

Графика в программе Scilab

Приме

р:

`plot(x,y)`

`x` — массив абсцисс; `y` — массив ординат;

`xtitle(title, xstr, ystr)`

`title` — название графика, `xstr` — название оси X, `ystr` — название оси Y;

`legend(leg1,leg2,...,legn[,pos][,boxed])`

`leg1` — имя первой легенды, `leg2` — имя второй легенды, ...

`legn` — имя n-й легенды; `pos` — месторасположение

легенды: 1 — верхнем правом углу (по умолчанию), 2 —

верхнем левом углу, 3 — нижнем левом углу, 4 — нижнем

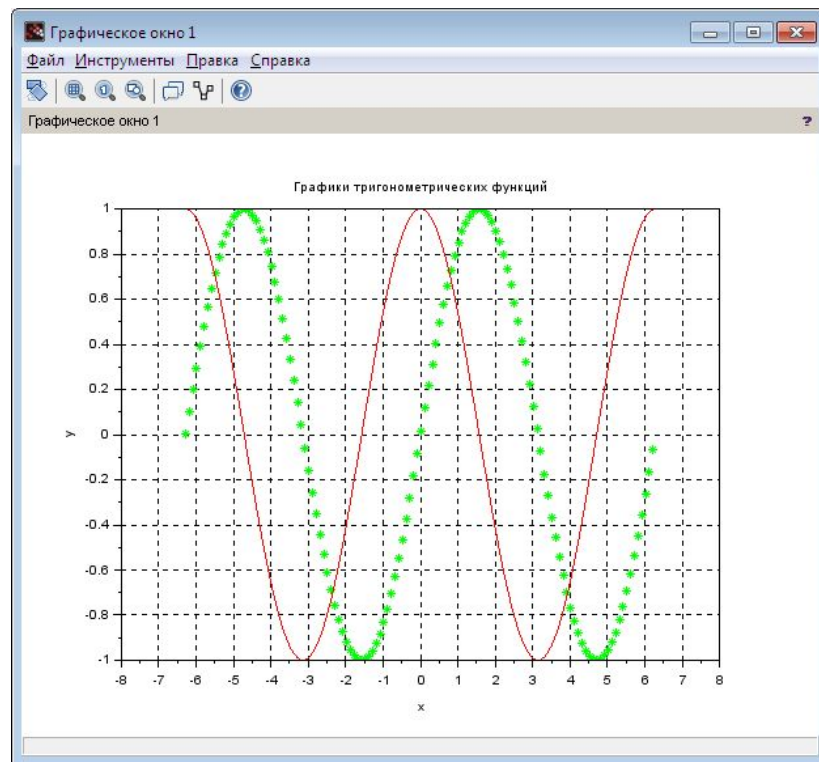
правом углу, 5 - определяется пользователем после

изображения графика; `boxed` — логическая переменная (по умолчанию %t), которая определяет рисовать или нет рамку вокруг легенды.

subplot(m, n, p) разбивает графическое окно на `m`
областей, где `m` — число областей по вертикали, `n` — число
областей по горизонтали, `p` — номер области, в которую
будет выводиться текущий график (области
отсчитываются последовательно по строкам)

`xgrid()` — показывает сетку

```
*example.sce
1 x=-2*pi:0.1:2*pi;
2 y1=sin(x);y2=cos(x);
3 figure(1)
4 plot(x,y1,'*g',x,y2,'-r')
5 xgrid()
6 xtitle('Графики тригонометрических функций','x','y')
```



Трехмерные графики

`[X,Y]=meshgrid(x,y)` – преобразует векторы x и y в матрицы X и Y , которые могут быть использованы для вычисления функции двух переменных и построения трехмерных графиков, например $Z = f(X, Y)$

