

## Практическое занятие №1

### Построение трехмерных графиков в SciLab

**Варианты заданий:** ваш порядковый номер в журнале.

Результат представить в отчете. Обязательно наличие программного кода и графиков.

Для построения трехмерных графиков используют операторы **plot3d**, **plot3d1** и **mesh**.

Обращение к ним следующее:

```
plot3d(x,y,z,[theta,alpha,leg,flag,ebox][keyn=valuen]),  
plot3d1(x,y,z,[theta,alpha,leg,flag,ebox][keyn=valuen])
```

Здесь  $x$  - вектор - столбец значений абсцисс;

$y$  - вектор - столбец значений ординат;

$z$  - матрица значений функции;

$\theta$ ,  $\alpha$  - действительные числа, которые определяют в градусах сферические координаты угла зрения на график. Попросту говоря, это угол, под которым наблюдатель видит отображаемую поверхность;

$leg$  - подписи координатных осей графика - символы, отделяемые знаком @. Например, 'X@Y@Z'.

$flag$  - массив, состоящий из 3 целочисленных параметров  $[mode, type, box]$ . Здесь  $mode$  устанавливает цвет поверхности.

Значения параметра  $mode$

>0 - поверхность имеет цвет «mode», выводится прямоугольная сетка.

0 - выводится прямоугольная сетка, заливка отсутствует (белый цвет).

<0 - поверхность имеет цвет «mode», отсутствует прямоугольная сетка.

По умолчанию, равен 2 - цвет заливки синий, прямоугольная сетка выводится.

$type$  - позволяет управлять масштабом графика, по умолчанию имеет значение 2.

Значения параметра  $type$

0 - применяется способ масштабирования, как у ранее созданной графики.

1 - границы графика указываются вручную с помощью параметра  $ebox$ .

2 - границы графика определяют исходные данные.

$Box$  - определяет наличие рамки вокруг отображаемого графика. По умолчанию равен 4.

Значения параметра  $box$

0 и 1 - нет рамки

2 - только оси, находящиеся за поверхностью

3 - выводится рамка и подписи осей

4 - выводится рамка, оси и их подписи.

$ebox$  - определяет границы области, в которую будет выводиться поверхность, как вектор  $[xmin, xmax, ymin, ymax, zmin, zmax]$ . Этот параметр может использоваться только при значении параметра  $type=1$ .

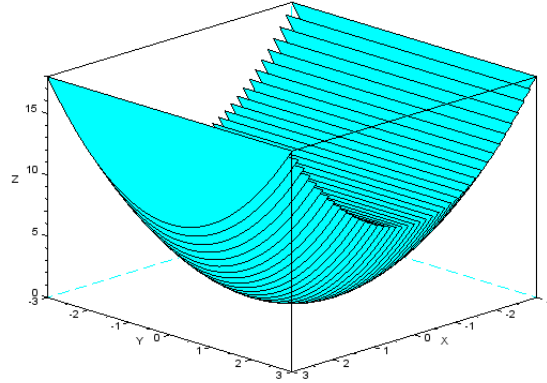
$keyn=valuen$  - последовательность значений свойств графика  $key1=value1, key2=value2, \dots, keyn=valuen$ , таких как толщина линии, ее цвет, цвет заливки фона графического окна, наличие маркера и др. Таким образом, функции  $plot3d$  ( $plot3d1$ ) в качестве параметров необходимо передать прямоугольную сетку и матрицу значений в узлах сетки.

### Пример1:

-->`[x, y]=meshgrid([-3:0.15:3])` % Задаётся опорная плоскость для построения трехмерной поверхности, переменные  $x, y$  меняются от  $-3$  до  $3$  с шагом  $0,15$ .

