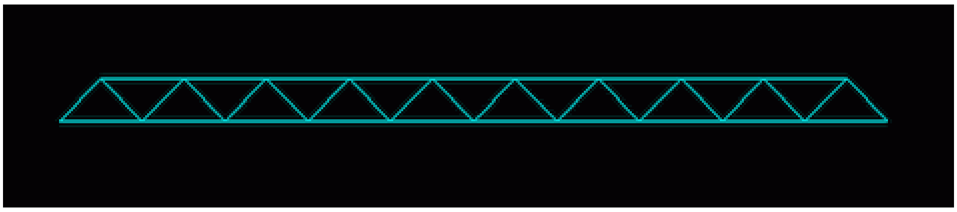
# ANSYS Parametric Design Language (APDL)

## Introduction

This tutorial was completed using ANSYS 7.0 The purpose of this tutorial is to familiarize the user with the ANSYS Parametric Design Language (APDL). This will be a very basic introduction to APDL, covering things like variable definition and simple looping. Users familiar with basic programming languages will probably find the APDL very easy to use. To learn more about APDL and see more complex examples, please see the APDL Programmer's Guide located in the help file.

This tutorial will cover the preprocessing stage of constructing a truss geometry. Variables including length, height and number of divisions of the truss will be requested and the APDL code will construct the geometry.



## Preprocessing: Use of APDL

Shown below is the APDL code used to construct the truss shown above, using a length of 200 m, a height of 10 m and 20 divisions. The following discussion will attempt to explain the commands used in the code. It is assumed the user has been exposed to basic coding and can follow the logic.

```
finish
/clear

/prep7

*ask,LENGTH,How long is the truss,100
*ask,HEIGHT,How tall is the truss,20
*ask,DIVISION,How many cross supports even number,2

DELTA_L = (LENGTH/(DIVISION/2))/2
NUM_K = DIVISION + 1
COUNT = -1
X_COORD = 0

*do,i,1,NUM_K,1

COUNT = COUNT + 1
OSCILATE = (-1)**COUNT

X_COORD = X_COORD + DELTA_L

*if,OSCILATE,GT,0,THEN
k,i,X_COORD,0

*else
k,i,X_COORD,HEIGHT

*endif

*enddo


KEYP = 0

*do,j,1,DIVISION,1
KEYP = KEYP + 1
```

```
L,KEYP,(KEYP+1)

*if,KEYP,LE,(DIVISION-1),THEN
L,KEYP,(KEYP+2)
*endif
*enddo

et,1,link1
r,1,100
mp,ex,1,200000
mp,prxy,1,0.3

esize,,1
lmesh,all

finish
```

1. ***ASK Command**

   The ***ASK** command prompts the user to input data for a variable. In this case, `*ask,LENGTH,How long is the truss,100` prompts the user for a value describing the length of the truss. This value is stored under the variable **LENGTH**. Thus in later parts of the code, LENGTH can be used in other commands rather than typing in 200 m. The 100 value at the end of the string is the default value if the user were to enter no value and just hit the enter key.

2. **Variable Definition Using the "=" Command**

   ANSYS allows the user to define a variable in a few ways. As seen above, the *ASK command can be used define a variable, but this is usually only used for data that will change from run to run. The *SET command can also be used to define variables. For more information on this command, see the help file. However, the most intutitive method is to use "=". It is used in the following manner: 'the variable you wish to define' = 'some arguement'. This argument can be a single value, or a mathematical expression, as seen in the line defining `DELTA_L`

3. ***DO Loops**

   Do-loops are useful when you want to repeat a command a known number of times. The syntax for the expression is *DO, Par, IVAL, FVAL, INC, where Par is the parameter that will be incremented by the loop, IVAL is the initial value the parameter starts as, FVAL is the final value the parameter will reach, and INC is the increment value that the parameter will be increased by during each iteration of the loop. For example, `*do,i,1,10_K,1` is a do-loop which increases the parameter "i" from 1 to 10 in steps of 1, (ie 1,2,3...8,9,10). It is necessary to use a ***ENDDO** command at the end of the loop to locate where ANSYS should look for the next command once the loop has finished. In between the *DO and *ENDDO, the user can place code that will utilize the repetative characteristics of the loop.

4. ***IF Statement**

   If-statements can be used as decision makers, determining if a certain case has occured. For example, in the code above there is a statement: `*if,OSCILATE,GT,0,THEN`. This translates to "if the variable, OSCILATE, is greater than zero, then...". Any code directly following the *if command will be carried out if the statement is true. If it is not true it will skip to the ***else** command. This command is only used in conjunction with the *if command. Any code directly following the *else command will be carried out when the original statement is false. An ***endif** command is necessary after all code in the *if and *else sections to define an ending.

**Reference**

http://www.mece.ualberta.ca/tutorials/ansys