

## Основы командного языка APDL ANSYS

Программа на командном языке APDL ANSYS представляет собой последовательность команд, записанных в обычном тестовом файле.

Поскольку ANSYS является интерпретатором, то команды выполняются последовательно с учетом операторов изменений в порядке выполнения, задаваемыми операторами цикла и перехода. Если же при выполнении программы на языке APDL ANSYS возникает ошибка, то выдается сообщение об ошибке, и, как правило, программа дальше работает некорректно и в дальнейшем приводит к останову.

Язык APDL ANSYS похож на язык FORTRAN, который сейчас, однако, не так хорошо известен и популярен, как ранее.

Команда APDL ANSYS записывается в отдельной строке, причем максимальное число символов не должно превышать 80. Команда состоит из *имени команды* и набора *аргументов*, или *операндов*, отделенных друг от друга и от имени команды запятыми. Строчные и заглавные буквы эквиваленты.

Имена команд могут состоять из набора символов от 1 до 8. Стандартные команды APDL имеют зарезервированные имена, из которых первые 4 символа являются значимыми (не считая начальных символов / и \*).

Таким образом, например, записи **/SOLU**, **/SOLUT** и **/SOLUTION** эквиваленты.

Некоторые команды являются макрокомандами (т.е. состоят из последовательности надлежащим образом оформленных команд APDL). Ряд макрокоманд входят в комплект поставки программного обеспечения ANSYS, но макрокоманды могут быть созданы и пользователем. Имена макрокоманд должны вводиться полностью без сокращений.

Формат ввода команд свободный, т.е. пробелы не учитываются. При вводе команд применяется правило умолчания. По этому правилу, если какой-то из аргументов не задан явно, программа пытается присвоить этому аргументу значение по умолчанию.

Например, ANSYS всегда работает с объектами, задаваемыми в трехмерном пространстве. Но если решается плоская задача, то третья координата не используется и по умолчанию равна нулю. Аналогично, для одномерных задач не нужна и вторая координата. Поэтому, если аргументы координат не заданы, то ANSYS принимает их равными значениям по умолчанию. Эти значения указываются для каждой команды в Руководстве по командам ANSYS (Command Reference). В большинстве случаев (но не всегда!) эти значения равны нулю.

Поэтому команды **K,1,10,0,2** и **K,1,10,,2** эквиваленты; команда **N,2,3.5**, в действительности означает, что узлу с номером 2 присвоены координаты (3.5,0,0) в соответствующей системе координат, и т.д.

Правила умолчания приняты для большинства команд ANSYS, являются обычно «наиболее естественно разумными», позволяя часто не вводить большого числа аргументов.

Несколько команд можно записать в одной строке, отделяя их друг от друга знаком \$, например:

```
L,1,2 $ L,2,3 $ L,3,4
```

Некоторые команды начинаются со знака «/», и обычно предназначены для управления основной программой и контроля. Есть еще команды, начинающиеся со знака «\*». Обычно это и есть команды собственно языка APDL.

Комментарии начинаются со знака «!» или, в начале строки, с символов **/COM**, или **C\*\*\***, . Отличие состоит в том, что комментарии, записанные после **/COM**, или **C\*\*\***, будут содержаться и файле вывода ANSYS.

В APDL ANSYS удобно использовать параметры или переменные. При определении параметров в ANSYS не требуется указывать их тип, за исключением многомерных данных в форме массивов.

Все численные переменные имеют целый или действительный тип и хранятся с двойной точностью. Если параметру не присвоено никакое значение, то считается, что он равен нулю. (Поэтому, если разделить на неопределенный ранее параметр какое-либо число, то программа выдаст ошибку.) Имеются также символьные параметры, которые используются для задания имен файлов, строк вывода, заголовков и пр. Значения этих параметров должны быть заключены в одинарные апострофы. Буквы кириллицы в них не допускаются.

Массивы, которые перед использованием требуется обязательно определять, могут иметь численный, текстовый или табличный типы.

Имена параметров должны начинаться с буквы и содержать не более 32 знаков, включающих только буквы, цифры и символ подчеркивания. Желательно, чтобы имя параметра не совпадало с зарезервированными именами ANSYS для команд, степеней свободы, ключевых слов и т.п.

Для определения параметров ANSYS предлагает использовать или команду **\*SET**, или обычный оператор равенства. Последняя возможность кажется наиболее естественной. Таким образом, синтаксис оператора задания параметра выглядит следующим образом:

*Name = Value*

где *Name* – имя параметра, *Value* – его значение, которое может быть численной величиной, строкой (в апострофах), именем другого параметра, математическим параметрическим выражением или функцией. Примеры задания параметров приведены ниже:

HL=0.4

F\_R='Mod\_ANS\_1'

RO1=7.86e3

PEL=(HAL/2)/(WL-WWL)

Для определения параметров, связанных с величинами из базы данных ANSYS, содержащей описание твердотельных и конечно-элементных моделей, чрезвычайно полезна команда **\*GET**. Описание этой команды занимает несколько страниц фирменного руководства. Следующие примеры иллюстрируют примеры использования этой важной команды:

**PLNSOL,U,X ! Вывод перемещений по оси X**

**\*GET,UXMAX,PLNSOL,0,MAX ! UXMAX - максимальные перемещения по оси X**

**\*GET,EL\_MAX,ELEM,,COUNT ! EL\_MAX - число элементов**

**\*GET,ND\_MAX,NODE,,COUNT ! ND\_MAX - число узлов**

**ETAB,S11,S,X**

**\*get,c11\_el,elem,ii,etab,s11**

Некоторые из полезных вариантов использования команды **\*GET** реализованы также в сокращенной форме GET-функций.

