EREAD	**Main Menu> Preprocessor> Modeling> Create> Elements> Read Elem File**	Reads in element (**Jobname.ELEM**) files
NREAD	**Main Menu> Preprocessor> Modeling> Create> Nodes> Read Node File**	Reads in node (**Jobname.NODE**) files
MPREAD	**Main Menu> Preprocessor> Loads> Load Step Opts> Other> Change Mat Props> Read from File** **Main Menu> Preprocessor> Material Props> Read from File** **Main Menu> Solution> Load Step Opts> Other> Change Mat Props> Read from File**	Reads in material property (**Jobname.MP**) files
ISFILE	**Main Menu> Solution> Load Step Opts> Initial Stress> Read IS File**	Reads in initial stress (**Jobname.IST**) files

Table 18.4: "Commands for Reading in Binary Files" lists GUI paths or commands you can use to read in binary data files.

Table 18.4 Commands for Reading in Binary Files

Command	GUI Menu Path	Purpose
RESUME	**Utility Menu> File> Resume from** **Utility Menu> File> Resume Jobname.DB**	Reads in database (**Jobname.DB**) files
SET[1]	**Utility Menu> List> Results> Load Step Summary**	Reads in results files (**Jobname.RST**, **Jobname.RTH**, **Jobname.RMG**, **Jobname.RFL**)
OPRESU[2]	**Main Menu> Design Opt> Opt Database> Resume**	Reads in the optimization data file (**Jobname.OPT**)

1. in the POST1 postprocessor
2. in the OPT postprocessor

18.6. Writing Your Own ANSYS Files from the ANSYS Program

Besides the files that the ANSYS program automatically writes during an analysis, you can also force files to be written as necessary. A commonly used file-write command is **/OUTPUT**, which allows you to redirect text output from the screen to a file. For example, to redirect POST1 stress printout to a file, the commands would be:

```
/OUTPUT,STRESS,OUT! Output to file STRESS.OUT
PRNSOL,COMP! Component stresses
/OUTPUT! Output back to screen
```

GUI equivalents for the **/OUTPUT** command are:

GUI:

> **Utility Menu> File> Switch Output to> File**
> **Utility Menu> File> Switch Output to> Output Window**

Table 18.5: "Other Commands for Writing Files" lists other file-write commands used during an analysis are:

Table 18.5 Other Commands for Writing Files

Command	GUI Menu Path	Purpose
SAVE	**Utility Menu> File> Save as**	Writes the database to **Jobname.DB**
PARSAV	**Utility Menu> Parameters> Save Parameters**	Writes parameters to **Jobname.PARM**
EWRITE	**Main Menu> Preprocessor> Modeling> Create> Elements> Write Elem File**	Writes element definitions to **Jobname.ELEM**
NWRITE	**Main Menu> Preprocessor> Modeling> Create> Nodes> Write Node File**	Writes node definitions to **Jobname.NODE**
MPWRITE	**Main Menu> Preprocessor> Loads> Other> Change Mat Props> Write to File** **Main Menu> Preprocessor> Material Props> Write to File** **Main Menu> Solution> Load Step Options> Other> Change Mat Props> Write to File**	Writes material properties to **Jobname.MP**

You can also redirect graphics output (plots) from the screen to a neutral graphics file.

18.7. Assigning Different File Names

As mentioned earlier, you can use the **/FILNAME** command at the Begin level to assign a jobname for all subsequently written files. Use the **/ASSIGN** command (**Utility Menu> File> ANSYS File Options**) to assign a different name, extension, and directory to a file. For example, the following command reassigns the element matrix file (identifier EMAT) to **MYFILE.DAT** in the "save_dir" directory:

```
/ASSIGN,EMAT,MYFILE,DAT,SAVE_DIR/
```

The "/" is a delimiter that separates the directory name from the file name. It is system-dependent, so you must use the delimiter(s) appropriate for your system. You can assign only a specific set of files. Refer to the **/ASSIGN** command description (in the *ANSYS Commands Reference*) for the complete list.

18.8. Reviewing Contents of Binary Files (AUX2)

The auxiliary processor AUX2 allows you to print ANSYS binary files in readable format. Use it mainly to verify file formats (for debugging purposes). The output from a "dumped" binary file is unlabeled and must be correlated with known formats documented in the *Guide to Interfacing with ANSYS*. Be aware, though, that a complete file dump may produce many pages of unnecessary printout. The `Format` argument on the **FORM** command (**Utility Menu> File> List> Binary Files**) allows you to control the amount of output.

Use **HBMAT** to dump any matrix written on the assembled global matrix file (**.FULL** file) or the superelement matrix file (**.SUB** file). This matrix is written to a new file (**.MATRIX**) in the standard Harwell-Boeing format.

Note — The Harwell-Boeing format is column-oriented. That is, non-zero matrix values are stored with their corresponding row indices in a sequence of columns. However, since the ANSYS matrix files are stored by row and not column, when the **HBMAT** command is used with a non-symmetric matrix, the transpose of the matrix is, in fact, written.

18.9. Operating on Results Files (AUX3)

The auxiliary processor AUX3 allows you to operate on results files by deleting sets or by changing values such as the load step, load substep, cumulative iteration, or time.

18.10. Other File Management Commands

Table 18.6: "Additional File Management Commands and GUI Equivalents" lists other useful file management commands.

Table 18.6 Additional File Management Commands and GUI Equivalents

Command	GUI Path	Purpose
/COPY	**Utility Menu> File> File Operations> Copy**	Copy existing binary files from within ANSYS
/CLOG	None	Copy the log file during an interactive ANSYS session
/RENAME	**Utility Menu> File> File Operations> Rename**	Rename files
/DELETE	**Utility Menu> File> File Operations> Delete**	Delete files
/FDELE	**Utility Menu> File> ANSYS File Options**	Delete certain files during a solution run (to save disk space)

Chapter 19: Memory Management and Configuration

19.1. Memory Management

This chapter presents an overview of the ANSYS memory management scheme and defines some frequently used terms. For a discussion of improving ANSYS performance, see Chapter 13, "Improving ANSYS Performance and Parallel Performance for ANSYS" in the *ANSYS Advanced Analysis Techniques Guide*.

19.2. Basic Concepts

The memory available on your computer is called *real memory*- the amount of actual, physical memory (RAM) available through memory chips. The minimum amount of real memory recommended for the ANSYS program varies from system to system and is listed in your *ANSYS Installation and Configuration Guide*.

