

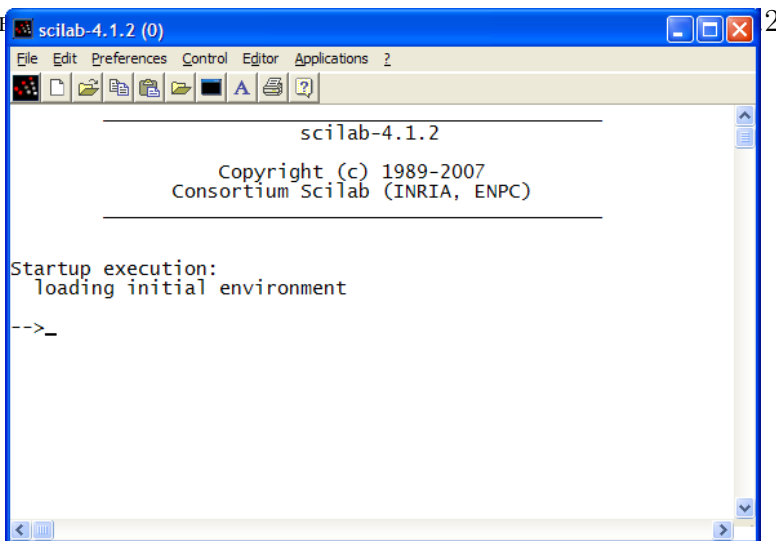
# 1. Лабораторная работа №1.

## Занятие №1.1. Основы работы в Scilab

### 1.1. Запуск Scilab

Запуск Scilab в операционных системах Linux и Windows немного отличается. Далее, для определенности, будем рассматривать работу с пакетом Scilab 4.1.2 в операционной системе Microsoft Windows XP.

Запуск Scilab производится с помощью выбора пунктов меню



Окно программы состоит из заголовка, меню, панели инструментов и рабочей области.

Заголовок программы содержит ее название и номер запущенного экземпляра программы (scilab-4.1.2(0)).

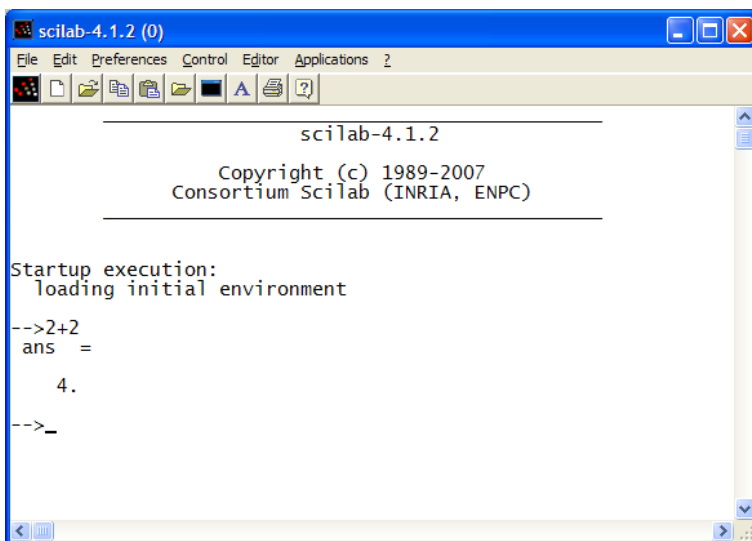
Меню содержит пункты File, Edit, Preferences, Control, Editor, Applications, ?.

Элементы панели инструментов дублируют отдельные пункты меню.

## 1.2. Простейшие вычисления в Scilab

Работа в Scilab производится с помощью команд, набираемых после приглашения `-->`. Выполнение команды осуществляется с помощью нажатия клавиши Enter.

Например, для вычисления  $2+2$  необходимо набрать  $2+2$  и нажать Enter.



Переменная `ans` (от англ. answer - ответ) содержит результат операции.

Если в конце команды стоит символ «;», то результат операции вычисляется, но не выводится на экран. Это удобно при вводе нескольких команд.