-->`z=x.^2+y.^2` %вычисляем  $z$

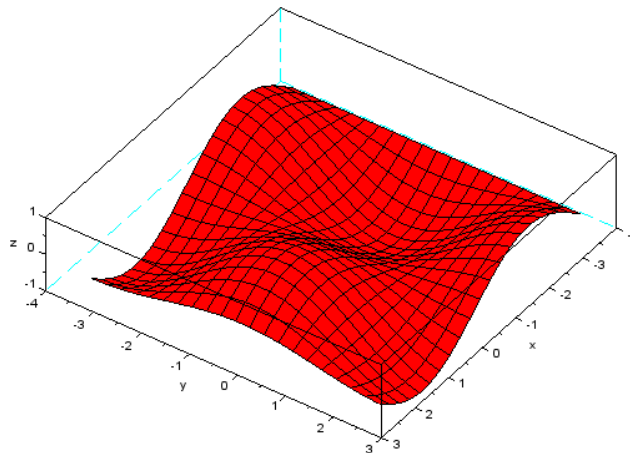
-->`plot3d(x, y, z)` %строим график



### Пример2:

`t=-%pi:3:%pi;`

`plot3d(t,t,sin(t)*cos(t),35,45,'x@y@z',[5,2,4]);`



Для формирования прямоугольной сетки впервые в Scilab 4.0 появилась функция *meshgrid*. Обращение к ней имеет вид:

`[X,Y[,Z]]=meshgrid(x,y[,z])`

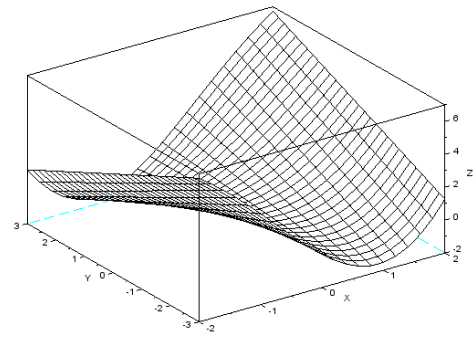
здесь  $x, y [,z]$  - массивы 2(3) исходных параметров  $X, Y (,Z)$ , указываемые через запятую;  $X, Y [,Z]$  - матрицы в случае 2 и массивы в случае 3 входных величин.

После формирования сетки вывести в нее графику можно с помощью функции *surf* либо *mesh*. Так же, как и в случае с функциями *plot3d* и *plot3d1*, *surf* строит поверхность, заливая каждую ячейку цветом, который зависит от конкретного значения функции в узле сетки, а *mesh* заливает ее одним цветом.

Таким образом, *mesh* является полным аналогом функции *surf* со значением параметров *Color mode*=индекс белого цвета в текущей палитре цветов и *Color flag*=0.

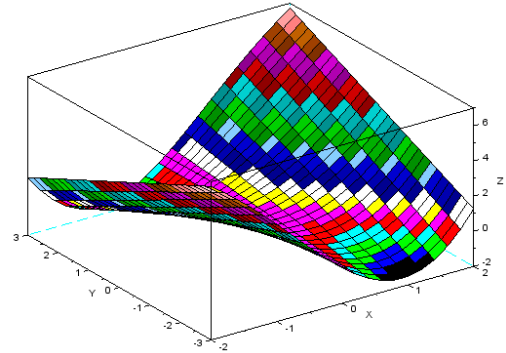
### Пример 3:

```
-->[x, y]=meshgrid(-2:0.2:2, -3:2:3);  
-->z=x.^2+y.*sin(x);  
-->mesh(x,y,z)
```



### Пример 4:

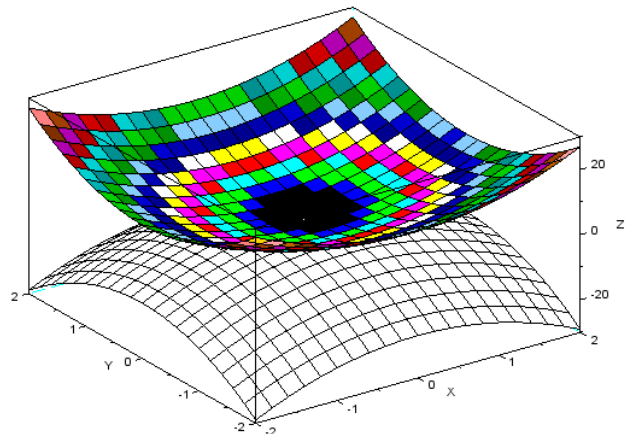
```
[x, y]=meshgrid(-2:0.2:2, -3:2:3);  
z=x.^2+y.*sin(x);  
surf(x,y,z)
```



