

### 1. Элементарная трехмерная (3-D) графика — функция **plot3**.

Функция **plot3** в определенном смысле является аналогом функции **plot**. С помощью **plot3** формируется построение линии в трехмерном пространстве по заданным трем векторам .

Пример 1.1. Построение трехмерной пространственной спирали.

```
t=0:0.05:9*pi; x=2*sin(t); y=cos(t); % t, x, y — вектора
одинакового размера
plot3(x,y,t, 'r*'), grid,
xlabel('ось X'), ylabel('ось Y'), zlabel('ось Z-t')
title('Пространственная спираль')
% Сохранить программу в текстовом редакторе MATLAB, например, под
именем sp3
% Поменять местами x, y, t в функции plot3
% Сменить начертание и цвет графика
% Добавить пояснение к начертанию в виде легенды
— legend('звездочки')
% Установить следующий диапазон для t: t=-9*pi:0.05:9*pi;
% Пояснения к графику с помощью функций gtext для 3D-графики не
применяются
```

Пример 1.2. Построение сферы по окружностям.

```
%n=100;
n=input('n='); % Клавиатурный ввод числа n по запросу в командном
окне
t1=pi*(-n:5:n)/n;
t2=(pi/2)*(-n:5:n)'/n; % транспонированный вектор
X=cos(t2)*cos(t1);
Y=cos(t2)*sin(t1);
E=ones(size(t1)); % матрица единиц размерности вектора t1
Z=sin(t2)*E;
plot3(X,Y,Z, 'r'), grid, title('Сфера')
% Сохранить программу в текстовом редакторе MATLAB под именем sfera3.
% Программу выполнить при различных значениях n.
% При n =100 изменить шаг в массивах t1 и t2: для t1 и t2 одновременно: 3, 1,
10, 25, 50
% Для t1=50, для t2=1; для t1=1, для t2=50.
% Программу sfera3 выполнить без клавиатурного ввода, а для заданного
числа n.
% Установить в программе различные цвета изображения сферы.
```

% Использовать различные коэффициенты для t1 и t2: pi/5, pi/10, pi/50, 2\*pi, 5\*pi одновременно  
% и в сочетании с коэффициентом, равным pi (3.14), при одном из векторов (t1 или t2).

2. Формирование прямоугольной сетки на плоскости — **meshgrid**.

```
» [x,y]=meshgrid(-5:0.5:5,-5:0.5:5);
```

```
» plot(x,y),xlabel('X'),ylabel('Y')
```

% Результатом действия функции **meshgrid** является формирование "основания" в плоскости **ХОУ** для построения над этим основанием пространственной фигуры.

3. Построение пространственных сетчатых фигур — **mesh**.

Пример 3.1.

```
[x,y]=meshgrid(-5:0.1:5,-5:0.1:5);
```

```
Z = 1*x.*exp(-x.^2 - y.^2); % Коэффициент 1 заменить: 2, 5, 10, 20  
mesh(Z),xlabel('X'),ylabel('Y'),zlabel('Z'),title('Z-  
поверхность')
```

% Для сравнения применить **plot3(x,y,Z),grid** вместо **mesh(Z)**.

Пример 3.2.

```
[x,y]=meshgrid(-15:0.2:15,-15:0.2:15);
```

```
R=sqrt(x.^2+y.^2)+0.001; %Коэфф. 0.001 введен для исключения  
деления на ноль
```

```
Z=1*sin(R)./R;
```

```
mesh(Z)
```

% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1

% Для сравнения применить **plot3(x,y,Z),grid**

4. Сетчатая поверхность с проекциями линий постоянного уровня — **meshc**.