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

7.

-->

Длинные команды можно разделить на несколько строк. Для этого в конце строки нужно поставить две или более точки.

```
-->2+2. .  
-->+3+3. . .  
-->+10  
ans  =  
    20.  
-->
```

Для создания комментариев можно использовать команду `//`. Все, что написано справа от символов `//`, рассматривается как текст.

```
-->2*2 //comment  
ans  =  
    4.  
-->
```

Редактирование команд осуществляется с помощью клавиш управления курсором  $\leftarrow$ ,  $\rightarrow$ , клавиш Delete (удаление символа, стоящего после курсора),  $\leftarrow$ Backspace (удаление символа, стоящего перед курсором), Home (переход в начало строки) и End (переход в конец строки).

Клавиши управления курсором  $\uparrow$  и  $\downarrow$  позволяют пролистывать выполненные команды, а клавиши PageUp и PageDown — пролистывать рабочую область.

При вводе действительного числа в качестве разделителя целой и дробной части десятичного числа используется точка (например 55.777).

Для записи скобок используются круглые скобки ( ).

При вводе выражений можно использовать операторы:

+	— сложение;
-	— вычитание;
*	— умножение;
^	— возведение в степень;
/	— деление слева направо;
\	— деление справа налево.

Также можно использовать встроенные переменные `%pi` (число  $\pi$ ) и `%e` (число  $e$ ).

В пакете существует большое количество встроенных функций, которые можно использовать при записи выражений.

Приведем некоторые из них:

<code>sqrt(x)</code>	— квадратный корень из числа $x$ ;
<code>abs(x)</code>	— модуль числа $x$ ;
<code>sin(x)</code>	— синус числа $x$ ;
<code>cos(x)</code>	— косинус числа $x$ ;
<code>tan(x)</code>	— тангенс числа $x$ ;
<code>cotg(x)</code>	— котангенс числа $x$ ;
<code>asin(x)</code>	— арксинус числа $x$ ;
<code>acos(x)</code>	— арккосинус числа $x$ ;
<code>atan(x)</code>	— арктангенс числа $x$ ;
<code>exp(x)</code>	— экспонента числа $x$ ;
<code>log(x)</code>	— натуральный логарифм числа $x$ ;
<code>log10(x)</code>	— десятичный логарифм числа $x$ .

Подробное описание всех функций можно найти в справочной системе Scilab.

Заметим, что по умолчанию результат содержит восемь значащих цифр. Количество цифр, отображаемых в результате, можно изменить с помощью команды `printf`.

Например,

```
-->printf("%.5f",%pi)
3.14159
-->printf("%.10f",%pi)
3.1415926536
-->
```

**Пример.** Вычислить значение выражения

$$\frac{2 + \frac{3}{7} + \sin^3\left(\frac{\pi}{2}\right)}{5 \cdot \sqrt{224}}.$$

*Решение.*

```
-->(2+3/7+sin(%pi/2)^3)/(5*sqrt(224))
ans =
    0.0458162
-->
```

### 1.3. Переменные и функции

Для присваивания значений переменным используется оператор “=”. Например, чтобы присвоить  $x$  значение 5, нужно выполнить команду  $x=5$ .

```
-->x=5
x =
    5.
-->
```

Если в конце команды стоит символ «;», то значение переменной не выводится на экран.

```
-->x=5;
-->
```

В имени переменной учитывается регистр. Например,  $x$  и  $X$  — две разные переменные. Имя переменной не должно совпадать с названием встроенных переменных, процедур и функций. В имени переменной можно использовать не более 24 символов. Переменная может быть числового типа или строкового типа. Для присваивания переменной строки символов необходимо заключить эту строку в одинарные кавычки. Например, чтобы присвоить  $x = abc$ , нужно набрать `x='abc'`.

```
-->x='abc'
x  =
abc
-->
```

Команда `clear` позволяет очищать значения переменных. Например, команда `clear` очищает значения всех переменных, а команда `clear x` очищает значение переменной  $x$ .