19.2.1. ANSYS Work and Swap Space Requirements

The ANSYS program requires some space to reside in memory, plus additional work space. The ANSYS work space defaults to 512 MB (1024 MB for the 64-bit Windows and 64-bit Linux platforms). As shown in Figure 19.1: "Comparing Available Memory", the total memory required for the ANSYS program usually exceeds the amount of real memory available. The additional memory comes from *system virtual memory*, which is simply a portion of the computer's hard disk used by the system to supplement physical memory. The disk space used for system virtual memory is called *swap space*, and the file is called the *swap file*. On some systems it is referred to as a *page file*. Other systems maintain multiple files, or even dedicated disk sectors to act as virtual memory. The amount of swap space required for the ANSYS program depends on the amount of real memory available, the size of the ANSYS executable, and the amount of ANSYS work space.

Figure 19.1 Comparing Available Memory
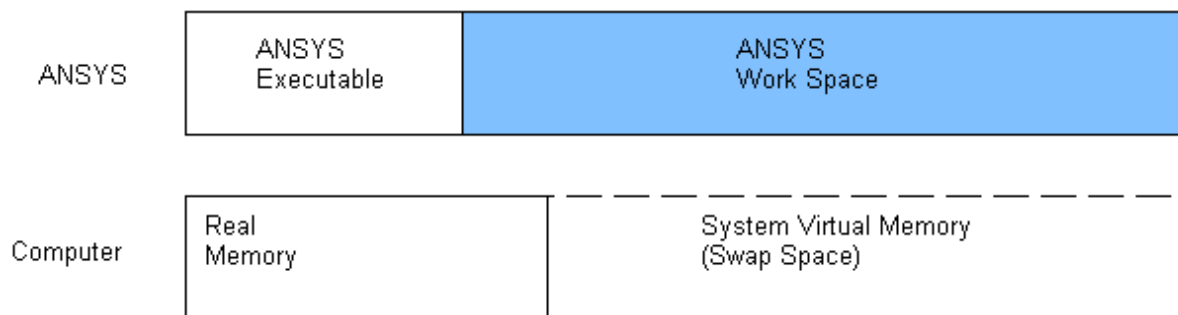
System virtual memory is used to satisfy additional ANSYS memory requirements.

19.2.2. How ANSYS Uses its Work Space

To understand how ANSYS uses its work space (the shaded portion in Figure 19.1: "Comparing Available Memory"), you need to understand its two components: *database space* and *scratch space*, as shown in Figure 19.2: "ANSYS Work Space". Database space is used to work with the ANSYS database (model geometry, material properties, loads, etc.). Scratch space is where all internal calculations are done - element matrix formulation, frontal solution, Boolean calculations, and so on. The default total work space is 512 MB, of which 256 MB are assigned to database

space, and 256 MB are assigned to scratch space. (Part of the scratch space stores binary file buffers; see the description of NUM_BUFR later in this chapter.)

Figure 19.2 ANSYS Work Space
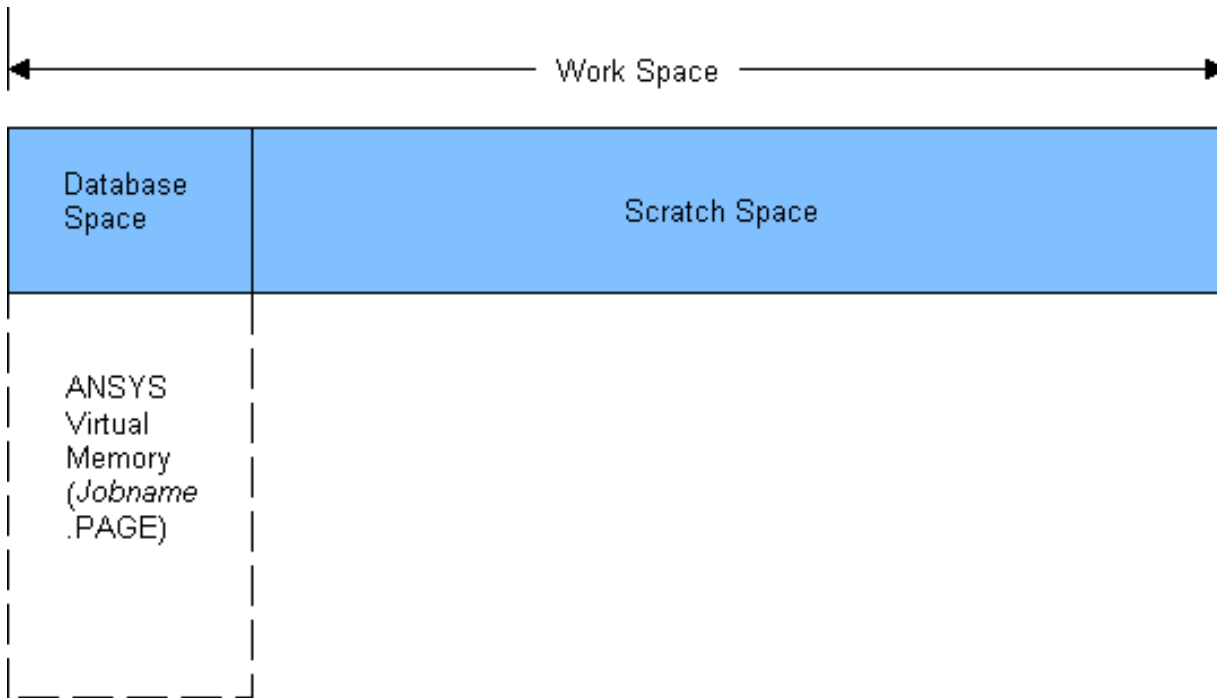
If your model database is too big to fit in the database space, the ANSYS program uses *ANSYS virtual memory*, which is, again, a portion of the hard disk used for data overflow. The main difference between *system* virtual memory and *ANSYS* virtual memory is that the former uses system functions to swap data between memory and disk, whereas the latter uses ANSYS programming instructions. The file used for ANSYS virtual memory is called the *page file* and has the name **Jobname.PAGE**. Its size depends entirely on the size of the database. When the page file is first written, the program issues a message to that effect. Use of the page file is not desirable because it is a less efficient way of processing data. You may be able to prevent it by allocating more database space (discussed in Section 19.3: How and When to Do Memory Management).