### Пример 5:

Программа, приведенная ниже, показывает, как можно в одном окне построить графики двух трехмерных функций (оператор `mtlb_hold('on')`). Правда, и без этого оператора график остается таким же.

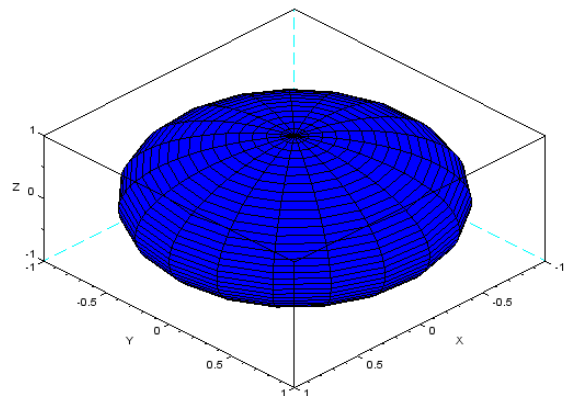
```
[x y]=meshgrid(-2:0.2:2,-2:0.2:2);  
z=3*x.^2+4*y.^2-1;  
z1=-3*x.^2-4*y.^2-1;  
surf(x,y,z);  
mtlb_hold('on');  
mesh(x,y,z1);
```



### Пример 6:

Сферу можно построить так:

```
u = linspace(-%pi/2,%pi/2,40);  
v = linspace(0,2*%pi,20);  
X = cos(u)*cos(v);  
Y = cos(u)*sin(v);  
Z = sin(u)*ones(v);  
plot3d2(X,Y,Z);
```

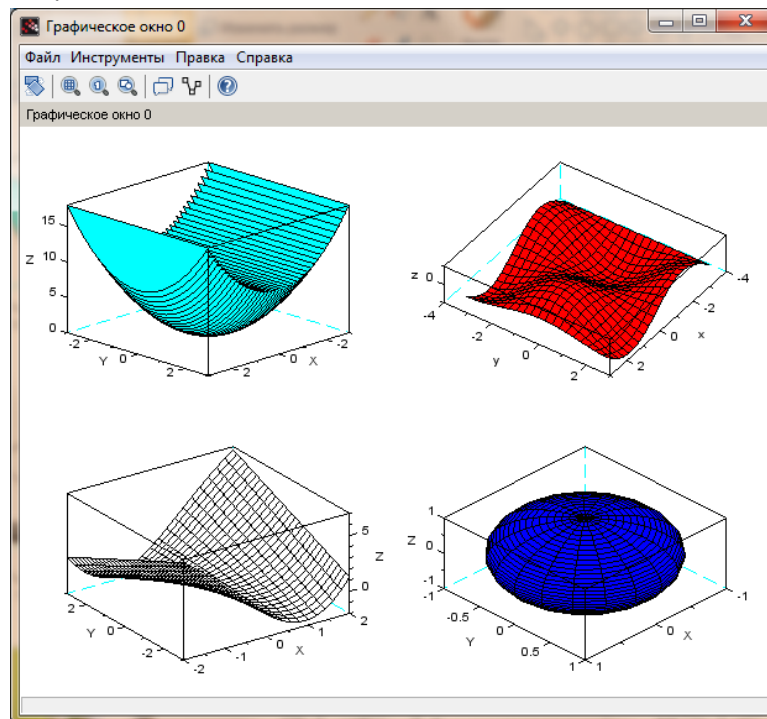


Оператор `v = linspace(0,2*%pi,20);` задает вектор `v`, состоящий из 20 значений с шагом  $\pi/10$ .