Функции в Scilab делятся на встроенные и определенные пользователем. Некоторые встроенные функции были приведены выше. Рассмотрим определение пользовательских функций.

Пользовательские функции можно определить с помощью команды `deff` или конструкции `function ... endfunction`.

Формат команды `deff`:

```
deff('[y1,y2,...,yn]=название_функции(x1,x2,...,xn)',
    '[y1=выражение_1';
    'y2=выражение_2';
    ...;
    'yn=выражение_n']),
```

где  $x_1, x_2, \dots, x_n$  — входные переменные,  $y_1, y_2, \dots, y_n$  — выходные переменные, `выражение_1`, `выражение_2`, ... , `выражение_n` — выражения для вычисления  $y_1, y_2, \dots, y_n$ .

**Пример.** Определить функцию  $y = x^2$  и вычислить ее значение при  $x = 2$ .

```
-->deff(' [y]=myfunction(x)', 'y=x^2')
-->myfunction(2)
ans  =
    4.
-->
```

**Пример.** Определить функцию  $z = x^2 + y^2$  и вычислить ее значение при  $x = 4$  и  $y = 5$ .

```
-->deff(' [z]=myfunction1(x,y)', 'z=x^2+y^2')
-->myfunction1(4,5)
ans  =
    41.
-->
```

**Пример.** Определить функцию для решения уравнения  $x^2 = a(a \geq 0)$ .

```
-->deff(' [x1,x2]=myfunction2(a)', ...
-->'x1=sqrt(a);x2=-sqrt(a);')
-->[x1,x2]=myfunction2(4)
x2  =
    - 2.
x1  =
    2.
-->
```

Формат команды function ... endfunction:

function[y1,y2,...,yn]=название\_функции(x1,x2,...,xn)  
команды  
endfunction

или в одну строку

`function[y1,y2,...,yn]=название_функции(x1,x2,...,xn)`, команды, `endfunction`, где  $y_1, y_2, \dots, y_n$  - выходные параметры функции,  $x_1, x_2, \dots, x_n$  - входные параметры функции.

**Пример.** Определить функцию  $y = x^3$  и вычислить ее значение при  $x = 2$ .

```
-->function[y]=myfunction4(x)
-->y=x^3
-->endfunction
-->myfunction4(2)
ans =
    8.
-->
```

или

```
-->function[y]=myfunction4(x), y=x^3, endfunction
-->myfunction4(2)
ans =
    8.
-->
```

Для вызова функции, находящейся в другом файле, можно использовать функцию `exec(path)`.

При работе с функциями можно использовать локальные и глобальные переменные. Локальные переменные определены только внутри функции. Глобальные переменные определены в любом месте программы. Для определения глобальных переменных используется функция `global('var1',..., 'varn')`. Для уничтожения глобальных переменных используется функция `clearglobal('var1',..., 'varn')`. Для уничтожения всех глобальных переменных используется функция `clearglobal()`.



## 1.4. Построение графиков

### Графики функций одной переменной

Для построения графиков функций одной переменной в Scilab можно использовать функцию `plot2d(x,y,options)`, где  $x$  — вектор значений аргумента функции,  $y$  — вектор значений функции, `options` — параметры графика.