If internal calculations can't fit in the scratch space, the ANSYS program will attempt to allocate additional memory to meet these requirements. If this occurs, you will see an alert message informing you that the problem has grown beyond the specified memory allocation and that ANSYS has allocated additional memory.

In general, you should have enough real physical memory to comfortably run an ANSYS job. If you are using extra memory beyond physical memory only temporarily (such as for meshing or equation reordering), the performance impact from using virtual memory will be small. However, in a situation such as the matrix storage for the PCG solver exceeding physical memory, the performance can be as much as ten times slower for the solve command.

19.3. How and When to Do Memory Management

Normally, you shouldn't worry about memory management in ANSYS. The ANSYS memory manager can allocate extra memory from the system when it needs to *in almost all cases*.

The following sections will give you guidance on when it is likely that you will need to use the −m command line option.

19.3.1. Why You Normally Don't Need the -M Switch

The -M switch tells ANSYS how much memory to allocate from the system at the start of the ANSYS run. This first block of memory that ANSYS uses won't be released back to the system, regardless of whether or not ANSYS is using it. This allows you to reserve a minimum block of memory for ANSYS use, keeping other applications from using it up. Since ANSYS doesn't release this first block of memory, it may be denying access to system memory to other applications, such as the help system.

Memory ANSYS allocates through the -M switch must be in one contiguous block. For example, a -m setting of 1800 and a -db option of 300 instructs ANSYS to allocate a 1500 MB contiguous block of scratch memory (1800 - 300 = 1500).

Starting with a large amount of memory on 32 bit systems can cause problems. When ANSYS starts, many different items compete for space within the process' virtual memory. This includes the ANSYS program itself, various shared libraries (or dynamic link libraries) required to run ANSYS, as well as other shared libraries that are required by the operating system. All of these items must reside within the process space given to the ANSYS program. If one or more of these items happens to land near the middle of the virtual address space given to the ANSYS process, it splits the memory that ANSYS sees into smaller pieces.

Figure 19.3 Split Memory Space
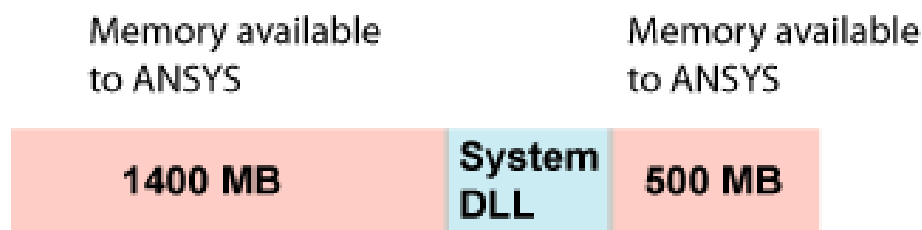
This occurs quite often on Windows machines. If you attempt to start ANSYS with a -m value of 1800 (requesting a very large scratch space) on a machine with memory as shown above, it will fail because there is no single block of memory large enough to allow ANSYS to start up with that much allocated space. There is 1900 MB of memory available to the ANSYS process, however.

The figure above shows a simplified example where a single DLL has split the ANSYS virtual memory space into two pieces. In actual practice, it is likely that several such splits may occur making many smaller memory pieces available.

If you allow ANSYS to start up with its default memory allocation, it will start with a rather modest memory block. When it needs more memory, it will allocate more from the system. In the above example, this would allow ANSYS to access all 1900 MB of available memory space, one piece at a time.

19.3.2. Exceptions: When to Specify the Amount of Work Space

The -m switch is necessary if you also need to use the -f (fixed memory mode) command line option. If ANSYS issues either of the following error messages, you must use the -f memory option to solve your model:

```
This model requires more scratch space than ANSYS can address in dynamic memory
mode. Turn on fixed memory via the -f command line option. Problem terminated.
The memory (-m) size requested cannot currently be addressed using dynamic memory
mode. ANSYS addressing can be changed by turning on fixed memory via the -f command
line option.
```

A side effect of the -f switch is that much of the ANSYS program cannot take advantage of memory growth. If you need to use the -f switch, then you must specify a scratch memory amount large enough to solve your model.

Other reasons to use the -m option include:

- You have been instructed by ANSYS customer support to use this as a "workaround" for a problem.

- Some platforms (currently 64-bit versions of Windows and Linux) can't take advantage of memory growth at all, making the -m value necessary.

- You are trying to solve a very large job using the sparse solver. The sparse solver can operate efficiently in out-of-core mode, but it still requires a contiguous block of memory. On 32-bit Windows systems, if a large job fails with the default memory option, increase the initial memory allocation so that the sparse solver memory is satisfied from that initial memory block. This strategy works best when ANSYS is run in batch mode and executes the solve portion as a separate run.

19.3.3. Changing the Amount of ANSYS Work Space

The easiest way to do this is to use the work space entry option (-m) while activating the program, either via the ANSYS launcher or via the ANSYS execution command. For example, to request 400 MB of ANSYS work space (instead of the default of 512 MB), the ANSYS execution command would read:

```
ansys81 -m 400
```

(The execution command syntax is system-dependent.)

When you use the -m option, system virtual memory is allocated at run time to achieve the work space requested.

Other ways to change the maximum ANSYS work space are:

- Specifying the work space size you want on the dialog boxes that appear when you select interactive mode or batch mode from the ANSYS launcher.

- Using a different VIRTM_MB value in your **config81.ans** file. A later section in this chapter discusses this file in detail.

  Caution: Be careful when specifying a value for the -m option. Entering an amount larger than needed will waste system resources and degrade system performance.

By changing the amount of ANSYS work space, you are in effect changing the available scratch space. This is because the database space is held constant at 256 MB, as illustrated in the following figure.