Пример 4.1.

```
[x,y]=meshgrid(-15:0.2:15,-15:0.2:15);
```

```
R=sqrt(x.^2+y.^2)+0.001;
```

```
Z=1*sin(R)./R;
```

```
meshc(Z)
```

% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1

Пример 4.2.

```
[x,y]=meshgrid(-5:0.1:5,-5:0.1:5);
```

```
Z = 1*x.*exp(-x.^2 - y.^2);
```

```
meshc(Z), xlabel('X'), ylabel('Y'), zlabel('Z'), title('Z-  
поверхность')
```

```
% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1
```

5. Сетчатая поверхность с пьедесталом плоскости отсчета на нулевом уровне— **meshz**.

Пример 5.1.

```
[x,y]=meshgrid(-15:0.2:15,-15:0.2:15);
```

```
R=sqrt(x.^2+y.^2)+0.001;
```

```
Z=1*sin(R)./R;
```

```
meshz(Z)
```

```
% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1
```

Пример 5.2.

```
[x,y]=meshgrid(-5:0.1:5,-5:0.1:5);
```

```
Z = 1*x.* exp(-x.^2 - y.^2);
```

```
meshz(Z), xlabel('X'), ylabel('Y'), zlabel('Z'), title('Z-  
поверхность')
```

```
% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1
```

6. Построение пространственных сплошных фигур — **surf**.

Пример 6.1.

```
[x,y]=meshgrid(-15:0.2:15,-15:0.2:15);
```

```
R=sqrt(x.^2+y.^2)+0.001;
```

```
Z=1*sin(R)./R;
```

```
surf(Z)
```

```
% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1
```

Пример 6.2.

```
[x,y]=meshgrid(-5:0.1:5,-5:0.1:5);
```

```
Z = 1*x.* exp(-x.^2 - y.^2);
```

```
surf(Z), xlabel('X'), ylabel('Y'), zlabel('Z'), title('Z-  
поверхность')
```

```
% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1
```

1. Сплошная поверхность с проекциями линий постоянного уровня — **surfc**.

Пример 7.1.

```
[x,y]=meshgrid(-5:0.1:5,-5:0.1:5);
```

```
Z = 1*x.* exp(-x.^2 - y.^2);
```

```
surfc(Z), xlabel('X'), ylabel('Y'), zlabel('Z'), title('Z-  
поверхность')
```

```
% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1
```

### Пример 7.2.

```
[x,y]=meshgrid(-15:0.2:15,-15:0.2:15);  
R=sqrt(x.^2+y.^2)+0.001;  
Z=1*sin(R)./R;  
surf(Z)  
% Коэффициент 1 заменить: 5, 10, 20, 0.5, 0.1
```

## 8. Стандартные сферические поверхности — **sphere**.

### 8.1. Сфера единичного радиуса — **sphere**;

```
» sphere(12)  
» sphere(102)  
» sphere % равносильно sphere(20);
```

### 8.2. Формирование сферы с произвольным радиусом, например, равным 5;

```
» [x,y,z]=sphere(40);  
» x5=5*x;  
» y5=5*y;  
» z5=5*z;  
» plot3(x5,y5,z5),grid  
% или при помощи функции mesh:  
» mesh(x5,y5,z5)  
» meshC(x5,y5,z5)  
» meshZ(x5,y5,z5)  
% или при помощи функции surf:  
» surf(x5,y5,z5)  
» surfC(x5,y5,z5)  
% Выполнить построения с надписями
```

## 9. Пространственное распределение вероятностей Гаусса — **peaks**.

### Пример 9.1.

```
peaks; % По умолчанию формируются матрицы размера 49×49, по которым  
строится  
%поверхность с пометками координат и титульной надписью 'Peaks'.  
peaks, title('Gauss'); % Со специальной титульной надписью  
peaks(78); % Размерность матриц построения 78×78  
peaks(25); % Размерность матриц построения 25×25
```