Параметрические выражения APDL ANSYS имеют точно такой же синтаксис, как в языке FORTRAN. При этом допустимыми операторами являются следующие: сложение (+), вычитание (-), умножение (\*), деление (/), возведение в степень (\*\*), сравнение «меньше чем» (<) и сравнение «больше чем» (>).

Применяется стандартный порядок выполнения операций:

- 1) действия внутри скобок;
- 2) возведение в степень (в порядке слева направо);
- 3) умножение и деление (в порядке слева направо);
- 4) сложение и вычитание (в порядке слева направо);

5) логическое равенство (в порядке слева направо).

Естественно, чтобы быть уверенным в правильности порядка выполнения операций, нужно просто использовать скобки и разбивать сложные выражения на более простые.

Некоторые математические функции, допустимые в программе ANSYS, собраны в таблице.

Обозначение функции	Действие
ABS(x)	Абсолютное значение величины x.
SIGN(x,y)	Абсолютное значение x со знаком y. При y=0 результат неотрицателен.
EXP(x)	Экспонента от x ( $e^x$ ).
LOG(x)	Натуральный логарифм x ( $\ln(x)$ ).
LOG10(x)	Десятичный логарифм x ( $\log_{10}(x)$ ).
SQRT(x)	Квадратный корень из x.
NINT(x)	Ближайшее целое к x.
MOD(x,y)	Остаток от частного x/y. При y=0 возвращается значение (0).
RAND(x,y)	Случайная величина (имеющая постоянное распределение) в пределах от x до y (x = нижняя граница, y = верхняя граница).
GDIS(x,y)	Случайная величина с Гауссовым (нормальным) распределением со средним значением x и стандартным отклонением y.
SIN(x), COS(x), TAN(x)	Синус, косинус и тангенс x. По умолчанию x задается в радианах, но единицы измерения x могут быть изменены командой *AFUN.

Помимо скалярных параметров в программах ANSYS могут использоваться массивы параметров. Массивы могут иметь размерности от 1 до 5 и иметь следующие типы: ARRAY (целые и действительные числа), CHAR (текстовый), TABLE и STRING (строковый).

Тип TABLE является специальным типом числового массива с автоматическим выполнением процедуры линейной интерполяции между элементами массива. Массивы определяются командой \*DIM.

Для изменения порядка выполнения команд в программе APDL ANSYS предлагает стандартные возможности программирования:

- оператор безусловного перехода \*GO
- условный оператор (команда \*IF)
- оператор повторения (команда \*REPEAT)
- оператор цикла (команда \*DO)
- дополнительное управление в операторе цикла (команда \*DOWHILE)

Оператор безусловного перехода может быть использован следующим образом:

```
*GO, :LABEL1
---
---
:LABEL1
---
```

Оператор условного перехода \*IF имеет следующий синтаксис:  
\*IF, VAL1, Oper, VAL2, Base

где

**VAL1** – первое сравниваемое выражение;

**Oper** – оператор сравнения, задаваемый ключевым словом;

**VAL1** – второе сравниваемое выражение;

**Base** – ключевое слово, определяющее действие, которое будет совершаться при выполнении условия

Операторы сравнения **Oper** могут быть следующими: **EQ** – логическое равенство; **NE** – логическое неравенство; **LT** – логическое меньше чем; **GT** – логическое больше чем; **LE** – логическое меньше или равно; **GE** – логическое больше или равно; **ABLT** – логическое меньше по абсолютному значению; **ABGT** – логическое больше по абсолютному значению.

Значением **Base** может быть ключевое слово **THEN**, что превращает команду **\*IF** в конструкцию **IF\_THEN-ELSE**.

После выполнения блока, следующего за **THEN**, могут располагаться ключевые слова: **\*ELSEIF** (может отсутствовать) и следующий за ним блок; **\*ELSE** (может отсутствовать) и следующий за ним блок; **\*ENDIF** (должен присутствовать обязательно).

Следующий пример иллюстрирует данную конструкцию:

```
*IF,A,GT,1,THEN  
  ! Block 1  
  ---  
  ----  
  *ELSEIF,A,GT,0  
  ! Block 2  
  ---  
  ----  
  *ELSEIF,A,EQ,0  
  ! Block 3  
  ---  
  ----  
  *ELSE  
  ! Block 4  
  ---  
  ----  
*ENDIF
```

Оператор цикла **\*DO** предназначен для выполнения блока команд определенное число раз. Этот блок команд должен быть заключен между командами **\*DO** и **\*ENDDO**. Следующий пример иллюстрирует возможность применения оператора цикла:

```
*DO,N,1,10  
  ---  
  ---  
*ENDDO
```

Таким образом, язык APDL является вполне развитым средством программирования, позволяющим вместе с набором команд ANSYS создавать пользовательские программы, реализующие различные типы конечно-элементного анализа.