Figure 19.4 Changing ANSYS Work Space
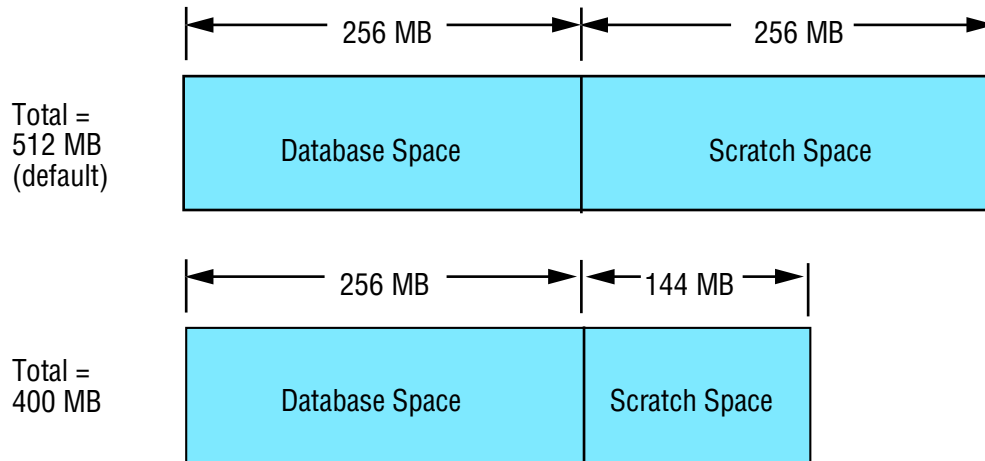
Only scratch space is changed, and database space is held constant.

The following list describes situations in which you may need to change ANSYS work space:

- Boolean operations among several volumes (for example, **VADD**,ALL). You need more scratch space in such a case. This is an example where a large memory space may be needed temporarily and may be satisfied from system virtual memory.

- Several users sharing one ANSYS installation (such as in a training class). Because all users share the available memory (real memory plus system virtual memory), having less ANSYS work space requested by each user would allow more users to run simultaneously (with correspondingly smaller model sizes available for each user).

- Some analyses and analysis options require more work space, including subspace modal analysis, random vibration (PSD) analysis, etc. In the case that such an analysis is right at the limit of physical memory, you should conserve memory by reducing database memory. In cases where you are running up to the physical limits of the machine, increase the initial memory block given to ANSYS in order to maximize the possibility of memory reuse within the ANSYS runs.

- Large wavefronts. The larger the wavefront of your model, the more scratch space needed. This guideline is only true for frontal solver jobs. Most jobs in ANSYS no longer use the frontal solver.

Most solvers in ANSYS now use approximately 1 GB of memory per million degrees of freedom (DOFs). If models are dominated by 2-D shell elements, the memory requirement will be lower; if they are dominated by 3-D higher-order brick elements, the memory requirement will be higher. The sparse solver will run larger jobs in out-of-core mode and will require disk space of approximately 10 GB per million DOFs. The disk memory requirement varies in the same increasing manner for shell 2-D models versus solid 3-D models. If the solver memory estimate far exceeds the default memory for ANSYS, try increasing the initial memory allocation for ANSYS.

19.3.4. Allocating Memory on Systems with 64-Bit Architecture

On some 64-bit UNIX systems, such as the HP AlphaServer (Compaq) or Silicon Graphics systems, you may encounter memory problems when running very large models. Such problems typically occur with models that require 4 GB or more of scratch space. Should this occur, you'll see the following error message:

```
The memory (-m) size requested cannot currently be addressed using
dynamic memory mode. ANSYS addressing can be changed by turning on
fixed memory via the -f command line option.
```

To remedy this problem, you can issue the -f ANSYS command line switch with *no arguments*. For your reference, the -f switch takes the following arguments:

nogrow
Sets ANSYS to use a fixed-mode memory addressing scheme (identical to memory addressing in versions of ANSYS previous to 5.5). No dynamic memory allocation is allowed.

no | off
Sets ANSYS to use dynamic memory allocation for scratch memory as needed. For more information about ANSYS and dynamic memory, see Section 19.2.2: How ANSYS Uses its Work Space. This argument is included to override a corresponding setting in the **config81.ans** file (MEM_GROW = 0). This is the default.

(no argument)
Sets ANSYS to use fixed-mode memory allocation for scratch memory, but allows other sections of the ANSYS work space to grow as required.

19.3.5. Changing Database Space From the Default

This is done by requesting more or less database space. Given a fixed amount of ANSYS work space, allocating more database space leaves less for scratch space and vice versa, as illustrated in the following figure.

Figure 19.5 Dividing Work Space
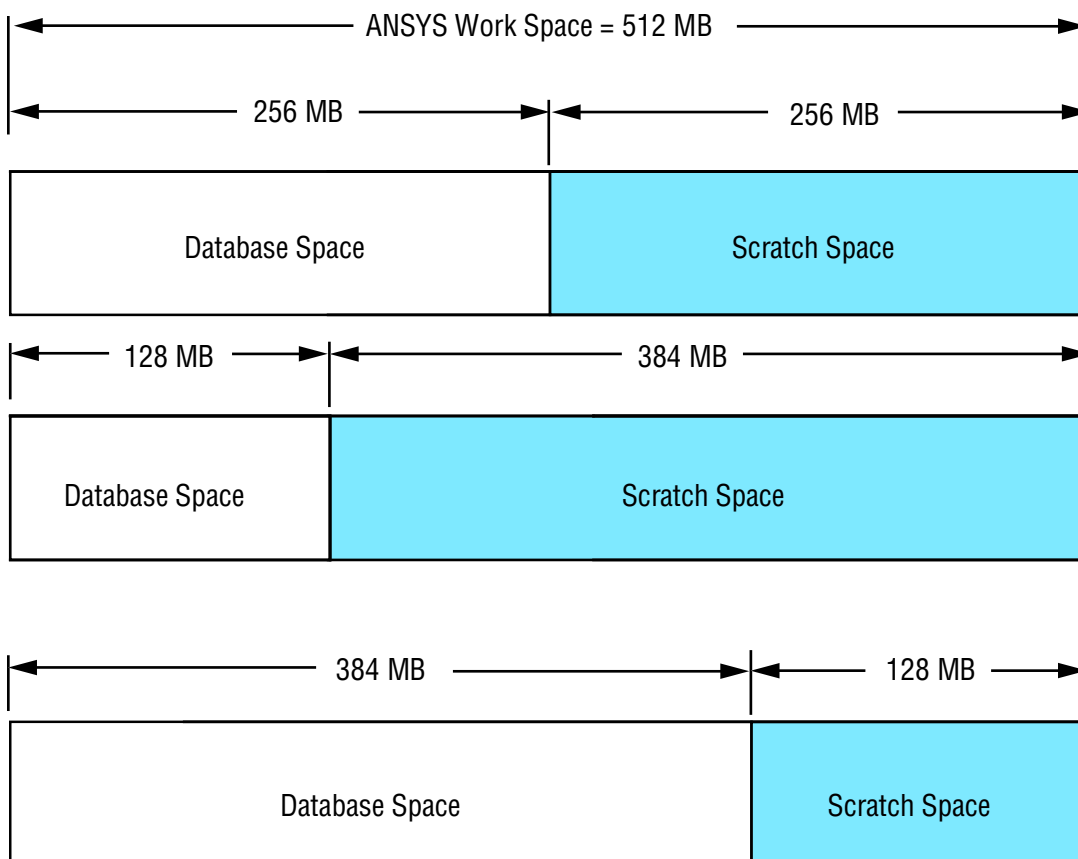
Allocating more database space leaves less for scratch space, and vice versa.

