Разработал Овчаренко В.В.

Лекция №14

- Параметры-массивы ANSYS.
- Команды APDL.

Параметры-массивы ANSYS

Параметры-массивы (далее массивы) в ANSYS определяются командой

- *DIM, Par, Type, IMAX, JMAX, KMAX, Var1, Var2, Var3

  Здесь Par – это идентификатор создаваемого массива.

  Параметр Type определяет тип массива, и может принимать такие значения:

  - ARRAY (значение по умолчанию) – соответствует 3-хмерной матрице с вещественными элементами, IMAX, JMAX, KMAX – определяют количество строк,

столбцов и «плоскостей» матрицы.

Одномерная матрица (вектор) считается частным случаем трехмерной, при JMAX=1, KMAX=1. Двумерная матрица – аналогично, при KMAX=1. Значение по умолчанию для IMAX, JMAX, KMAX – 1.

Например, двумерная матрица z (2×3) может быть определена

```
*dim,z,array,2,3,1
```

или

```
*dim,z,,2,3
```

Доступ к элементу матрицы осуществляется через индексы, которые указываются в круглых скобках после названия массива. В качестве индексов матрицы используются целые числа (вещественные значения округляются). Индексация начинается с 1. Полный список индексов состоит из 3 значений, разделяемых запятой. В сокращенном списке могут быть опущены последние несколько индексов, в этом случае они считаются равными 1.

Например, записи

```
z(1,2,1)=12.7
k=z(1,2)          !k = 12.7
```

И

```
z(1,2)=12.7
k=z(1,2,1)        !k = 12.7
```

полностью эквивалентны.

- **CHAR – соответствует 3-хмерной матрице со строчными (до 8 символов) элементами. Принципы использования аналогичны ARRAY.**

- TABLE – соответствует 3-хмерной табличной функции.

  IMAX, JMAX, KMAX – количество строк, столбцов и «плоскостей» табличной функции.

  Var1, Var2, Var3 – определяют названия аргументов функции.

  Принципы доступа к значениям функции при чтении и при записи различны.

  Величины в круглых скобках после имени функции трактуются:

- при записи, как целочисленные индексы, определяющие положение ячейки табличной функции.
- при чтении, как вещественные аргументы функции, при этом если значения аргументов не совпадают с табличными, то производится вычисление значения функции при помощи линейной интерполяции.

При определении значений аргумента Var1 функции, они должны быть записаны в соответствующие ячейки 0-го столбца,

Var2 - 0-й строки, Var3 - 0-го столбца и 0-й строки (изменяется k).

Следующий код иллюстрирует работу с элементарной табличной функцией с одним аргументом:

```
*dim,sf,table,3
sf(1,0)=0.1
sf(2,0)=0.4
sf(3,0)=0.7
sf(1)=1.1
sf(2)=4.4
sf(3)=7.7
k1=sf(0.1)
k2=sf(0.2)
```

```
k3=sf(0.3)
k4=sf(0.4)
k5=sf(0.5)
k6=sf(0.6)
k7=sf(0.7)
```

в результате параметры k1-k7 получат такие значения

```
k1=1.1
k2=2.2
k3=3.3
k4=4.4
k5=5.5
k6=6.6
k7=7.7
```

- **STRING – соответствует двумерной матрице со строчными (до IMAX символов) элементами. По характеру использования подобно двумерной матрице указателей на символ (т.е. строк в формате C). Первый индекс определяет номер символа строки.**

Например, выполнение кода

```
*dim,tx,string,40,4
tx(1,1)='Array entries are character'
tx(1,2)='strings (up to IMAX each).'
tx(1,3)='Index numbers for columns and'
tx(1,4)='planes are sequential values.'
```

```
z1=tx(1,2)
z2=tx(15,2)
tx(15,2)='ZZZZZ'
z3=tx(1,2)
z4=tx(15,2)
```

приведет к тому, что параметры z1-z4 получат значения

```
z1='strings (up to IMAX each).'
z2=' IMAX each).'
z3='strings (up toZZZZZ'
z4='ZZZZZ'
```

Команды APDL

Команды APDL являются аналогом операторов в большинстве языков программирования и предназначены для управления процессом выполнения макроса.

- *GO, Base

  Команда реализует функциональность оператора безусловного перехода.

  Параметр Base определяет выполняемое действие:

  :label – переход на определенную

пользователем метку. label – определяет идентификатор метки.

STOP – завершение работы (и в пакетном, и в интерактивном режимах).

- *IF, VAL1, Oper, VAL2, Base

  Команда реализует функциональность условного оператора. VAL1 и VAL2 – характеризуют сравниваемые значения.

  Параметр Oper определяет операцию сравнения.

Он может принимать следующие значения:

EQ – равенство (VAL1 = VAL2);
NE – неравенство (VAL1 ≠ VAL2);
LT – меньше (VAL1 < VAL2);
GT – больше (VAL1 > VAL2);
LE – меньше или равно (VAL1 ≤ VAL2);
GE – больше или равно (VAL1 ≥ VAL2);
ABLT – меньше по модулю
(| VAL1 | < | VAL2 |);
ABGT – больше по модулю
(| VAL1 | > | VAL2 |).

Параметр Base определяет действие выполняемое если условие VAL1, Oper, VAL2 выполняется:

:label – переход на определенную пользователем метку. label – определяет идентификатор метки;

STOP – завершение работы (в пакетном режиме);

EXIT – выход из цикла;

CYCLE – переход к следующему шагу цикла;

THEN – используется для конструкции if-then-else.