**Пример.** Построить график функции  $y = x^2$  на отрезке  $[-10, 10]$  с шагом изменения аргумента  $h = 0.001$ .

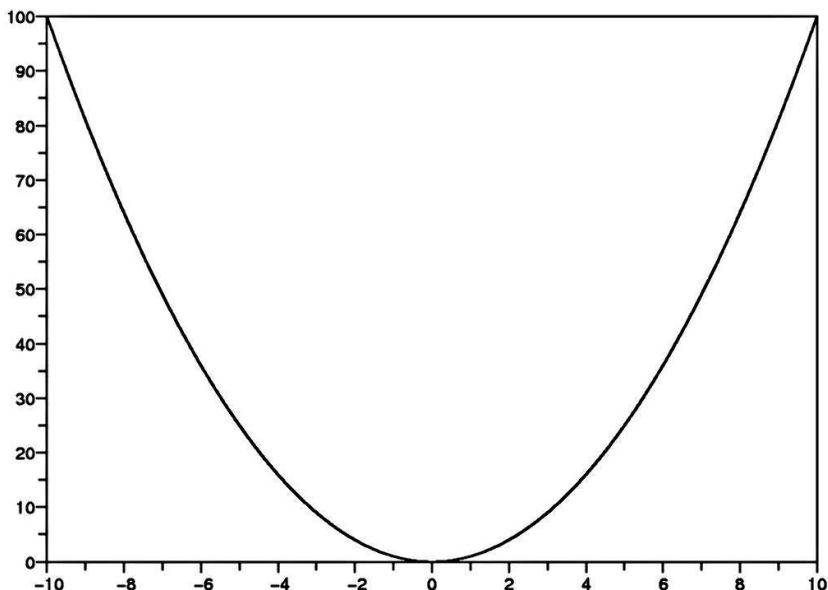
```
-->x=[-10:0.001:10]';
```

```
-->y=x^2;
```

```
-->plot2d(x,y)
```

```
-->
```

В результате выполнения этих команд открывается графическое окно, содержащее график



Полученный график можно сохранить в графический файл, выбрав пункт меню File—Save или File—Export.

Рассмотрим подробнее параметры настройки графика `options`. `Options` представляет собой последовательность выражений `параметр1=значение1`, `параметр2=значение2`, ..., где `параметр1`, `параметр2`, ... может быть одним из следующих:

`style` — устанавливает стиль графика (например, команда `plot(x,y,style=3)` делает цвет линии графика зеленым). Если изображены графики нескольких функций, то стилем `i`-графика управляет параметр `style(i)`;

`rect` — устанавливает размеры графического окна вокруг графика. Значение должно быть вектором вида `[xmin,ymin,xmax,ymax]`;

`logflag` — устанавливает масштаб оси (линейный или логарифмический). Значение должно быть строкой с возмож-

ными значениями 'nn', 'nl', 'ln', 'll'. Первая буква отвечает за масштаб оси абсцисс, а вторая — оси ординат (н-обычный, l-логарифмический);

frameflag — устанавливает параметры графической области. Может принимать целые значения от 0 до 8;

axesflag — устанавливает стиль осей координат. Может принимать целые значения от 0 до 5. (0 — оси координат не отображаются, 1 — ось ординат слева, 2 — график заключен в прямоугольник без числовых меток, 3 — ось ординат справа, 4 — оси координат в середине графической области, 5 — оси координат проходят через точку (0,0) (если она отсутствует, то не отображаются));

пах — используется для изменения меток на осях координат (требуется, чтобы axesflag=1). Значение должно быть вектором [px,Nx,py,Ny], где px,py — число делений на осях абсцисс и ординат (без цифр), Nx,Ny — число делений с цифрами на осях;

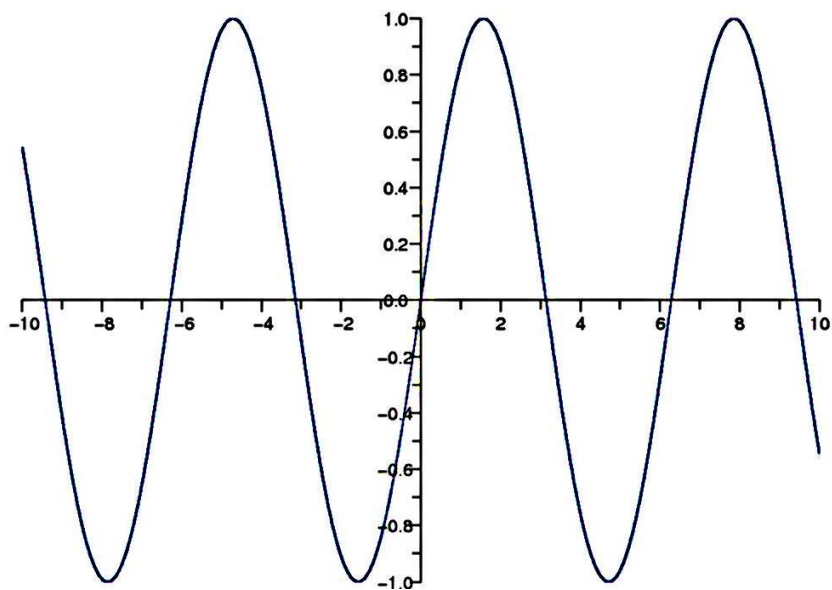
leg — используется для создания заголовков линий. Значение должно быть строкой вида 'leg1@leg2@...', где leg1, leg2, ... -заголовки линий.