To allocate more or less database space, use the database space entry option (-db) while starting up the program, either on the ANSYS launcher or on the ANSYS execution command. Although all of the above diagrams correspond to a -m value of 512 MB, their -db values correspond to 256, 128 and 384 MB, respectively.

You may need to divide work space differently in these situations:

- When you are about to solve a large-wavefront model or a computationally intensive analysis such as a subspace modal analysis or random vibration (PSD) analysis. By reducing the amount of database space, you will increase the scratch space, allowing more memory for solution.

- When you see the message about the page file (**Jobname.PAGE**) being written. This means that the database space is too small for your model, so you should increase it. *Be aware, though, that increasing the database space decreases scratch space,* so you also may want to increase total work space if you plan to do memory-intensive operations (Boolean operations, load case combinations, etc.).

19.4. The Configuration File

When you execute the ANSYS program, it reads a configuration file, **config81.ans**, if one exists. This file controls system-dependent settings such as the size of each file buffer, maximum number of database pages in memory, etc. On computer systems that support mixed-case file names, such as UNIX systems, the file name is assumed to be in lower case: **config81.ans**.

ANSYS searches for the **config81.ans** file first in the current (working) directory, next in the login (home) directory, and finally in the ANSYS **/apdl** directory. The search path for **config81.ans** is identical to the **start81.ans** and **stop81.ans** files.

The configuration file is a fixed-format file, consisting of a list of keywords followed by an equal (=) sign and a number. The keyword must begin in column 1, the equal sign must be in column 9, and the number must begin in column 10. A sample **config81.ans** file is shown next, followed by a brief explanation of each of the keywords.

Sample config81.ans File

```
NUMRESLT=1000
NO_RSTGM=1
NO_ELDBW=0
NUM_BUFR=2
SIZE_BIO=4096
VIRTM_MB=512
NUM_VPAG=128
SIZ_VPAG=16384
NUM_DPAG=8192
MEM_GROW=12
NUM_PROC=1
LOCALFIL=0
CONTACTS=1000
ORDERER_=2
EXT_FILE=1
MX_NODES=5000
MX_ELEMS=2000
MX_KEYPT=500
MX_LINES=1000
MX_AREAS=300
MX_VOLUS=200
MX_REALS=10
MX_COUPS=10
MX_CEQNS=10
FILESPLT=128
```

Note — Since many of the values for **config81.ans** are dependent on the system being used, a range of values for each keyword is provided. Note that on most computer systems, 1 integer word = 4 bytes.

NUMRESLT is the maximum number of result sets (substeps) allowed on the results file: 10 minimum. It defaults to 1000 and can be changed at the Begin level with the **/CONFIG**,NRES command.

NO_RSTGM is a key that determines whether or not the geometry data will be written to the results file: 0 (write the data) or 1 (do not write the data). This is especially useful for large, complex analyses where the results file would become excessively large during the solution.

NO_ELDBW is a key that determines whether or not to write results into the database after a solution. When VALUE = 0, write results into the database. When VALUE=1, do not write results into the database. It defaults to 0 and can be changed at the Begin level with the **/CONFIG**,NOELDB command.

NUM_BUFR is the number of buffers per file stored in scratch space: 1 to 32. A buffer is a chunk of space used to "hold" data in memory before they are written to the hard disk. The program waits for the buffer to be completely "filled up" and only then "empties" it onto the hard disk. This prevents frequent disk input-output activity, which can be time consuming.

NUM_BUFR defaults to 4 and can be changed at the Begin level with the **/CONFIG**,NBUF command. It is used for the **EROT**, **ESAV**, **EMAT**, **FULL**, and **TRI** files. On systems with a large amount of real memory, you can increase NUM_BUFR or SIZE_BIO (or both) to keep ANSYS solution files in memory rather than on disk. This can save a significant amount of disk input-output activity and may be practical for small problems with many substeps.

SIZE_BIO is the size of each file buffer: 1024 to 4194304 integer words (4 KB to 16 MB). It defaults to 16384 and can be changed (on most systems) at the Begin level with the **/CONFIG**,SZBIO command. See NUM_BUFR for details.

VIRTM_MB is the amount of total ANSYS work space requested for the current ANSYS session. It defaults to 512 MB; it can be changed with the work space entry option, as explained in the basic concepts section earlier in this chapter. (See Figure 19.6: "Memory Diagram in Terms of Configuration Keywords".) You can also use VIRT_MEM in place of VIRTM_MB to specify the ANSYS work space in integer words.

NUM_VPAG is the maximum number of database pages in memory, ranging from 16 to 8192. It defaults to 512 for Windows and 1024 for UNIX. You can change NUM_VPAG or SIZ_VPAG (or both) to change the amount of database space; see Section 19.3: How and When to Do Memory Management in this chapter, and Figure 19.6: "Memory Diagram in Terms of Configuration Keywords".