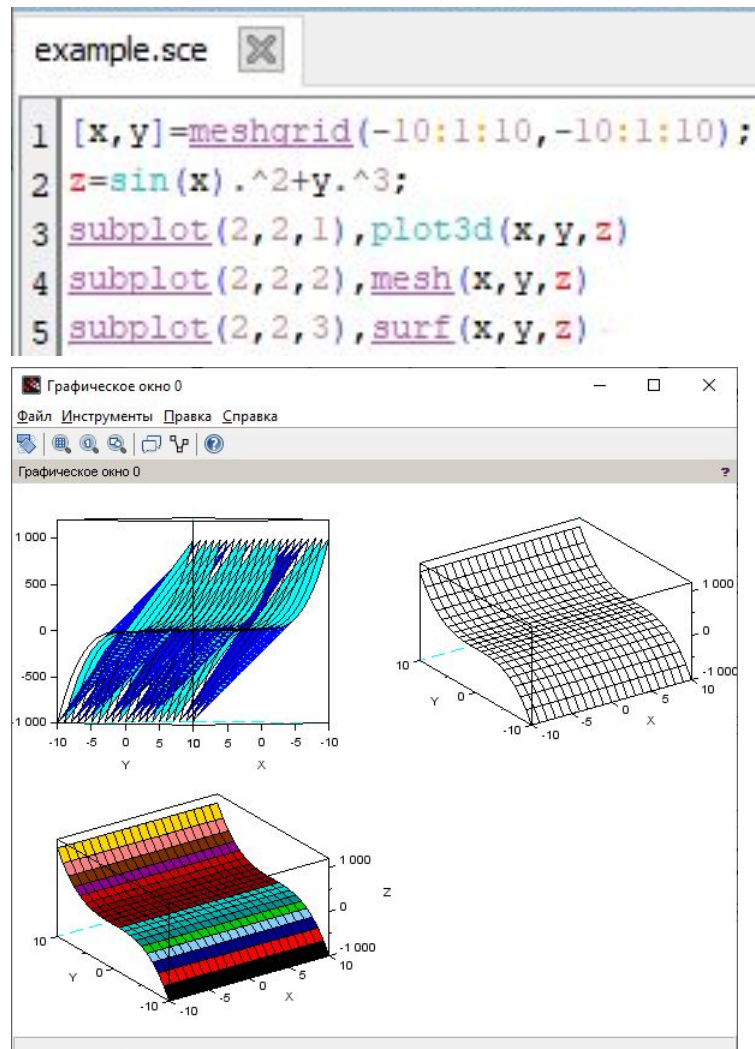
`plot3d(X,Y,Z)` – строит поверхность, соединяя узловые точки отрезками прямых. X, Y, Z – матрицы одинакового размера.

`plot3d(X,Y,Z,S)` – аналогична предыдущей команде, но со спецификацией стиля линий и точек. Здесь S – строковая константа, куда задается тип линии.

`mesh(X,Y,Z)` – выводит в графическое окно сетчатую поверхность $Z(X, Y)$, цвет зависит от высоты поверхности.

`surf(X,Y,Z)` – строит цветной сетчатый график поверхности по данным матриц X, Y и Z с функциональной закраской ячеек цветом.

Приме



Нахождение нулей, минимальных и максимальных значений функции по графику

Если функция задана полиномом:

$p = \text{poly}(a, x, f)$

a – число, x – символьная переменная,
 f – определяет способ задания полинома;

$f = 'c'$ – для формирования полинома с
коэффициентами,

$f = 'r'$ – значения параметра a

воспринимаются как корни.

$\text{roots}(p)$ – находит нули функции,
заданной полиномом p .

В остальных случаях:

$\text{fsolve}(x0, \text{func})$ – находит решение
трансцендентного уравнения func при
начальном значении корня $x0$ (может
быть вектором, если более одного корня)

Примеры:

--> $p1 = \text{poly}([1 \ 2 \ 3], 'x', 'c')$

$p1 =$
 2

$1 + 2x + 3x^2$

--> $p2 = \text{poly}([1 \ 2 \ 3], 'x', 'r')$

$p2 =$
 $2 \ 3$

$-6 + 11x - 6x^2 + x^3$

--> $\text{roots}(p2)$

$\text{ans} =$

3.

2.

1.

Решение нелинейного
уравнения:

```
1.sce 
1 x=-10:0.1:10
2 y=x.^2+5*x-10
3 plot(x,y)
4 xgrid()
1 function y=func(x)
2     y=x^2+5*x-10
3 endfunction
8 x0=[-8 2]
9 Rez=fsolve(x0, func)
10 disp('Решение')
11 disp(Rez)
12 //Проверка
13 Prov=func(Rez)
14 disp('Проверка')
15 disp(Prov)
```

optim(func, xopt) – находит экстремум функции (минимум), заданной в func в окрестности точки xopt.

