11. APDL Basics
APDL Basics

Overview

• APDL is an acronym for ANSYS Parametric Design Language, a powerful scripting language that allows you to parameterize your model and automate common tasks.

• Using APDL, you can:
  – input model dimensions, material properties, etc. in terms of parameters rather than numbers.
  – retrieve information from the ANSYS database, such as a node location or maximum stress.
  – perform mathematical calculations among parameters, including vector and matrix operations.
  – define abbreviations (short cuts) for frequently used commands or macros.
  – create a macro to execute a sequence of tasks, with if-then-else branching, do-loops, and user prompts.
APDL Basics

...Overview

• The purpose of this chapter is to introduce you to basic APDL capabilities so that you are able to:
  – define scalar parameters and use them
  – retrieve information from the ANSYS database

• We will cover the following topics:
  A. Defining Parameters
  B. Using Parameters
  C. Retrieving Database Information
  D. Workshop
APDL Basics

A. Defining Parameters

• To define a parameter, use the format

\[ \text{Name} = \text{Value} \]

– Can be typed in the input window or in the Scalar Parameters dialog (Utility Menu > Parameters > Scalar Parameters...)

– Name is the parameter name, eight alphanumeric characters or less.

– Value may be a number, a previously defined parameter, a mathematical function, a parametric expression, or a character string.
### APDL Basics

**...Defining Parameters**

- **Examples:**
  - `inrad=2.5`
  - `outrad=8.2`
  - `numholes=4`
  - `thick=outrad-inrad`
  - `e=2.7e6`
  - `density=0.283`
  - `bb=cos(30)`
  - `pi=acos(-1)`
  - `g=386`
  - `massdens=density/g`
  - `circumf=2*pi*rad`
  - `area=pi*r**2`
  - `dist=sqrt((y2-y1)**2+(x2-x1)**2)`
  - `slope=(y2-y1)/(x2-x1)`
  - `theta=atan(slope)`
  - `jobname='proj1'`
APDL Basics

...Defining Parameters

• The examples above are *scalar* parameters, which have a single value — either numeric or character.

• ANSYS also supports *array* parameters, which have multiple values. Both numeric and character arrays are available. Array parameters will not be discussed in this course.

\[
xvalues = \begin{pmatrix} 28.7 \\ -9.2 \\ -2.1 \\ 51.0 \\ 0.0 \end{pmatrix} \quad \text{filnam} = \begin{pmatrix} \text{job1} \\ \text{job2} \\ \text{job3} \\ \text{job4} \\ \text{job5} \end{pmatrix}
\]
Some naming rules:

- Parameter names must be eight characters or less, beginning with a letter.

- Only letters, numbers, and the underscore character _ are allowed.

- Avoid underscore _ as starting character… reserved for ANSYS use.

- Names are not case-sensitive, i.e, “RAD” and “Rad” are the same. All parameters are internally stored in capital letters.

- Avoid common ANSYS labels such as STAT, DEFA, and ALL.
**APDL Basics**

**B. Using Parameters**

- To *use* a parameter, simply enter its name in the appropriate field in the dialog box or on the command.

- For example, to define a rectangle using the parameters $w=10$ and $h=5$,
  - you can use the menu:
    Preprocessor > Create > Rectangle > By 2 Corners +
  - or commands:
    `/prep7`
    `blc4,,,w,h`
Note:

• Whenever you use parameters, ANSYS immediately substitutes their values.

   The rectangle in the previous example is stored as a 10x5 area, not as w x h. That is, if you change the value of w or h after creating the rectangle, the area will NOT be updated.
APDL Basics

...Using Parameters

- Other examples of using parameters:
  
  ```plaintext
  jobname='proj1'
  /filnam,jobname ! Jobname
  /prep7
  ex=30e6
  mp,ex,1,ex ! Young's modulus
  force=500
  fk,2,fy,-force ! Force at KP 2
  fk,6,fx,force/2 ! Force at KP 6
  ```
APDL Basics

C. Retrieving Database Information

• To retrieve information from the database and assign it to a parameter, use the *GET command or Utility Menu > Parameters > Get Scalar Data...

• A vast amount of information is available, including model and results data. Refer to the *GET command description for details.


**APDL Basics**  
**...Retrieving Database Information**

- **Examples:**
  
  ```markdown
  *get,x1,node,1,loc,x ! x1 = X coordinate of node 1 [CSYS]*
  /post1
  *get,sx25,node,25,s,x ! sx25 = X stress at node 25 [RSYS]*
  *get,uz44,node,44,u,z ! uz44 = UZ displacement at node 44 [RSYS]*
  nsort,s,eqv ! Sort nodes by von Mises stress
  *get,smax,sort,,max ! smax = maximum of last sort
  etable,vol,volu ! Store element volumes as vol
  ssum ! Sum all element table columns
  *get,totvol,ssum,,vol ! totvol = sum of vol column
  ```

  *CSYS = In the active coordinate system (CSYS)*
  
  **RSYS = In the active results coordinate system (RSYS)**
Some data can be retrieved with a *get function*.

**Examples:**

\( x_1 = n_x(1) \)  
\( n_n = n_o_d_e(2.5,3,0) \)

\( /p_o_s t 1 \)

\( u_x25 = u_x(25) \)

\( t_e_m_p93 = t_e_m_p(93) \)

\( w_i_d t_h = d_i_s t_n_d(23,88) \)

*CSYS = In the active coordinate system (CSYS)*

*RSYS = In the active results coordinate system (RSYS)*
You can use a get function directly in a field, just like a parameter. For example:

- \( k,10,kx(1),ky(3) \) ! KP 10 at X of KP 1, Y of KP 3 [CSYS]*
- \( k,11,kx(1)\times2,ky(3) \) ! [CSYS]*
- \( f,node(2,2,0),fx,100 \) ! FX force at node(2,2,0) [CSYS]*

*CSYS = In the active coordinate system (CSYS)
Summary:

- Define parameters using the format $Name=Value$.

- $Value$ may be a number, a previously defined parameter, a mathematical function, a parametric expression, or a character string.

- Use *GET or get functions to retrieve data from the ANSYS database.

- ANSYS stores data in their actual form (numbers or strings), not in the form of parameter names.
APDL Basics

D. Workshop

- Refer to your Workshop Supplement for instructions on:
  
  W10. 2-D Bracket Using Parameters