Для очистки графической области служит команда clf(). Для изображения линий сетки служит команда xgrid().

### **Пример.**

```
-->x=[-10:0.001:10]';  
  
-->y=sin(x);  
  
-->plot2d(x,y,style=2,axesflag=5)  
  
-->
```

Результат:



Пример.

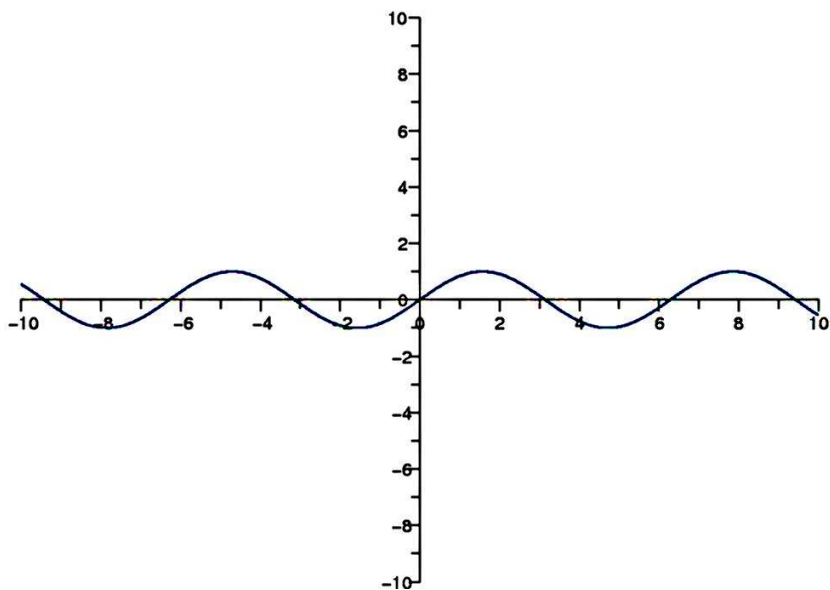
```
-->x=[-10:0.001:10]';
```

```
-->y=sin(x);
```

```
-->plot2d(x,y,style=2,axesflag=5,rect=[-10,-10,10,10])
```

```
-->
```

Результат:



Пример.