Figure 19.6 Memory Diagram in Terms of Configuration Keywords
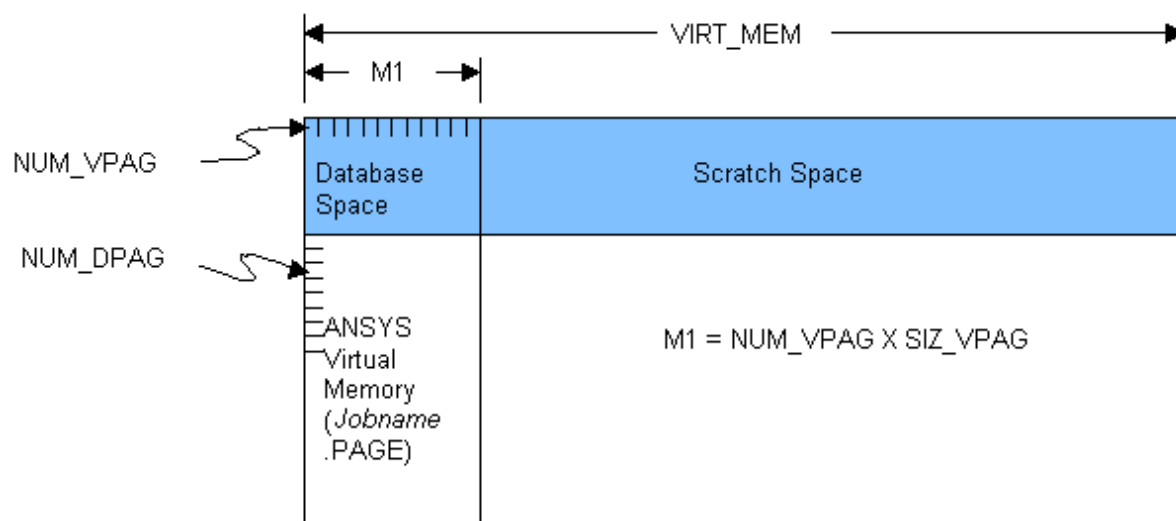
SIZ_VPAG is the size of each database page: 4096 to 131072 integer words (16 to 512 KB), defaults to 16384 (64 KB). You can change SIZ_VPAG or NUM_VPAG (or both) to change the amount of database space. SIZ_VPAG also affects the size of the page file; see NUM_DPAG.

Note — However, the product of the SIZ_VPAG X NUM_VPAG cannot exceed 2 gigawords . If you increase one of those values, you may need to decrease the other to stay within this limit.

NUM_DPAG is the number of database pages on disk: NUM_VPAG defaults to 131068 (the maximum). This number times SIZ_VPAG determines the maximum size of the page file (**Jobname.PAGE**), which is written only if the database becomes too large to fit in the database space in memory. If the page file is written, sufficient disk space must be available to accommodate it, or the program will abort.

MEM_GROW is the starting size of the memory block (in MB) that ANSYS will attempt to allocate should a problem grow larger than will fit in the current scratch space allocation. If ANSYS attempts to allocate additional scratch space, it will start with a memory block size equal to MEM_GROW and then reduce this by halves until it can allocate additional memory. If not specified, MEM_GROW defaults to one-half the initial scratch space. To turn off dynamic memory allocation (use a fixed-memory model), set MEM_GROW=0.

NUM_PROC is the number of processors to use. The default is one processor. The NUM_PROC value that you specify must be less than or equal to the total number of available processors; however, if the number you specify exceeds the number of available processors, ANSYS uses the maximum number of available processors. You can change the number of processors at the Begin level (overriding the NUM_PROC setting) via the **/CONFIG**,NPROC command or the **SETNPROC** macro. The setting affects only multiprocessing computers with the appropriate parallel versions of the ANSYS product (versions of the program specifically programmed for parallel processing). See Chapter 13, "Improving ANSYS Performance and Parallel Performance for ANSYS" in the *ANSYS Advanced Analysis Techniques Guide* for more information about Parallel Performance for ANSYS.

Note — On some systems, an additional user-specified environmental variable (NCPUS) may be necessary to select the number of processors available. Most systems use a single processor by default. See the installation guide for your computer system for more information about your system's default configuration.

LOCALFIL is a key that determines when files are to be closed: 0 (globally closed) or 1 (locally closed). It defaults to 0 (globally closed) and can be changed at the Begin level with the **/CONFIG**,*LOCFL* command. This key is applicable only to the **EROT**, **ESAV**, **EMAT**, **FULL**, and **TRI** files. Locally closed files (LOCALFIL=1) may be deleted earlier during solution if requested with the **/FDELE** command. This may be helpful while running large problems. Globally closed files are closed at the end of the run and are not opened and closed each substep. This saves time in analyses with many substeps.

CONTACTS is the number of contact elements that are expected to be in contact at any given time. It defaults to 1000 and can be changed at the Begin level with the **/CONFIG**,NCONT command. This is *not* the same as the total number of contact elements in the model.

ORDERER_ is a key that determines the automatic element reordering scheme: 0 (**WSORT**,ALL), 1 (**WAVES**), or 2 (both). It defaults to 2 and can be changed at the Begin level with the **/CONFIG**,ORDER command.

EXT_FILE is a key that determines the format of binary files: 0 (internal format) or 1 (external format). It defaults to 1 (external format) and can be changed at the Begin level with the **/FTYPE** command. (On some systems, internal format and external format are identical.)

Any of the following nine keywords that you do not specify is set to 100 the first time ANSYS encounters it. Whenever the current maximum is exceeded, the keyword value automatically doubles. The maximum values are dynamically expanded, even at first encounter.

•   MX_NODES is the maximum number of nodes. You can change the value (on most systems) at the Begin level with the **/CONFIG**,MXND command.

- MX_ELEMS is the maximum number of elements. You can change the value (on most systems) at the Begin level with the **/CONFIG**,MXEL command.

- MX_KEYPT is the maximum number of keypoints. You can change the value (on most systems) at the Begin level with the **/CONFIG**,MXKP command.

- MX_LINES is the maximum number of lines. You can change the value (on most systems) at the Begin level with the **/CONFIG**,MXLS command.

- MX_AREAS is the maximum number of areas. You can change the value (on most systems) at the Begin level with the **/CONFIG**,MXAR command.

- MX_VOLUS is the maximum number of volumes. You can change the value (on most systems) at the Begin level with the **/CONFIG**,MXVL command.

- MX_REALS is the maximum number of sets of real constants (element attributes). You can change the value (on most systems) at the Begin level with the **/CONFIG**,MXRL command.

- MX_COUPS is the maximum number of sets of coupled degrees of freedom. You can change the value (on most systems) at the Begin level with the **/CONFIG**,MXCP command.

- MX_CEQNS is the maximum number of constraint equations. You can change the value (on most systems) at the Begin level with the **/CONFIG**,FMXCE command.