### Пример 9.2.

```
z=peaks; surf(z); % Построение поверхности Гаусса по заданной 49×49  
матрице z.
```

```

% Сравнить с поверхностью на основе функции mesh(z)
z=peaks; surf(z), title('Gauss'), xlabel('X'), ylabel('Y'), zlabel('Z')
z=peaks(25); surf(z), title('Gauss_25')
% Сравнить с функцией mesh(z)
z=3*peaks(78); surf(z), title('Gauss') % С множителем по оси Z.
% Применить множители: 4, 10, 33, 50

```

Пример 9.3. Цветовые массивные уровни на плоскости от распределения Гаусса.

```

[X,Y,Z]=peaks; pcolor(X,Y,Z)
[X,Y,Z]=peaks; pcolor(X,Y,Z), title('Цвет')
[X,Y,Z]=peaks; pcolor(7*X,12*Y,20*Z), title('Цвет') % С
расширенной областью определения на плоскости XOY — множители 7 и 12 и
с изменением по Z — коэффициент 20

```

Пример 9.4. Распределение Гаусса с задаваемой областью определения на плоскости XOY.

```

[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5); peaks(X,Y)
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5); Z=peaks(X,Y); surf(Z)
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);
Z=peaks(X,Y); pcolor(X,Y,Z)
% Функция pcolor позволяет наблюдать область определения в плоскости
XOY
% Изменить шаг по x и y : 0.2, 0.1, 0.5
% Изменить границы по x и y одновременно и порознь:(-7:0.2:3),(-7:0.2:1),(-
7:0.2:-1),(-7:0.2:-3)

```

Пример 9.5. Линии уровня трехмерной поверхности — **contour**.

```

% Линии уровня без учета масштаба плоскости XOY:
contour(peaks), grid, title('Линии уровня') % 10 линий уровня по
умолчанию
contour(peaks, 4), grid, title('Линии уровня') % 4 линии уровня
contour(peaks, 10), grid, title('Линии уровня') % 10 линий уровня
contour(peaks, 15), grid, title('Линии уровня') % 15 линий уровня
contour(peaks, 25), grid, title('Линии уровня') % 25 линий уровня

% Применить функцию contour для стандартной сферической
поверхности sphere с различным числом линий уровня.

```

Пример 9.6. Линии уровня с учетом масштаба плоскости XOY (два примера):

```
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour(X,Y,Z),grid  
[X,Y]=meshgrid(-10:0.3:10,-5:0.3:5);  
Z=peaks(X,Y);contour(X,Y,Z),grid
```

Пример 9.7. Линии уровня с учетом масштаба плоскости XOY и заданным числом (три примера):

```
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour(X,Y,Z,4),grid  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour(X,Y,Z,14),grid  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour(X,Y,Z,24),grid
```

Пример 9.7. Линии уровня с цветовой окраской плоскости XOY — **contourF**.  
**contourF(peaks),title('Цвет линий уровня')**

```
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contourF(Z)  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contourF(X,Y,Z)  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contourF(X,Y,Z,15)
```

Пример 9.8. "Пространственные" линии уровня — **contour3**.

```
contour3(peaks)  
contour3(peaks,25),title('25 линий уровня в пространстве')  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour3(Z)  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour3(Z,25)  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour3(X,Y,Z)  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour3(X,Y,Z,25)  
[X,Y]=meshgrid(-7:0.3:7,-5:0.3:5);  
Z=peaks(X,Y);contour3(X,Y,12*Z)
```

```
[X,Y]=meshgrid(-7:0.3:7,-
5:0.3:5);Z=peaks(X,Y);contour3(2*X,8*Y,12*Z)
```

Пример 9.9. Обзор поверхности из заданной точки пространства — **view**.

```
peaks,view(10,45); % Число 10 — азимут обзора, 45 — угол обзора (все в
градусах)
```

```
[X,Y]=meshgrid(-7:0.3:7,-
5:0.3:5);Z=peaks(X,Y);surf(Z),view(10,45)
[X,Y]=meshgrid(-7:0.3:7,-
5:0.3:5);Z=peaks(X,Y);surf(Z),view(-10,45)
```

```
[X,Y]=meshgrid(-5:0.2:5,-5:0.2:5);Z=X.*exp(-X.^2-Y.^2);
surf(X,Y,Z) %
surf(X,Y,Z),view(-45,60)
```

```
[X,Y]=meshgrid(-5:0.2:5,-5:0.2:5);
Z=X.*exp(-X.^2-Y.^2);
surf(X,Y,Z),view(-10,60)
```

% Задать различные параметры функции **view** и выполнить построение различных поверхностей.