### Задание 1:

Построить четыре любых трехмерных графика в разных подокнах одного графического окна.

Пример выполнения:



### Функция contour

В Scilab, кроме построения объемных графиков, также реализована возможность создания пространственных моделей объектов. На практике часто возникает необходимость построения карт в изолиниях значений показателя, где  $X$ ,  $Y$  -координаты задают положение конкретной изучаемой точки на плоскости, а  $Z$ -координата – зафиксированную величину показателя в этой точке. Точки с одинаковыми значениями показателя соединяют так называемые изолинии – линии одинаковых уровней значений исследуемой величины.

Для построения изолиний в Scilab существует функция **contour**. Обращение к ней имеет вид:

`contour(x,y,z,nz[theta,alpha,leg,flag,ebox,zlev])`

Здесь

- x, y** – массивы действительных чисел;
- z** – матрица действительных чисел – значения функции, описывающей поверхность  $Z(x, y)$ ;
- nz** – параметр, который устанавливает количество изолиний. Если **nz** – целое число, то в диапазоне между минимальным и максимальным значениями функции  $Z(x, y)$  через равные интервалы будут проведены **nz** изолиний. Если же задать **nz** как массив, то изолинии будут проводиться через все указанные в массиве значения;
- theta, alpha** – действительные числа, которые определяют в градусах сферические координаты угла обозрения наблюдателя. Попросту говоря, это угол, под которым наблюдатель видит отображаемую поверхность;
- leg** – подписи координатных осей графика □ символы, отделяемые знаком @. Например, 'X@Y@Z'.
- flag** – массив, состоящий из трех целочисленных параметров: [mode,type,box]. Здесь **mode** – устанавливает способ и место нанесения линий уровня.

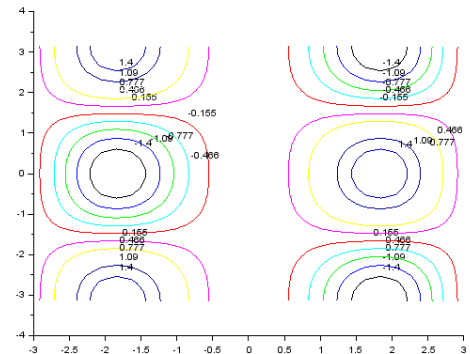
- type** – позволяет управлять масштабом графика (см. табл. 5.2), по умолчанию имеет значение 2;
- box** – определяет наличие рамки вокруг отображаемого графика. По умолчанию равен 4;
- ebox** – определяет границы области, в которую будет выводиться поверхность, как вектор  $[x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}]$ . Этот параметр может использоваться только при значении параметра `type=1`;
- zlev** – математическое выражение, которое задает план (горизонтальную проекцию заданной поверхности) для построения изолиний. По умолчанию совпадает с уравнением, описывающим плоскость,  $\square$  в этом случае может не указываться.

### Пример 7:

```

Построение изолиний поверхности  $Z=x \cdot \sin(x)^2 \cdot \cos(y)$ 
t=linspace(-%pi,%pi,30);
function z=my_surface(x,y)
z=x*sin(x)^2*cos(y)
endfunction
contour(t,t,my_surface,10);

```



### Функция **contour**

В Scilab существует функция `contourf`, которая не просто изображает поверхность на горизонтальной плоскости в виде изолиний, но и заливает интервалы между ними цветом, в зависимости от конкретного уровня значений показателя.

Обращение к функции имеет вид:

```
contourf (x,y,z,nz,[style,strf,leg,rect,nax])
```

Здесь

**x, y** – массивы действительных чисел;

**z** – матрица действительных чисел  $\square$  значения функции, описывающей поверхность  $Z(x, y)$ ;

**nz** – параметр, который устанавливает количество изолиний. Если `nz` – целое число, то в диапазоне между минимальным и максимальным значениями функции  $Z(x, y)$  через равные интервалы будут проведены `nz` изолиний. Если же задать `nz` как массив, то изолинии будут проводиться через все указанные в этом массиве значения;

**style** – массив того же размера, что и `nz`  $\square$  устанавливает цвет для каждого интервала уровней значений;

**strf** – строка, состоящая из трех чисел – “`csa`”. Здесь `c` (Captions) устанавливает режим отображения подписей графика; `s` (Scaling) – режим масштабирования; `a` (Axes) – определяет положение осей графика.