Если нужно найти максимум, то у функции меняют знак, а у результата вычислений меняют знак на противоположный

Пример:

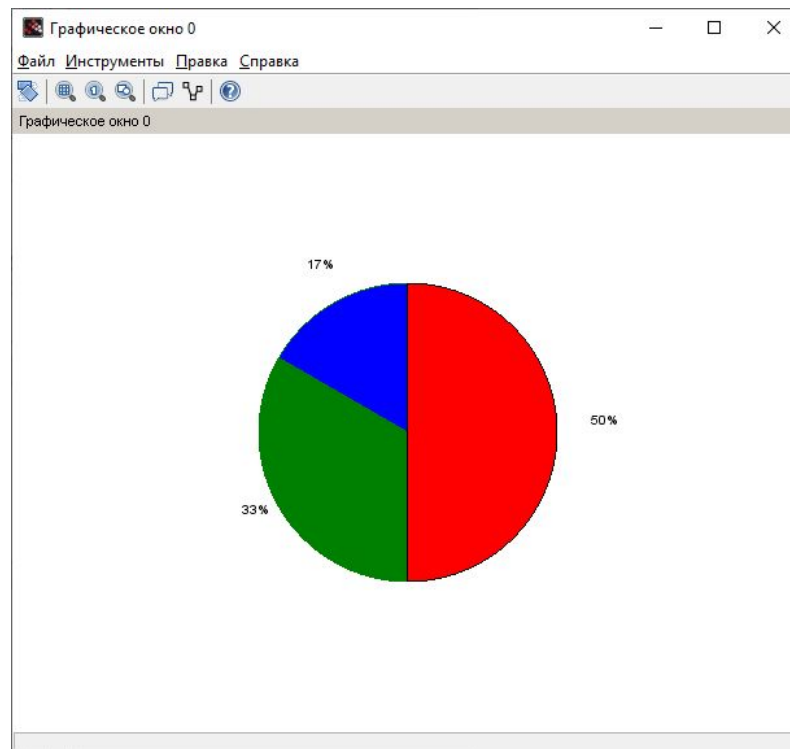
Экстремум функции (минимум):

```
1 function [y1, g1, ind]=func1(x, ind)
2 ... y1=x^2+5*x-10
3 ... g1=2*x+5
4 endfunction
21 xopt=-18;
22 [fmin, xmin]=optim(func1, xopt)
23 disp('Минимум функции')
24 disp(fmin)
```

Построение диаграмм

`pie([1 2 3])` – строит диаграмму в процентном соотношении относительно максимального элемента

Примеры:

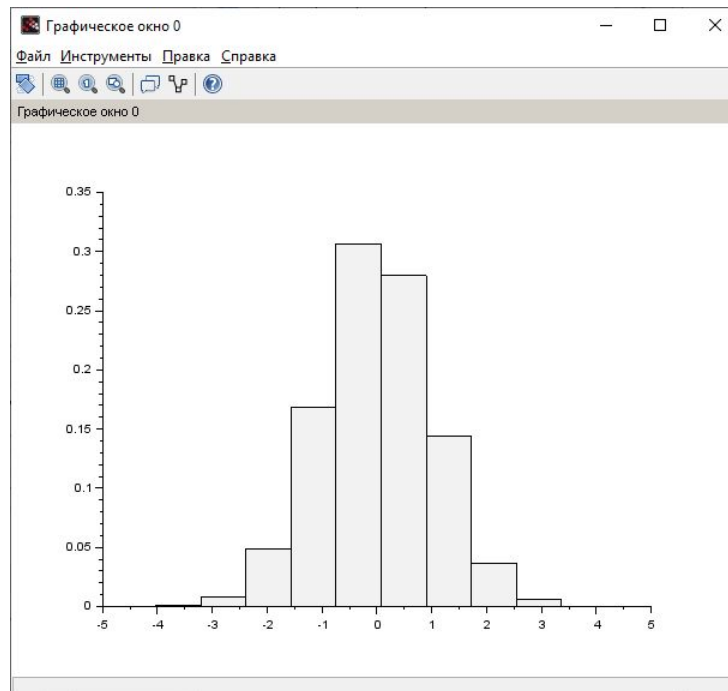


Построение гистограмм

`d=rand(1,10000,'normal');` - генерация случайного вектора с нормальным законом распределения

`histplot(10,d)` – строит гистограмму случайного вектора с нормальным законом распределения

Примеры:



Построение линий тренда

Формат линии графика:

Тип линии		Тип точки		Цвет линии	
-	Сплошная	.	Точка	y	Желтый
:	Двойной пунктир	o	Окружность	m	Фиолетовый
-.	Штрих-пунктир	x	Крест	c	Голубой
--	Штриховая	+	Плюс	r	Красный
		*	Звездочка	g	Зеленый
		s	Квадрат	b	Синий
		d	Ромб	w	Белый
		v	Треугольник вниз	k	Черный