10. Построение поверхностей относительно полярной системы координат.

Преобразование **pol2cart**.

```
[X,Y]=meshgrid(-5:0.2:5,-5:0.2:5);[x1,y1]=pol2cart(X,Y);
peaks(x1,y1);
```

```
[X,Y]=meshgrid(-5:0.2:5,-5:0.2:5);[x1,y1]=pol2cart(X,Y);
Z=peaks(x1,y1);surf(x1,y1,Z),title('Полярная плоскость')
```

```
[X,Y]=meshgrid(-5:0.2:5,-5:0.2:5);[x1,y1]=pol2cart(X,Y);
Z=x1.*exp(-x1.^2-y1.^2);surf(x1,y1,Z),
```

% С изменением точки обзора

```
[X,Y]=meshgrid(-5:0.2:5,-5:0.2:5);[x1,y1]=pol2cart(X,Y);
Z=peaks(x1,y1);surf(x1,y1,Z),title('Полярная
плоскость'),view(-12,45)
```

```
[X,Y]=meshgrid(-5:0.2:5,-5:0.2:5);[x1,y1]=pol2cart(X,Y);
Z=x1.*exp(-x1.^2-y1.^2);surf(x1,y1,Z),view(-12,45)
```

## 11. Построение цилиндрических поверхностей — `cylinder`.

`cylinder` % Построение цилиндра без параметров, по умолчанию

`cylinder(40)` % Построение цилиндра с заданным размером  
плоскости XOY

`cylinder(40,60)` % Цилиндр с заданным размером плоскости XOY и  
числом образующих %граней — 60

`cylinder(40,6)`

% Построение совокупности цилиндробразующих поверхностей

`[x,y,z]=cylinder([5 0],160); surf(x,y,z)` % Конус по заданному  
вектору [5,0]

`[x,y,z]=cylinder([5 0 5],160); surf(x,y,z)` % 2 конуса

`[x,y,z]=cylinder([5 0],3); surf(x,y,z)` % Пирамида

`[x,y,z]=cylinder([5 0 5],3); surf(x,y,z)` % 2 пирамиды

`[x,y,z]=cylinder([5 12 12],100); surf(x,y,z)`

`[x,y,z]=cylinder([5 12 5],100); surf(x,y,z)`

`[x,y,z]=cylinder([5 12 12 5],100); surf(x,y,z)`

`[x,y,z]=cylinder([5 12 5 12 5 12],100); surf(x,y,z)`

`[x,y,z]=cylinder([5 12 12 5 18],100); surf(x,y,z)`

`[x,y,z]=cylinder([5 12 12 5 5 18 18],100); surf(x,y,z)`

% Построение совокупности цилиндробразующих поверхностей с  
масштабированием

`[x,y,z]=cylinder([5 12 12 5 5 18 18],100); surf(x,y,20*z)`

`plot(x,20*z),grid` % Масштаб плоскости XOZ

`[x,y,z]=cylinder([2:10;10:18],100); surf(x,y,10*z)`

`[x,y,z]=cylinder([2:10;10:18],100); surf(7*x,y,10*z)`

`plot(7*x,y),grid` % Масштаб плоскости XOY

`[x,y,z]=cylinder([2:10;10:18],100); surf(7*x,13*y,10*z)`

`plot(7*x,13*y),grid` % Масштаб плоскости XOY