**leg** – легенда графика, подпись каждой из кривых  $\square$  символы, отделяемые знаком `@`. По умолчанию – “ ”.

**rect** – вектор  $[x_{min}, y_{min}, x_{max}, y_{max}]$ , который определяет границы изменения `x` и `y` координат графической области окна;

**Пример 8:** Построение поверхности  $Z=\sin(x) \cdot \cos(x)$  и ее изображения на горизонтальной поверхности в одном графическом окне

```

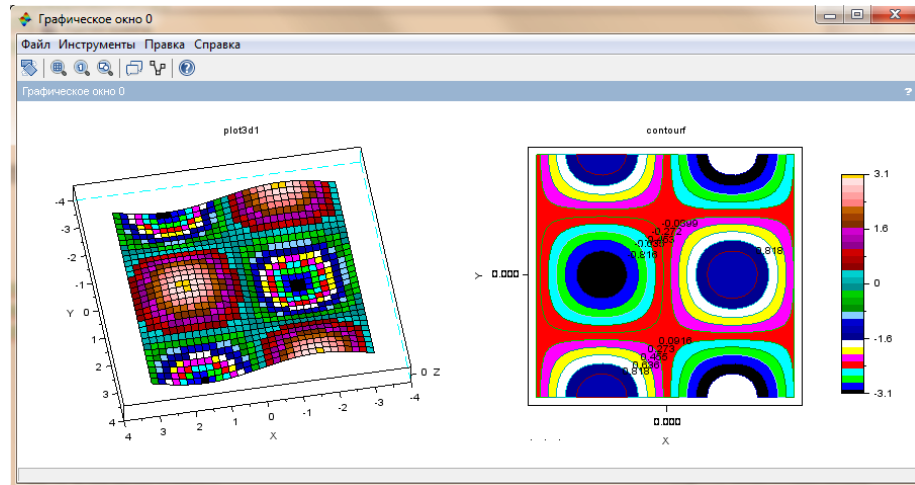
t=0:0.2:pi;
def('z=Surf(x,y)', 'z=sin(x)*cos(y)');
subplot(121);

```

```

z=feval(t,t,Surf);
plot3d1(t,t,z,80,15);
xlabel('plot3d1');
subplot(122);
contourf(t,t,z,10,10:20,strf="121");
colorbar(-%pi,%pi);
xlabel('contourf','X','Y');

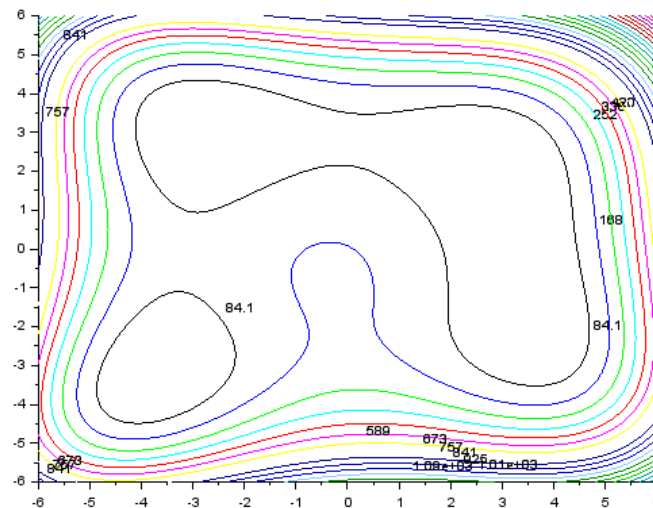
```



## Задание 2:

Построить изолинии для функции (по вариантам) в заданном диапазоне  $-6 < x < 6$  и  $-6 < y < 6$  с шагом 0.1.

