



Основы работы с системой компьютерной алгебры Scilab

1. Текстовые комментарии

Текстовый комментарий в Scilab — это строка, начинающаяся с символов `//`. Использовать текстовые комментарии можно как в рабочей области, так и в тексте файла-сценария. Строка после символов `//` не воспринимается как команда, и нажатие клавиши Enter приводит к активизации следующей командной строки:

```
--> //6+8
-->
```

2. Элементарные математические выражения

Для выполнения простейших арифметических операций в Scilab применяют следующие операторы: `+` сложение, `-` вычитание, `*` умножение, `/` деление слева направо, `\` деление справа налево, `^` возведение в степень. Вычислить значение арифметического выражения можно, если ввести его в командную строку и нажать клавишу Enter. В рабочей области появится результат:

```
--> 2.35*(1.8-0.25)+1.34^2/3.12
ans =
    4.2180
```

Если вычисляемое выражение слишком длинное, то перед нажатием клавиши Enter следует набрать три или более точек. Это будет означать продолжение командной строки:

```
--> 1+2+3+4+5+6....
7+8+9+10+....
+11+12+13+14+15
ans =
    120
```

Если символ точки с запятой «`;`» указан в конце выражения, то результат вычислений не выводится, а активизируется следующая командная строка:

```
--> 1+2;
--> 1+2
ans =
    3
```

3. Переменные в Scilab

В рабочей области Scilab можно определять переменные, а затем использовать их в выражениях. Любая переменная до использования в формулах и выражениях должна быть определена. Для определения переменной необходимо набрать имя переменной, символ «`=`» и значение переменной. Здесь знак равенства — это оператор присваивания, действие которого не отличается от аналогичных операторов языков программирования. Т. е., если в общем виде оператор присваивания записать как

имя_переменной = значение_выражения

то в переменную, имя которой указано слева, будет записано значение выражения, указанного справа. Имя переменной не должно совпадать с именами встроенных процедур, функций и встроенных переменных системы и может содержать до 24 символов. Система различает большие и малые буквы в именах переменных. Т. е. ABC, abc, Abc, aBc — это имена разных переменных. Выражение в правой части оператора присваивания может быть числом, арифметическим выражением, строкой символов или символьным выражением. Если речь идет о символьной или строковой переменной, то выражение в правой части оператора присваивания следует брать в одинарные кавычки. Если символ «`;`» в конце выражения отсутствует, то в качестве результата выводится имя переменной и ее значение. Наличие символа «`;`» передает управление следующей командной строке. Это позволяет использовать имена переменных для записи промежуточных результатов в память компьютера:

```
--> a=2.3
a = 2.3000
--> b=-34.7
b = -34.7000
--> x=1;y=2; z=(x+y)-a/b
z = 3.0663
```

4. Системные переменные Scilab

Если команда не содержит знака присваивания, то по умолчанию вычисленное значение присваивается специальной системной переменной `ans`. Причем полученное значение можно использовать в последующих вычислениях, но важно помнить, что значение `ans` изменяется после каждого вызова команды без оператора присваивания:

```
--> 25.7-3.14
ans = 22.5600
--> 2*ans
ans = 45.1200
```

Другие системные переменные в Scilab начинаются с символа `%`:

`%i` — мнимая единица ($\sqrt{-1}$); `%pi` — число $\pi = 3.141592653589793$; `%e` — число $e = 2.7182818$; `%inf` — машинный символ бесконечности (∞); `%NaN` — неопределенный результат ($0/0, \infty/\infty$ и т. п.); `%eps` — условный ноль `%eps=2.220E-16`.

5. Ввод вещественного числа и представление результатов вычислений

Числовые результаты могут быть представлены с плавающей (например, $-3.2E-6$, $-6.42E+2$) или с фиксированной (например, 4.12 , 6.05 , -17.5489) точкой. Числа в формате с плавающей точкой представлены в экспоненциальной форме $mE\pm r$, где m — мантисса (целое или дробное число с десятичной точкой), r — порядок (целое число). Для того, чтобы перевести число в экспоненциальной форме к обычному представлению с фиксированной точкой, необходимо мантиссу умножить на десять в степени порядка. Например, $-6.42E+2 = -6.42 \cdot 10^2 = -642$ или $3.2E-6 = 3.2 \cdot 10^{-6} = 0.0000032$

При вводе вещественных чисел для отделения дробной части используется точка.

Примеры ввода и вывода вещественных чисел:

```
--> 0.123
ans = 0.123
```

Рассмотрим пример вывода значения системной переменной `pi` и некоторой переменной `q`, определенной пользователем:

```
--> %pi
%pi = 3.1415927
```

Нетрудно заметить, что Scilab в качестве результата выводит только восемь значащих цифр. Это формат вывода вещественного числа по умолчанию. Для того, чтобы контролировать количество выводимых на печать разрядов, применяют команду `printf` с заданным форматом, который соответствует правилам, принятым для этой команды в языке C.

6. Функции в Scilab

Все функции, используемые в Scilab, можно разделить на два класса:

- встроенные;
- определенные пользователем.

В общем виде обращение к функции в Scilab имеет вид:

имя_переменной = имя_функции(переменная1 [,переменная2, ...])

где имя_переменной — переменная, в которую будут записаны результаты работы функции; этот параметр может отсутствовать, тогда значение, вычисленное функцией, будет присвоено системной переменной `ans`; имя_функции — имя встроенной или ранее созданной пользователем функции; переменная1, переменная2, ... — список аргументов функции.

6.1. Элементарные математические функции

Пакет Scilab снабжен достаточным количеством всевозможных встроенных функций. Вот некоторые элементарные математические функции, используемые чаще всего: `sin(x)` — синус числа x , `cos(x)` — косинус числа x , `tan(x)` — тангенс числа x , `cotg(x)` — котангенс числа x , `asin(x)` — арксинус числа x , `exp(x)` — экспонента числа x , `sqrt(x)` — корень квадратный из числа x .

6.2. Функции, определенные пользователем

Функция, как правило, предназначена для неоднократного использования, она имеет входные параметры и не выполняется без их предварительного задания. Рассмотрим несколько способов создания функций в Scilab. Первый способ — это применение оператора `deff`, который в общем виде можно записать так:

`deff('имя1,...,имяN) = имя_функции(переменная_1,...,переменная_M)', 'имя1=выражение1;...;имяN=выражениеN')`

где `имя1,...,имяN` — список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений, `имя_функции` — имя с которым эта функция будет вызываться, `переменная_1,...,переменная_M` — входные параметры.

Второй способ создания функции это применение конструкции вида:

`function[имя1,...,имяN]=имя_функции(переменная_1,...,переменная_M)`

тело функции

`endfunction`

где `имя1,...,имяN` — список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений; `имя_функции` — имя с которым эта функция будет вызываться, `переменная_1, ..., переменная_M` — входные параметры.