FILESPLT is the integer value indicating the file split point in megawords. A megaword is 1024*1024 4-byte words, or 4 MB. All files that are eligible for splitting will be split into a new file every increment of *xxxx* megawords (where *xxxx* is the value specified with this keyword). You can change the value (on most systems) at the Begin level with the **/CONFIG**,FSPLIT command. See Section 6.2: Splitting Files Across File Partitions in the *ANSYS Operations Guide* for more information on splitting files.

You can change many of the configuration settings at program start-up using ANSYS entry options, at the Begin level with the **/CONFIG** command, or with other ANSYS commands. In most cases, therefore, there is no need to create your own **config81.ans** file. Also, the default settings for each computer system have been chosen for efficient running of "typical" models on typical system configurations. Change them only for atypical models or atypical systems.

19.5. ANSYS Memory Error Messages

The memory (-m) size requested is not currently available. Reenter ANSYS command line with less memory requested.

The database (-db) space requested is not currently available. Reenter ANSYS command line with less database space requested.

Either of these messages may occur at ANSYS startup.

The memory used by ANSYS resides within the system virtual memory. The required system virtual memory for the amount of memory that you requested for either the workspace (-m) or database space (-db) for ANSYS (via either the ANSYS command line or the configuration file) is not currently available. Request less workspace or database space if possible and execute the ANSYS program again. If the initial requested memory is required, then wait until sufficient system virtual memory is free and try again.

Another alternative is to increase the system's virtual memory via the System icon in the Control Panel.

The memory (-m) size requested cannot currently be addressed using dynamic memory mode. ANSYS addressing can be changed by turning on fixed memory via the -f command line option.

This message may occur at ANSYS startup.

Include the -f command line option on the ANSYS execution command to remedy the problem.

For more information, see Memory Management and Configuration in the *ANSYS Basic Analysis Guide*.

This model requires more scratch space than ANSYS can address in dynamic memory mode. ANSYS addressing can be changed by turning on fixed memory via the -f command line option. Problem terminated.

This message may occur while you are running ANSYS.

Include the -f command line option on the ANSYS execution command to remedy the problem.

For more information, see Memory Management and Configuration in the *ANSYS Basic Analysis Guide*.

This model requires more scratch space than available, currently xx words (YY MB). The scratch space may be increased by increasing the work space, currently xx words (YY MB), via the ANSYS command line memory option. Problem terminated.

This message may occur while you are running ANSYS.

If you are running ANSYS in dynamic memory mode and the internal calculations that ANSYS is trying to perform cannot fit in the scratch space, the ANSYS program will *attempt* to allocate additional memory to meet the requirements. However, this message may still appear if:

- Some portions of ANSYS cannot use the additional memory that was allocated or allocate the memory when it is needed. The processing that ANSYS was trying to perform when this message appeared took place in one of these portions.
- You used the -f command line switch so that ANSYS would use a fixed-mode memory addressing scheme. Thus, no dynamic memory allocation is allowed.

If you receive this message, specify a higher -m value on the ANSYS execution command or change your -f command line option and execute ANSYS again.

For more information, see Memory Management and Configuration in the *ANSYS Basic Analysis Guide*.

This model requires more scratch space than available, currently xx words (YY MB). ANSYS was not able to allocate more memory to proceed. Increasing the work space, currently xx words (YY MB), via the ANSYS command line memory option may help. Problem terminated.

This message may occur while you are running ANSYS.

This message appears when you are running ANSYS in dynamic memory mode and ANSYS has attempted to allocate additional memory, but it failed because it could not find a large enough "chunk" of memory to proceed.

If you receive this message, specifying a higher -m value on the ANSYS execution command and executing ANSYS again may help. However, as noted above, if a large enough "chunk" of memory is still unavailable, changing the -m value will not help.

For more information, see Memory Management and Configuration in the *ANSYS Basic Analysis Guide*.

Problem too large. Matrix wavefront of M rows exceeds allowed N.

You have exceeded the maximum wavefront allowed on the version of ANSYS that you are authorized to run. Contact your ANSYS Support Distributor (ASD) for licensing information.

Index

Symbols
3-D graphics devices, 8–2

A
abridged solution menus, 3–6, 3–6
acceleration, 2–24
acceleration fields
    applying to a body, 2–24
accumulating forces, 2–12
ACEL command, 2–24
action buttons, 1–10
Algebraic Multigrid (AMG) solver, 3–1, 3–5
analysis
    restarting, 3–18
analysis options, 1–20
analysis title definition, 1–2
analysis type
    defining, 1–20
ANCNTR command, 15–2
ANCUT command, 15–2
ANDATA command, 15–2
ANDSCL command, 15–2
ANFLOW command, 15–2
/ANGLE command, 9–3
angular velocity, 2–24
ANHARM Command, 15–2
ANIM command, 15–1, 16–4
animation, 15–1
    capturing animation sequences offline, 15–2, 16–5
    commands, 15–1
    creating animated displays, 15–1
    in Windows environments, 15–5
    macros, 15–1
    of contoured deformed shapes, 15–2
    of cutting planes, 15–2
    of deformed mode shapes, 15–2
    of deformed shapes, 15–2
    of deformed shapes varying over time, 15–2
    of isosurfaces, 15–2
    viewing animated sequences on a screen, 16–4
ANISOS command, 15–2
anisotropic material properties, 1–8
ANMODE command, 15–2
annotation, 14–1
ANSYS
    changing division of work space, 19–6
    changing work space size, 19–4
    getting started, 1–1
    upwardly compatible files written by, 18–3

work space and swap memory requirements, 19–1
ANSYS windows
    activating and deactivating, 9–1
    copying displays between, 9–1
    defining, 9–1
    deleting, 9–1
    removing a frame from, 9–2
ANTIME command, 15–2
antisymmetry boundary conditions, 2–7
ANTYPE command, 1–20
APLOT command, 11–1
applying
    forces, 2–11
    loads, 2–14
        function of node number vs. surface load, 2–14
arc-length method, 2–3
areas
    specifying body loads for, 2–22
arithmetic operations among results data, 5–51
array parameter solution method, 3–16
arrays
    defining parameters, 3–16
ASCII files, 18–2
assemblies, 7–5
    definition of, 7–1
    nesting, 7–6
/AUTO command, 9–4
automatic graphics scaling, 8–7
automatic iterative (fast) solver, 3–1, 3–6
automatic time stepping, 2–51
AUX2, 18–7, 18–8
.AVI files, 15–5
AVRES command, 10–3
axisymmetric loads and reactions, 2–26
/AXLAB command, 13–3

B
B-H curve, 1–7
backing store, 8–4
beams
    applying pressure loads on, 2–14
BFCUM, 2–22
BFE command, 2–19
BFECUM command, 2–22
BFESCAL command, 2–22
BFK command, 2–21
BFSCALE command, 2–22
BFTRAN command, 2–22
BFUNIF command, 2–22
binary files, 18–2
    accessing over NFS, 18–3
    external type, 18–2

pretension, 2–58
ramped, 2–4
stepped, 2–4
surface, 2–13
surface loads, 2–1
tapered, 2–15
verifying in steady-state thermal analysis, 2–34
verifying loads defined with TABLE array parameters, 2–34
LPLOT command, 11–1
LSCLEAR command, 2–57
LSDELE command, 2–57
LSREAD command, 2–57
LSSOLVE command, 3–15
LSWRITE command, 2–57

M

macros
animation, 15–2
SETSPEED, 3–28
magnetic command macros, 5–60
magnetic forces, 5–60
magnetic loads, 2–1
magnetomotive force, 5–60
MAGOPT command, 2–52
mapping results, 5–58
mass density, 2–24
mass matrix load, 2–1
master degrees of freedom
graphics symbols for, 11–10
MAT command, 1–4
material library
advantages of, 1–17
conventions for material library filenames, 1–18
creating a material library file, 1–18
format of material library files, 1–18
reading a material library file, 1–19
setting a read/write path for files, 1–18
using material library files, 1–17
material model interface, 1–8
material properties
anisotropic, 1–8
constant properties, 1–4
defining, 1–4, 1–4
isotropic properties, 1–4
linear properties, 1–4
material library files, 1–17
nonlinear properties, 1–7
orthotropic properties, 1–4
plotting versus temperature, 1–4
temperature-dependent properties, 1–4, 1–4, 1–4
material reference numbers, 1–4

memory
error messages, 19–10
memory management, 19–1
memory requirements, 3–29
memory saving
when generating stiffness matrix, 3–3
meshing, 1–19
midside nodes
results, 5–53
modal analysis
limitations on, 1–20
MODE command, 2–26, 2–52
mode numbers, 2–52
models
basic steps in creating, 1–1
creating geometry for, 1–19
MODOPT command, 1–20
MP command, 1–4, 1–4
MPAMOD command, 1–4
MPDATA command, 1–4
MPDRES command, 1–4
/MPLIB command, 1–17, 1–18
MPLIST, 1–4
MPPLOT command, 1–4, 13–1
MPREAD command, 1–19
MPTEMP command, 1–4
MPTGEN command, 1–4
MPTRES command, 1–4
MPWRITE command, 1–18
multi-plotting, 8–8
multiframe restart, 3–21
multiple load steps
defining, 2–57
solving, 3–14
multiple solve solution method, 3–15
Multiprocessor solvers, 3–1

N

neutral graphics file, 16–2
Newton-Raphson options, 1–20
NOCOLOR command, 16–4
node rotation angles, 2–7
nodes
sorting by number, 5–30
/NOERASE command, 9–1
non-summable data, 5–57
nonlinear analysis
substeps in, 2–50
nonlinear load step options, 2–54
nonlinear material properties, 1–7
NPLOT command, 11–1
NPRINT command, 6–15

for SHELL61, 5–8
printing
    from UNIX systems, 16–2
    from Windows systems, 16–1
PRTIME command, 6–15
PRVAR command, 6–15
PSOLVE command, 3–27
/PSYMB command, 9–7

Q
Q-slices, 12–6, 15–2
query picking, 10–1

R
RALL command, 3–29
ramped loads, 2–4, 2–51
raster mode, 11–5
rate-independent analysis, 2–3
/RATIO command, 11–4
reaction loads, 5–28
real constants
    defining, 1–3
    listing, 1–3
real memory, 19–1
reference temperature, 2–52
repeating
    body loads, 2–22
/REPLOT command, 8–8
report generator, 17–1
    animation capture, 17–3
    ANSYS Graphics window behavior, 17–2
    assembling the report, 17–8
    custom table creation, 17–5
    image capture, 17–2
    JavaScript interface, 17–10
    listing capture, 17–7
    PNG graphics file format, 17–2
    setting defaults, 17–13
    specifying a working directory, 17–2
    starting, 17–1
    table capture, 17–4
RESET command, 6–5
RESP command, 6–20
restarting
    an analysis, 3–18, 3–18, 3–21
        from an incompatible database, 3–21
        nonlinear, 3–18
    multiframe analysis, 3–21
    singleframe, 3–18
restarting an analysis, 1–20
results
    mapping onto a surface, 5–22

using as loads in another analysis, 2–1
results coordinate system, 5–48
results files, 1–22, 4–2
results mapping, 5–58
reviewing analysis results, 1–22
reviewing results (see Postprocessing)
RFILSZ command, 3–29
RITER command, 3–28
RLIST command, 1–3
RMEMRY command, 3–29
rotating results, 5–48
RSPEED command, 3–28
RSTAT command, 3–29
RTIMST command, 3–28
run time estimation, 3–28
/RUNSTAT command, 3–3, 3–28
RUNSTAT module, 3–28
RWFRNT command, 3–29

S
SBCTRAN command, 2–12
scalar magnetic potential, 2–52
scaling
    body loads, 2–22, 2–23
    temperature constraints, 2–9
scratch space, 19–1
screens
    sending output to, 18–2
/SEG command, 15–1, 16–4
selecting
    by component or assembly, 7–7
    via commands, 7–2
    via the GUI, 7–3
selecting logic, 7–1
    used in postprocessing, 7–4
selecting subsets (see Selecting logic)
selection commands, 7–2
SEPC (structural percent error), 5–38
SETSPEED macro, 3–28
SFBEAM command, 2–14
SFCUM command, 2–18
SFFUN, 2–14
SFGRAD command, 2–15
SFSCALE command, 2–18
SFTRAN command, 2–18
/SHADE command, 11–4
SHELL61
    principal stresses on, 5–8
/SHOW command, 8–1, 16–2
/SHOWDISP command, 16–4
/SHRINK command, 11–4
singularities, 2–15, 3–29