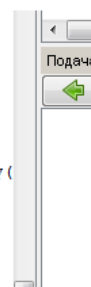
Пример выполнения для функции  $f(x,y)=(x^2+y-11) \times (x^2+y-11) + (x+y^2-7) \times (x+y^2-7)$  в области определения  $-6 < x < 6$  и  $-6 < y < 6$  с шагом 0.1 с использованием функций **contour** и **contourf**:



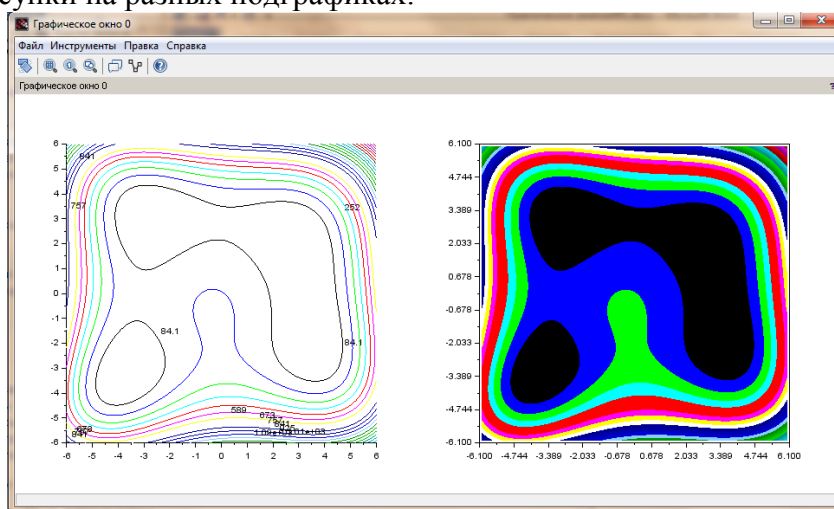
```

--> x=[-6:0.1:6];
--> y=[-6:0.1:6];
--> for i=1:length(x)
> for j=1:length(y)
> z(i,j)=(x(i)*x(i)+y(j)-11)*(x(i)*x(i)+y(j)-11)+(x(i)+y(j)*y(j)-7)*(x(i)+y(j)*y(j)-7);
> end
> end
--> contour(x,y,z,25);

```



Разместите рисунки на разных подграфиках.



### Варианты:

Диапазон

$-6 < x < 6$  и  $-6 < y < 6$ ,  $a, b - \text{const}$

1.  $f(x, y) = x^2 + y - 3xy + 3$
2.  $f(x, y) = x^2 + 2xy^3 - 5xy + y^2$
3.  $f(x, y) = (1 - x^2) + (2 - y^2)$
4.  $f(x, y) = 8x^2 + 4xy + 5y^2$
5.  $f(x, y) = 2x^3 + 4xy^3 - 10xy + y^2$
6.  $f(x, y) = 4x^2 + 3y^2 - 4xy + x$
7.  $f(x, y) = 100(y - x^2) + (1 - x)^2$
8.  $f(x, y) = 2y^3 + 4x^2y - 10x + y^2$
9.  $f(x, y) = (x - a^2y^2)(x - b^2y^2)$
10.  $f(x, y) = (x^2 + (y + 1)^2)(x^2 + (y - 1)^2)$
11.  $f(x, y) = x^3 + 5y - xy + 7$
12.  $f(x, y) = 3x^2 + 3y^2 - 3x$
13.  $f(x, y) = x^2 + y^2 \times (x + y) - 3xy$
14.  $f(x, y) = 5(x^2 + y^2) \times x - 3xy + 7$
15.  $f(x, y) = 4x^2 + 3y^2 - (x + y)$
16.  $f(x, y) = a \times x^2 + b \times y^2 - a^2 \times (x + y) - b$