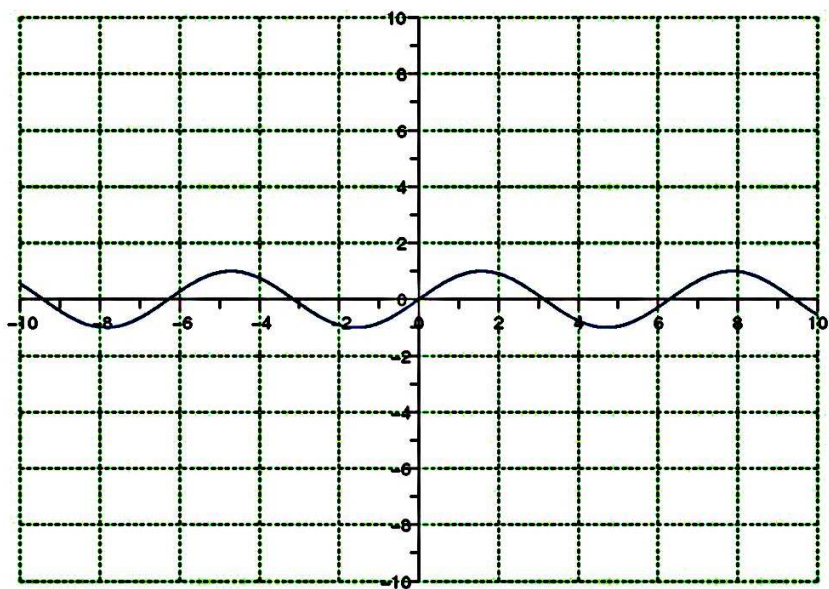
```
-->x=[-10:0.001:10]'; y=sin(x);
```

```
-->plot2d(x,y,style=2,axesflag=5,rect=[-10,-10,10,10])
```

```
-->xgrid(3)
```

```
-->
```

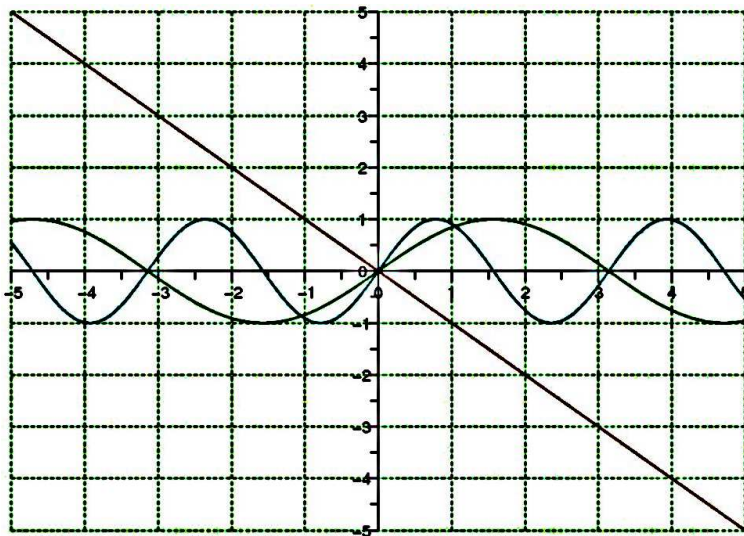
Результат:



Пример.

```
-->x=[-5:0.001:5]';  
-->y1=sin(x);  
-->y2=sin(2*x);  
-->y3=-x;  
  
-->plot2d(x,[y1 y2 y3],style=[3 4 5],...  
-->axesflag=5,rect=[-5,-5,5,5])  
  
-->xgrid(3)  
-->
```

Результат:

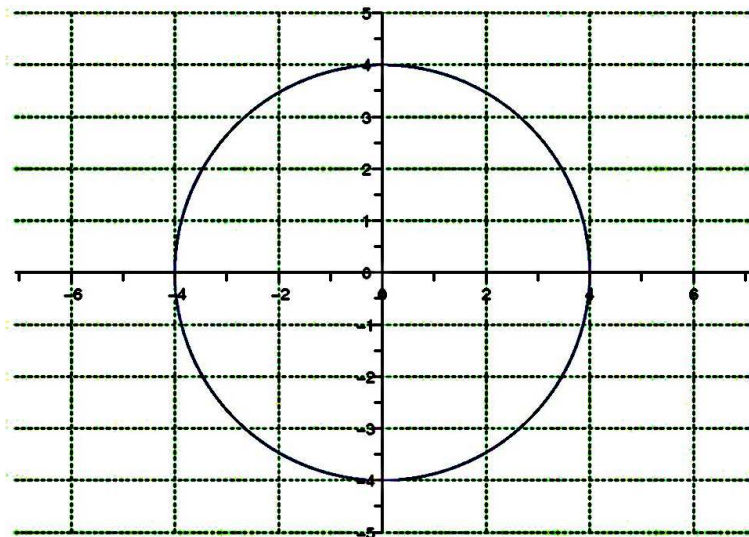


Построение графика функции,  
заданной параметрически

Пример.

```
-->t=0:0.01:2*pi;  
-->r=4;  
-->x=r*sin(t);  
-->y=r*cos(t);  
-->plot2d(x,y,style=2,frameflag=3,...  
-->rect=[-5,-5,5,5],axesflag=5)  
-->xgrid(3)  
-->
```