Конструкция if-then-else описывается следующим образом

```
*IF,VAL1,Oper,VAL2,THEN
----
*ELSEIF,VAL1,Oper,VAL2
----
*ELSEIF,VAL1,Oper,VAL2
----
*ELSE
----
*ENDIF
```

Примеры использования команды *IF при Base = THEN

```
/PREP7
*if,arg1,eq,0,then
BLC4,0,0,1,1,1
*elseif,arg1,eq,1
BLC4,0,0,2,1,1
*else
BLC4,0,0,3,1,1
*endif
```

и Base = :label

```
/PREP7
*if,arg1,eq,0,:lb
BLC4,0,0,1,1,1
:lb
BLC4,1,1,2,1,1
```

- *DO, Par, IVAL, FVAL, INC

  Команда реализует функциональность параметрического цикла.

  Par определяет параметр, который является счетчиком цикла.

  IVAL – начальное значение счетчика.

  FVAL – конечное значение счетчика.

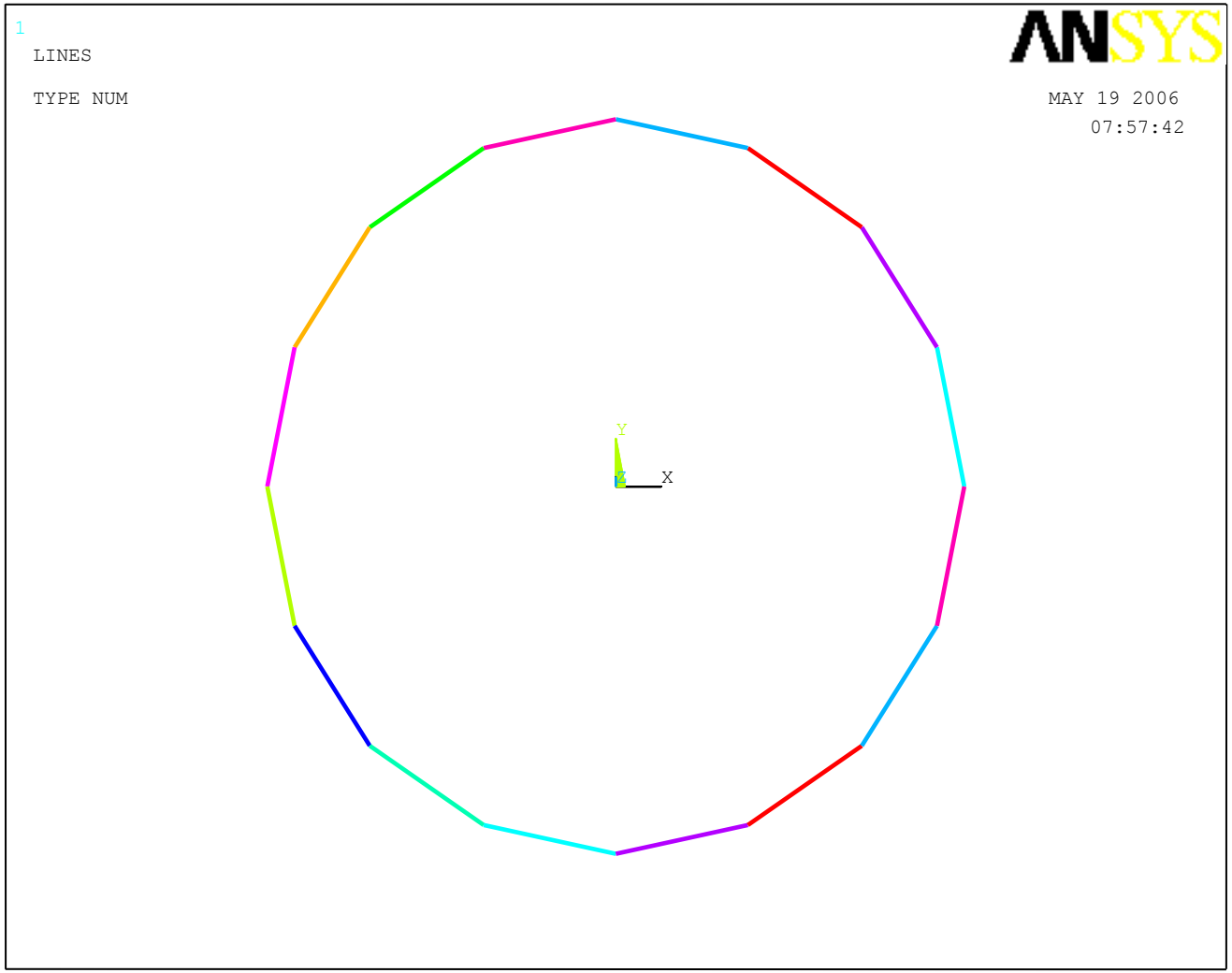
  INC – приращение счетчика на каждом шаге цикла (по умолчанию = 1).
  IVAL, FVAL, INC – вещественные значения.

  Тело цикло завершается командой *ENDDO.

Например, код

```
/PREP7
pi=4*atan(1)
t=0
*do,i,0,2*pi,pi/8
t=t+1
k,t,cos(i),sin(i),0
*enddo
*do,j,1,t-1
l,j,j+1
*enddo
```

выполняет построение вершин и сторон правильного 16-тиугольника, вписанного в окружность радиусом 1 м.


1
LINES
TYPE NUM
ANSYS
MAY 19 2006
07:57:42
Y
Z
X

Команды APDL

- *CYCLE

  Команда реализует функциональность оператора continue в Pascal и C, т.е. осуществляет переход к следующему шагу цикла.

- *EXIT

  Команда реализует функциональность оператора break в Pascal и C, т.е. осуществляет выход из цикла.