Результат:



### Построение графика функции в полярной системе координат

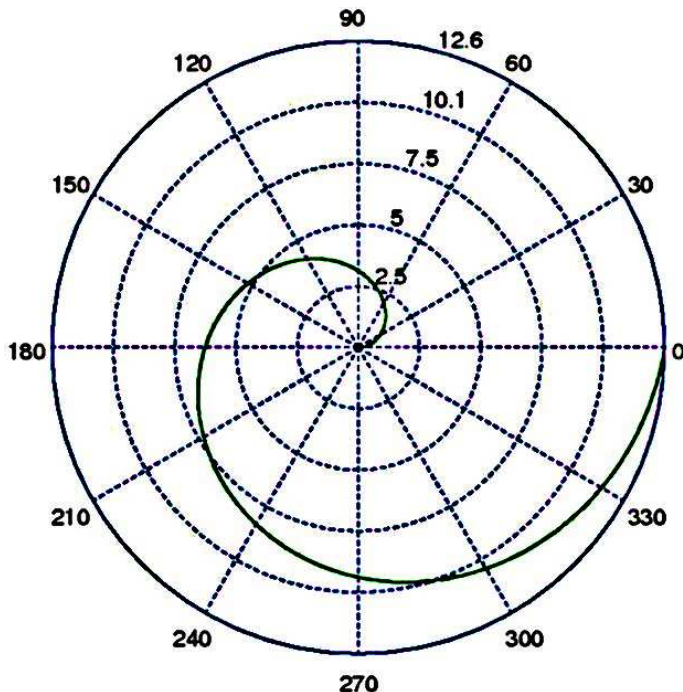
Для построения графика в полярной системе координат используется функция `polarplot(phi,rho,options)`, где `rho` — вектор, содержащий значения радиуса, `phi` — вектор со значениями угла, `options` — параметры графика.

#### Пример.

```
-->phi=0:0.01:2*%pi;  
-->rho=2*phi;  
-->clf(); xset("color",2);  
-->polarplot(phi,rho,style=3,frameflag=3)  
-->
```



Результат:



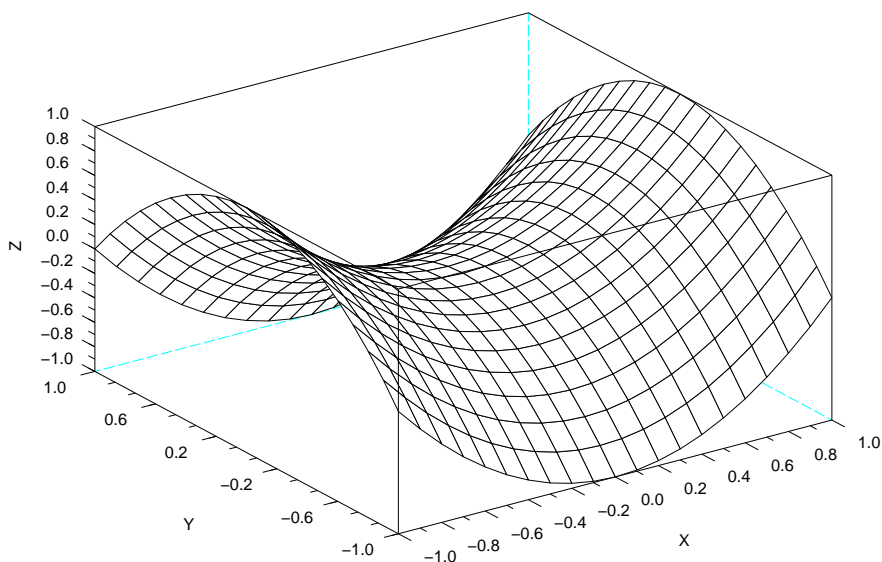
### Графики функций двух переменных

Для построения графиков функций двух переменных в Scilab можно использовать функцию `mesh(X,Y,Z)`, где  $X$  и  $Y$  определяют сетку, а  $Z$  — значения функции в узлах сетки. Для формирования сетки удобно использовать функцию `meshgrid()`.

**Пример.** Построить график функции  $z = x^2 - y^2$  в области  $D = [-1, 1] \times [-1, 1]$ .

```
-->[X,Y]=meshgrid(-1:0.1:1,-1:0.1:1);  
-->Z=X.^2-Y.^2;  
-->mesh(X,Y,Z)  
-->
```

Результат:



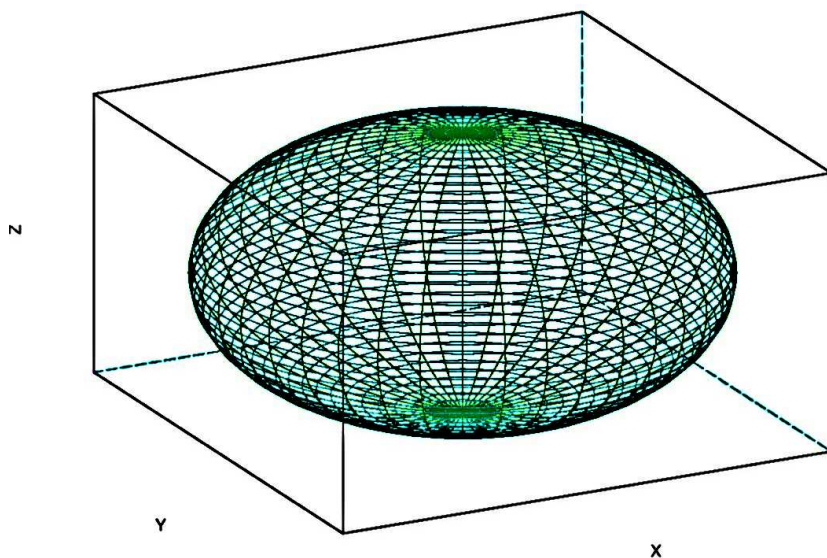
### Построение графика функции двух переменных, заданной параметрически

**Пример.** Построить график функции, заданной параметрически

$$X = \cos(u) \cdot \cos(v), Y = \cos(u) \cdot \sin(v), Z = \sin(u).$$

```
-->u = [-%pi/2:%pi/50:%pi/2];  
-->v = [0:%pi/20:2*%pi];  
-->X = cos(u)'*cos(v);  
-->Y = cos(u)'*sin(v);  
-->Z = sin(u)'*ones(v);  
-->plot3d3(X,Y,Z);  
-->
```

Результат:



### Построение нескольких графиков в одной графической области

Функция `subplot(m,n,p)` делит графическую область на  $m \times n$  подобластей, в каждой из которых можно построить свой график, и делает активной  $p$ -ю область.

#### Пример.

```
-->x=[-5:0.01:5]';  
-->subplot(2,2,1);  
-->plot2d(x,sin(x),style=3,...  
-->axesflag=5,rect=[-5,-5,5,5])  
-->subplot(2,2,2);  
-->plot2d(x,cos(x),style=3,...
```

```

-->axesflag=5,rect=[-5,-5,5,5])
-->subplot(2,2,3);
-->plot2d(x,abs(x),style=3,...
-->axesflag=5,rect=[-5,-5,5,5])
-->subplot(2,2,4);
-->plot2d(x,x.*sin(x),style=3,...
-->axesflag=5,rect=[-5,-5,5,5])
-->

```

Результат:

