Графика

Основы MATLAB

Юдинцев В. В.

Кафедра теоретической механики

25 ноября 2021 г.

САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

Построение графиков в окне Workspace

Создать массив x:

```
1 x=0:0.1:10;
```

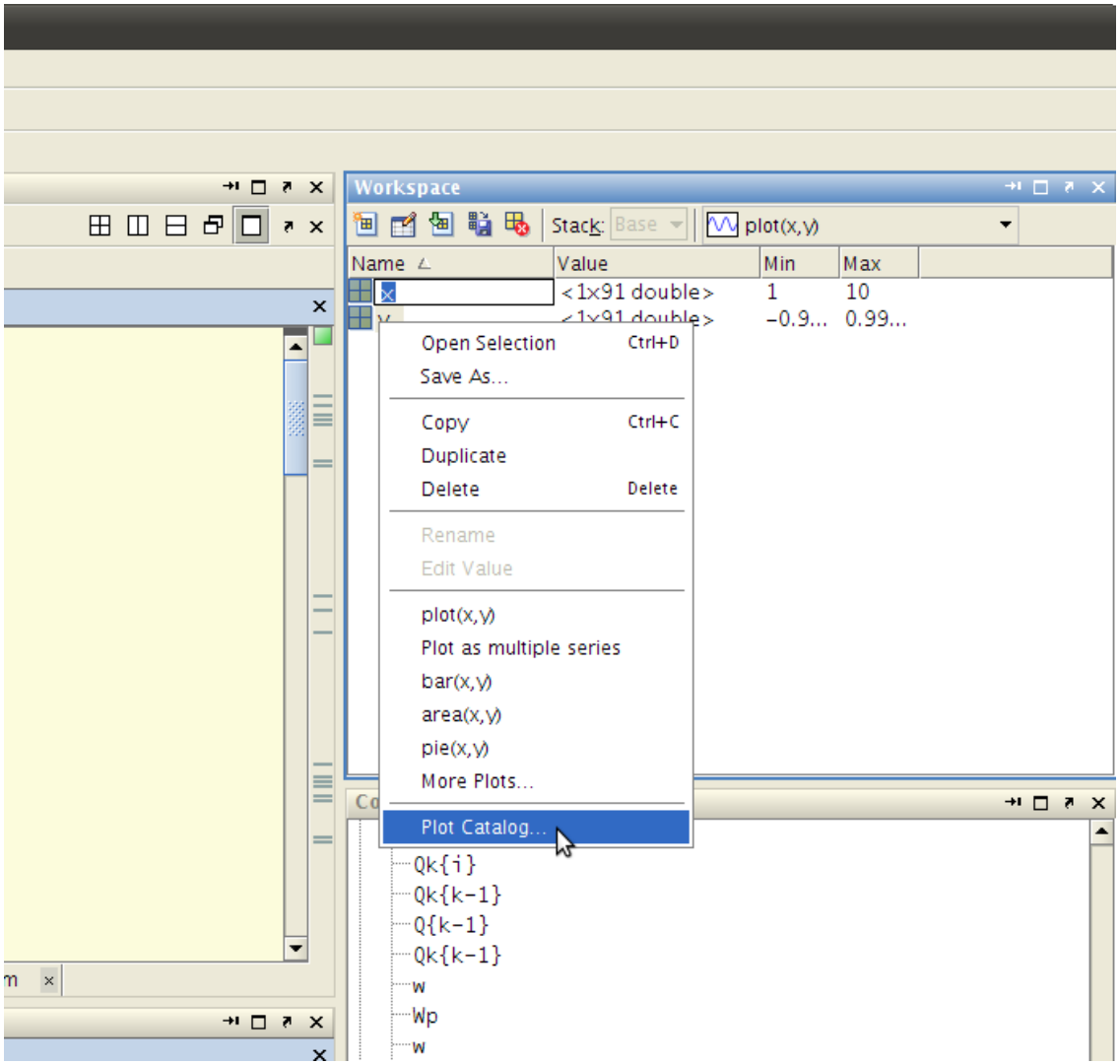
Создать массив y:

```
1 y=sin(x);
```

В окне **Workspace** выбрать переменные x и y (удерживая [Ctrl]) и по правой кн. мыши выбрать в контекстном меню `plot(x,y)`.

Типы графиков

В контектсном меню можно открыть каталог типов графиков



Типы графиков

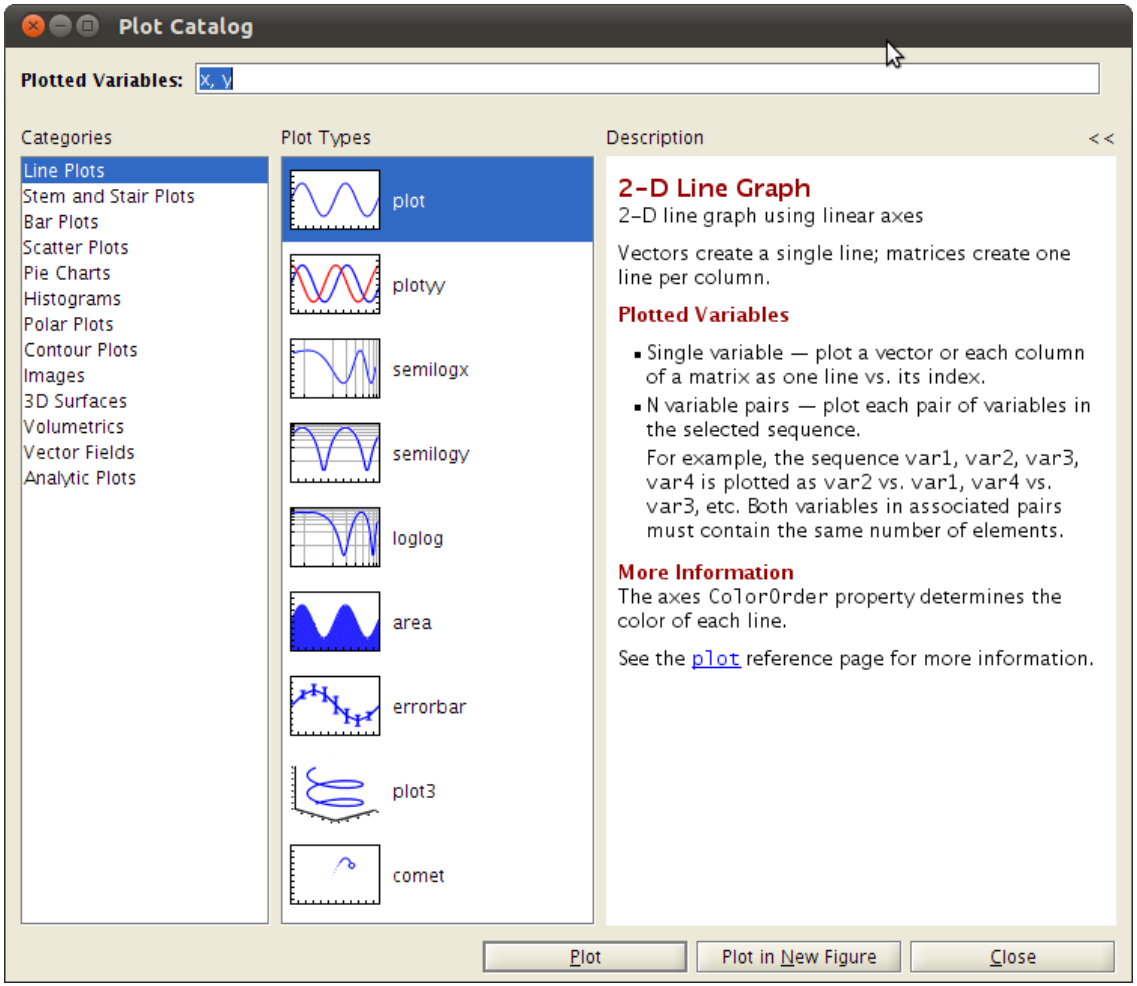
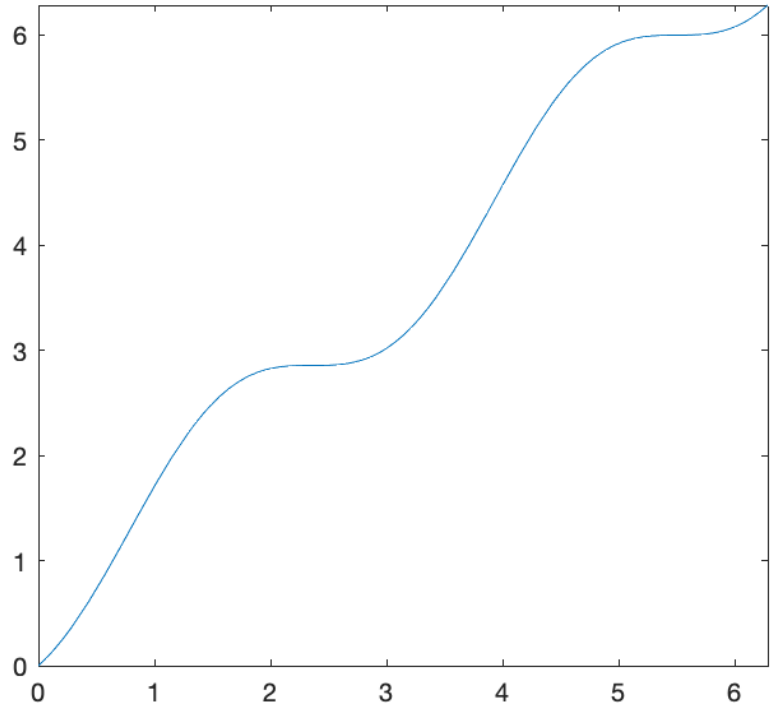


График функции-выражения

Для построения графиков функций, заданных выражением (m-файл, анонимная функция), используется функция `fplot`:

```
1 function y = myfun(x)
2     y = sin(x).^2 + x;
3 end
```

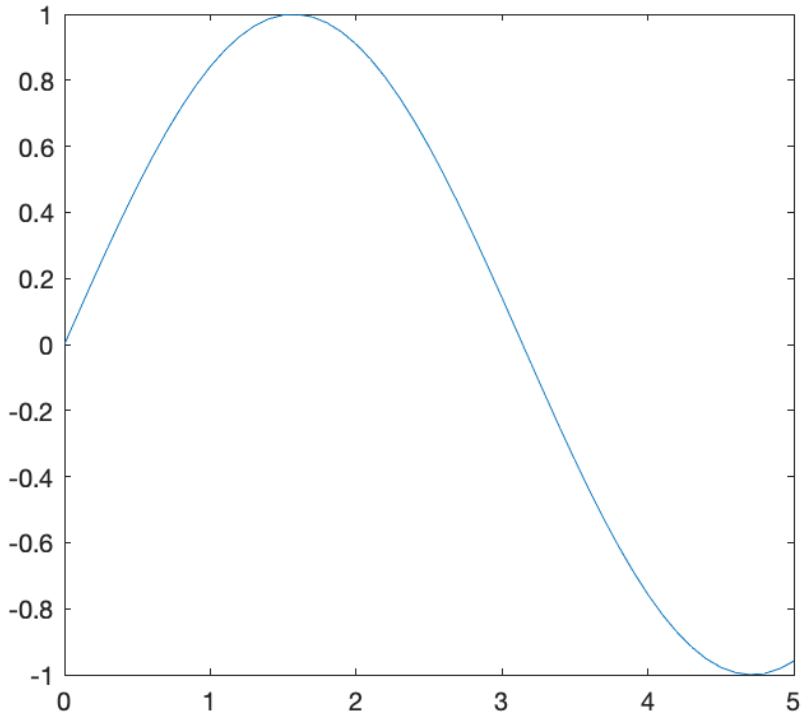
```
1 fplot(@myfun, [0 2*pi])
```



Функция plot

Для построения графиков табличных функций используется функция `plot`.

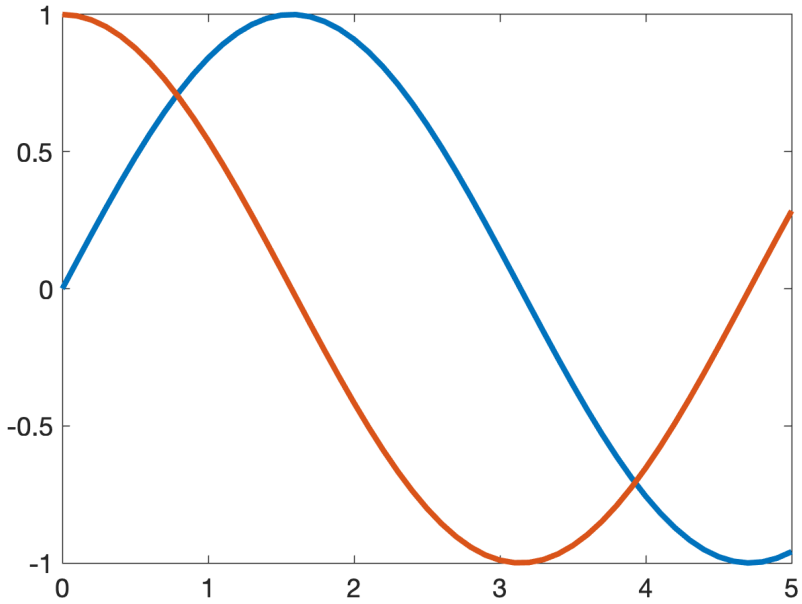
```
1 x=0:0.1:5;
2 y=sin(x);
3 plot(x,y);
```



Функция plot

Несколько графиков на одном рисунке

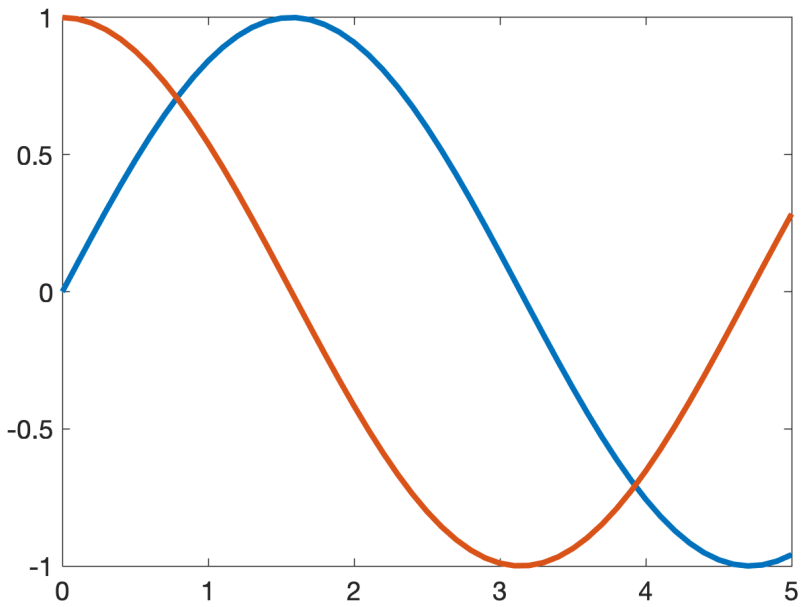
```
1 x=0:0.1:5;
2 y1=sin(x);
3 y2=sin(x);
4 plot(x,y1,x,y2);
```



Функция plot

Построение нескольких графиков на одном рисунке, используя директивы hold on, hold off:

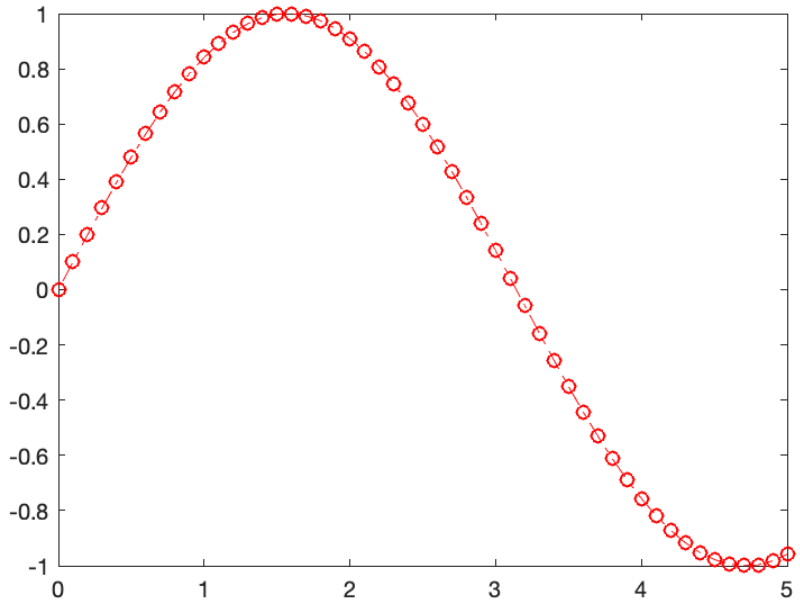
```
1  x=0:0.1:5;
2  y1=sin(x);
3  plot(x,y1);
4
5  hold on;
6
7  y2=sin(x);
8  plot(x,y2);
9
10 hold off;
```



Свойства линий

Третим аргументом функции `plot` можно передать строковую константу, описывающую свойства графика: цвет, тип маркера и тип линии.

```
1 x=0:0.1:5;
2 y=sin(x);
3 plot(x,y,'ro-.');
```



Тип линии

`-`	сплошная
`--`	штрих
`-.`	штрих-точка
`:`	точки

Коды цвета

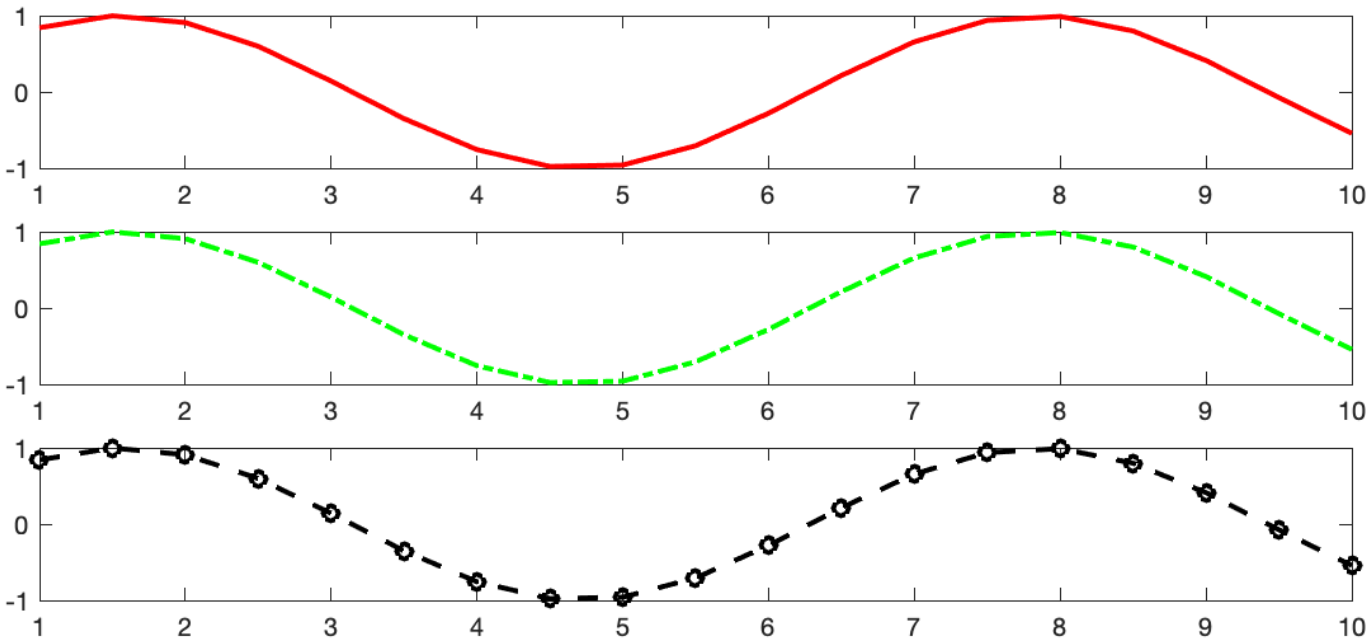
Символ цвета	Цвет графика
y	желтый (yellow)
g	зеленый (green)
m	малиновый (magenta)
b	синий (blue)
c	цианový (cyan)
w	белый (white)
r	красный (red)
k	черный (black)

Маркеры

Символ	Маркер
.	точка
o	кружок
x	косоугольный крест
+	прямоугольный крест
*	снежинка
s	квадрат
d	ромб
v	треугольник вершиной вниз
^	треугольник вершиной вверх
>	треугольник вершиной вправо
<	треугольник вершиной влево
p	пятиугольник
h	шестиугольник

Стили

```
1 x = 1:0.5:10;
2 subplot(311);plot(x,sin(x),'r-');
3 subplot(312);plot(x,sin(x),'g-.');
4 subplot(313);plot(x,sin(x),'ko--');
```



Подписи к осям

```
1 plot(x,y1,x,y2);
2 grid on;
3 title('Скорости точек 1 и 2');
4 xlabel('t, c');
5 xlabel('v, м/с');
6 legend('точка 1','точка 2',pos);
```

Последний необязательный аргумент функции `legend` может задавать положении “легенды” в окне графика:

- -1 - вне графика
- 0 - лучшее расположение
- **1**,2,3,4 - от верхнего правого до нижнего правого угла (по часовой стрелке)

Границы осей

При построении графика функции MATLAB автоматически выбирает масштаб осей. Для “ручного” управления используются функции

- `xlim([xmin, xmax])`
- `ylim([ymin, ymax])`

Комбинирование графиков

Создание нового окна для графика - команда `figure`:

```
figure;
plot(t,x1);
figure;
plot(t,x2);
```

Последнее созданное окно становится текущим, команды форматирования графика (`title(..),colormap(...)`) влияют на график в этом окне.

Функция subplot

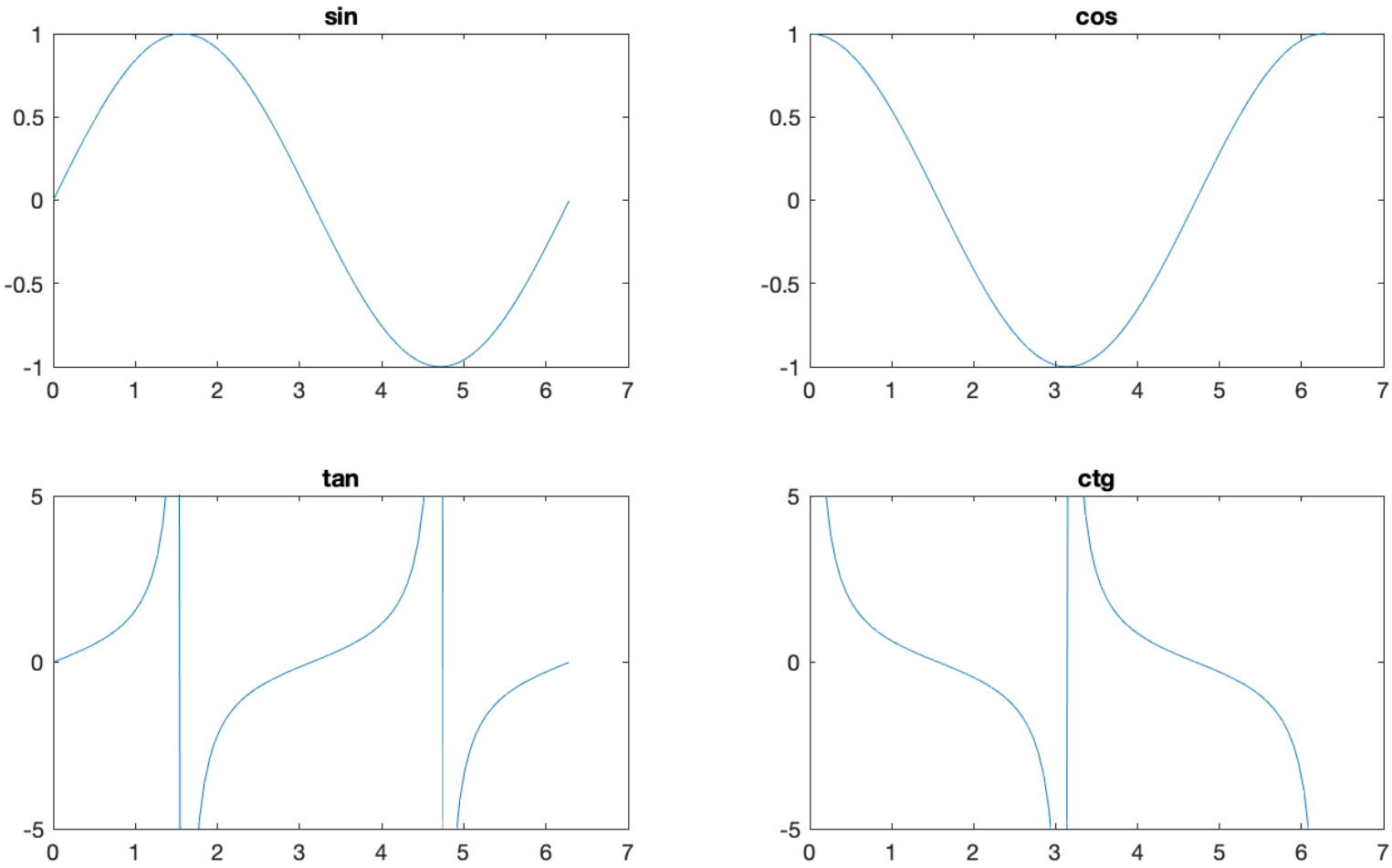
Функция `subplot(mnp)` или `subplot(m, n, p)`, где mnp - 3 цифры, производит разбивку графического окна (`figure`) на несколько областей, создавая при этом новые объекты axes.

- m указывает, на сколько частей разбивается окно по горизонтали,
- n - по вертикали,
- p - номер подокна, куда будет выводиться после функции,

subplot

```
x  = linspace(0,2*pi,100);
y1 = sin(x);
y2 = cos(x);
y3 = tan(x);
y4 = cot(x);

subplot(2,2,1);
plot(x,y1); title('sin');ylim([-1,1]);
subplot(2,2,2);
plot(x,y2); title('cos');ylim([-1,1]);
subplot(2,2,3);
plot(x,y3); title('tan');ylim([-5,5]);
subplot(2,2,4);
plot(x,y4); title('ctg');ylim([-5,5]);
```

subplot

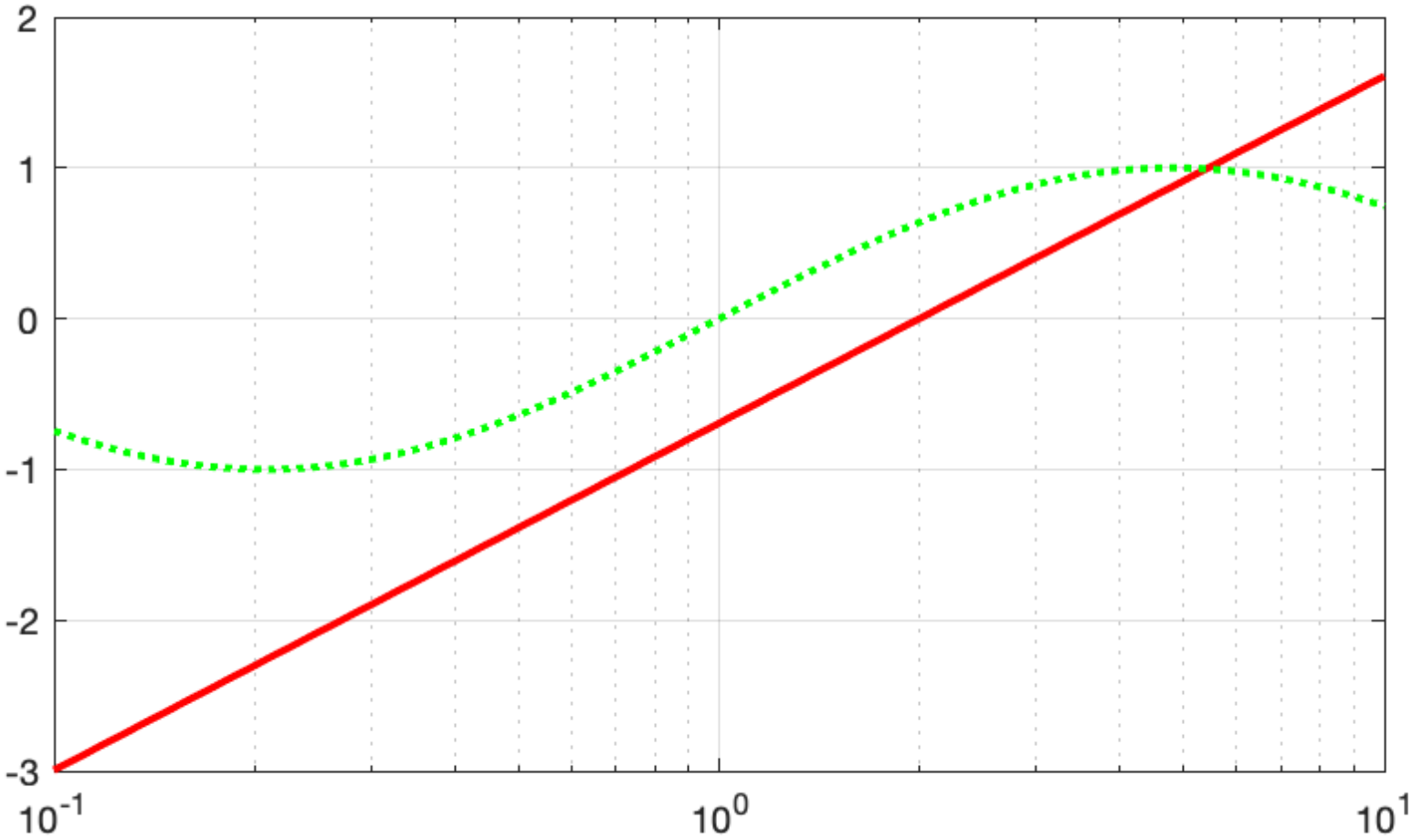


Логарифмическая шкала

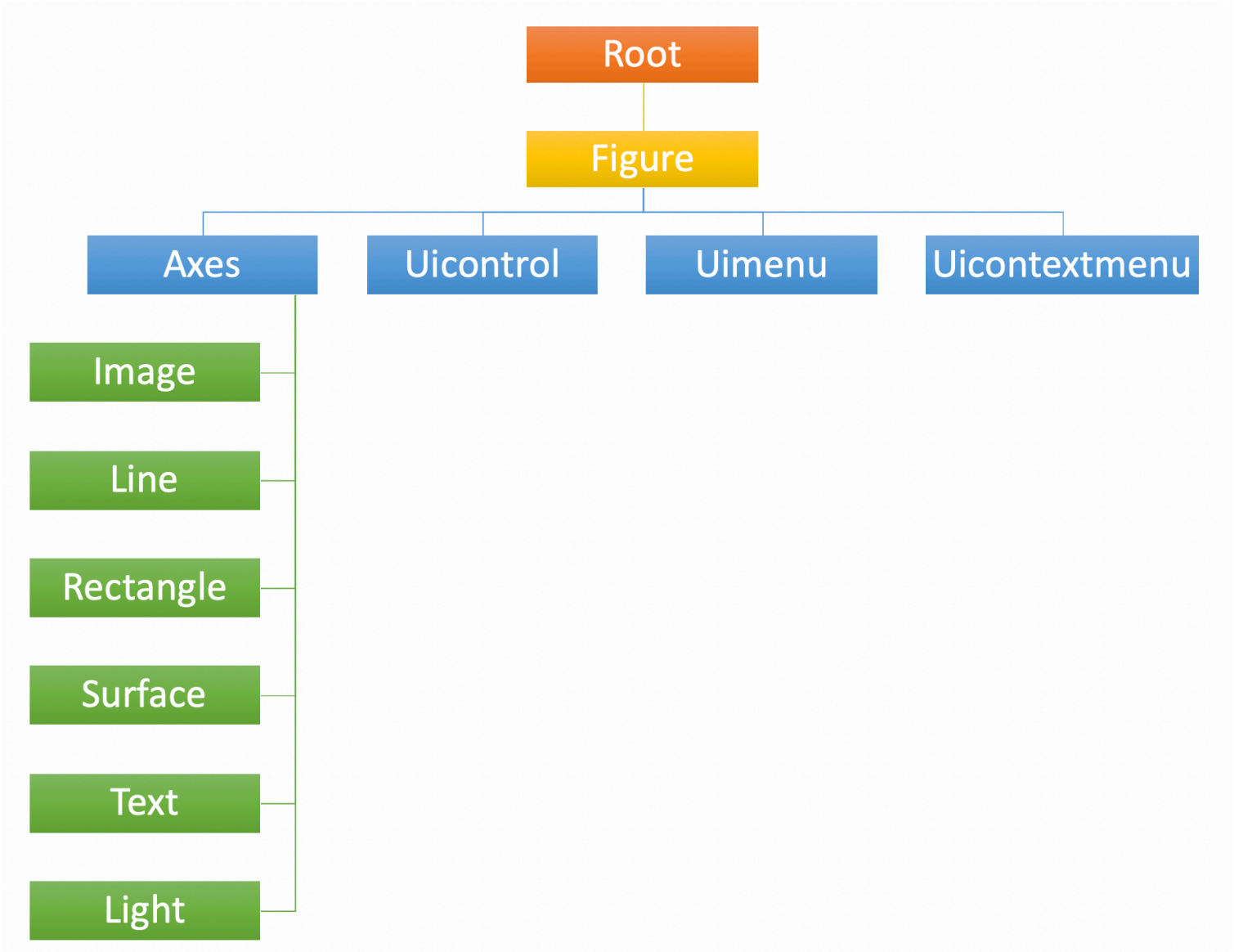
- `loglog` - логарифмический масштаб по обеим осям.
- `semilogx` - логарифмический масштаб по оси абсцисс.
- `semilogy` - логарифмический масштаб по оси ординат.

```
1 x=0.1:0.01:10;
2 y=log(0.5*x);
3 g=sin(log(x));
4 semilogx(x,y,'r-', x,g, 'g:');
```

Логарифмическая шкала



Иерархия графических классов



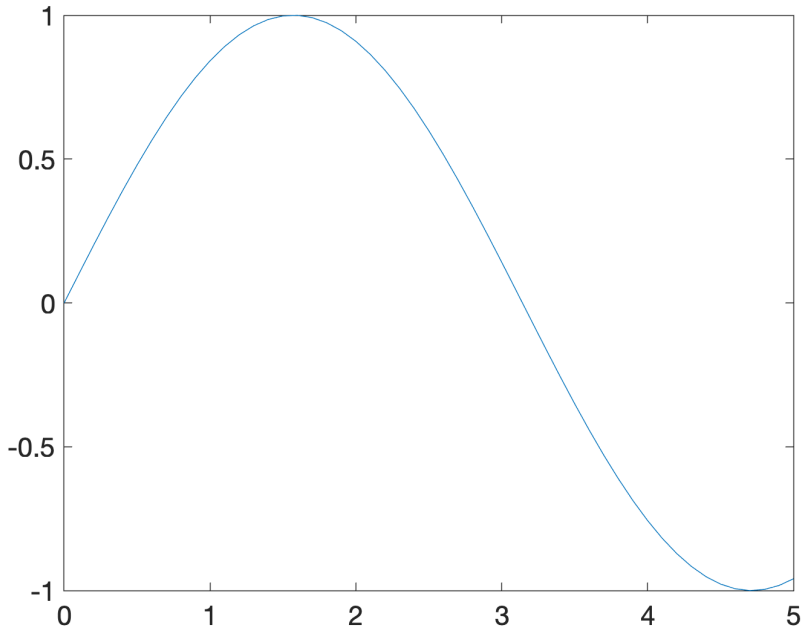
Иерархия графических классов

- Root
  Экран компьютера
- Figure
  Окно с рисунком.
- Axes
  Координатная область, в которой отображаются графики, текст, ...
  (результаты работы функций plot, line, принадлежат объекту axes.)

figure и axes

```
figure('Color','white');
x = 0:0.1:5;
plot(x,sin(x))
```

```
x = 0:0.1:5;
plot(x,sin(x));
```



Изменяем свойства текущего объекта figure (gcf - ссылка на текущий активный объект figure):

```
set(gcf,'Color','white');
```

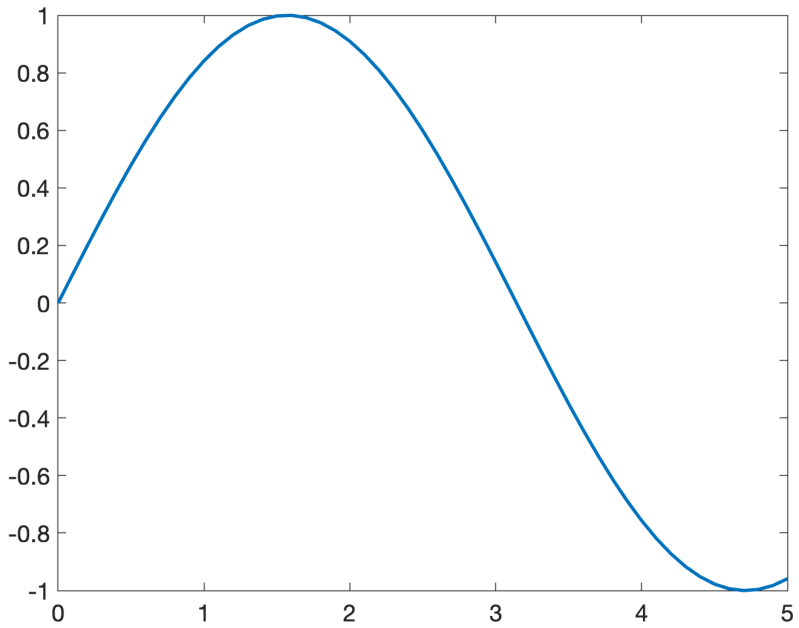
Изменяем свойства текущего объекта axes (gca - ссылка на текущий активный объект axes):

```
set(gca,'FontSize',16);
```

figure и axes

```
figure('Color','white');
x = 0:0.1:5;
h = plot(x,sin(x));
set(h,'LineWidth',2)
```

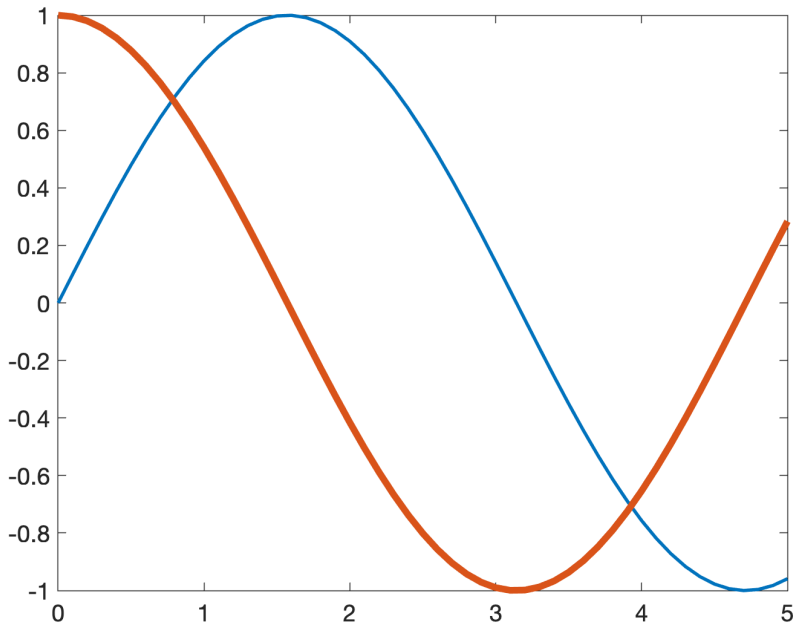
```
x = 0:0.1:5;
plot(x,sin(x),'LineWidth',2);
set(gcf,'Color','white');
set(gca,'FontSize',16);
```



```
x = 0:0.1:5;
h = plot(x,sin(x),'LineWidth',2);

>> h

  Line with properties:

            Color: [0 0.4470 0.7410]
        LineStyle: '-'
        LineWidth: 2
           Marker: 'none'
       MarkerSize: 6
  MarkerFaceColor: 'none'
            XData: [1×51 double]
            YData: [1×51 double]
            ZData: [1×0 double]
```

figure и axes

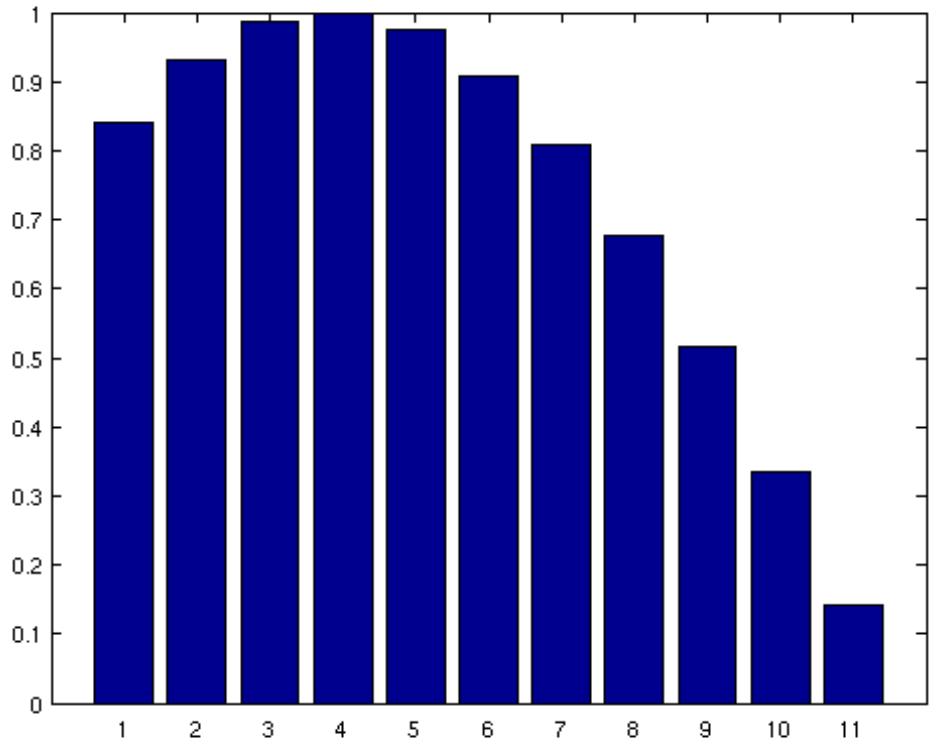
```
figure('Color','white');
x = 0:0.1:5;
h = plot(x,sin(x),x,cos(x));
set(gca,'FontSize',14)
set(h,{'LineWidth'},{2;4});
```



Столбчатые диаграммы

```
1 x=1:0.2:3;
2 y=sin(x);
3 bar(x,y);
```

```
1 x=1:0.2:3;
2 y=sin(x);
3 bar(y);
```

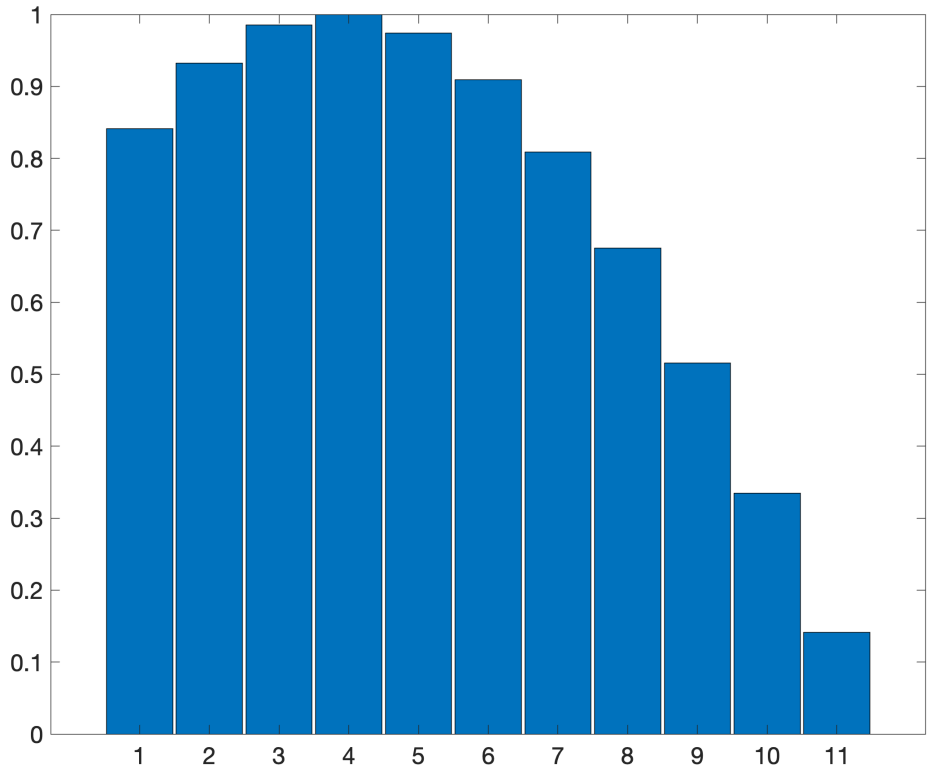


Столбчатые диаграммы

Третий (второй) аргумент задает ширину столбца в долях от полной ширины

```
1 x=1:0.2:3;
2 y=sin(x);
3 bar(x,y,0.5);
```

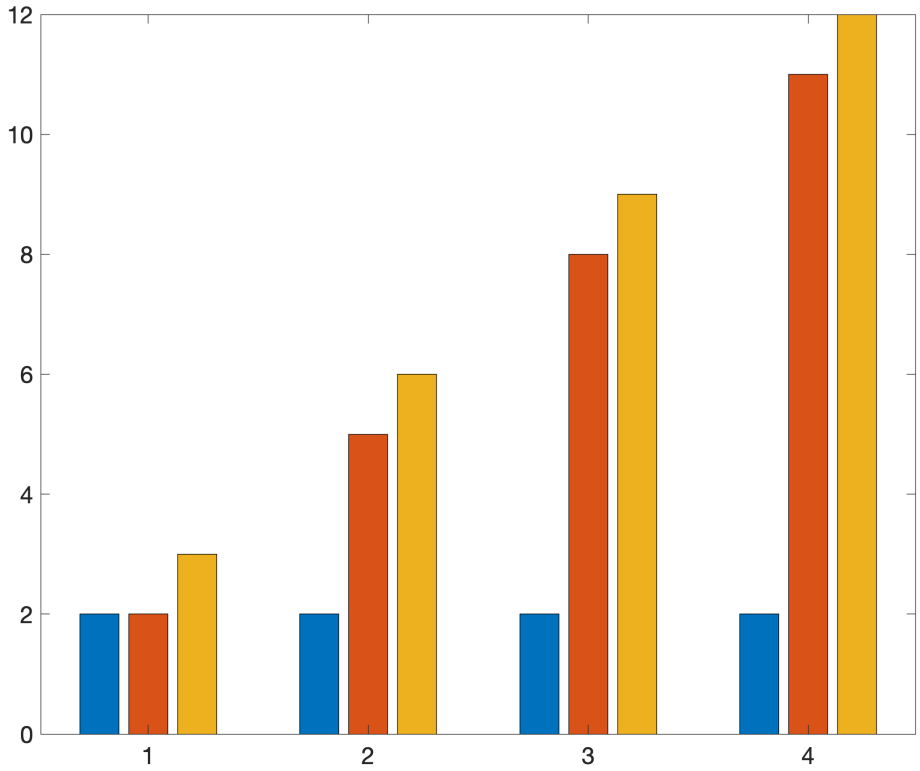
```
1 x=1:0.2:3;
2 y=sin(x);
3 bar(y,0.95);
```



Столбчатые диаграммы

Группы столбцов

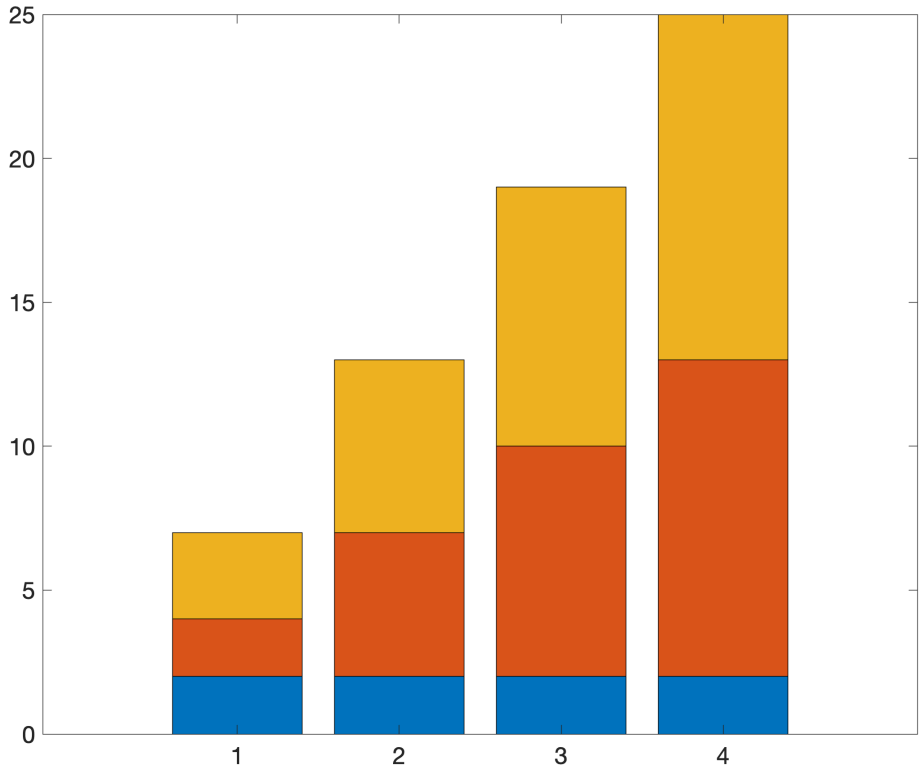
```
1  y = [2  2  3;
2       2  5  6;
3       2  8  9;
4       2  11  12];
5  bar(y);
```



Столбчатые диаграммы

Графики с накоплением

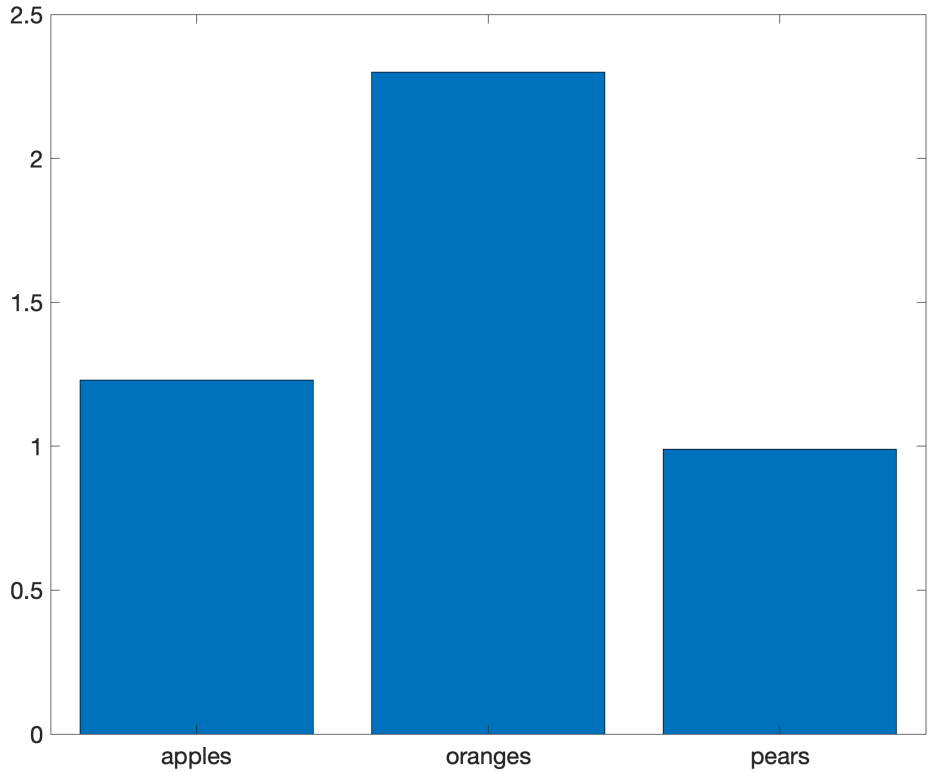
```
1 y = [2  2  3;
2       2  5  6;
3       2  8  9;
4       2  11  12];
5 bar(y,'stacked')
```



Столбчатые диаграммы

Подписи осей

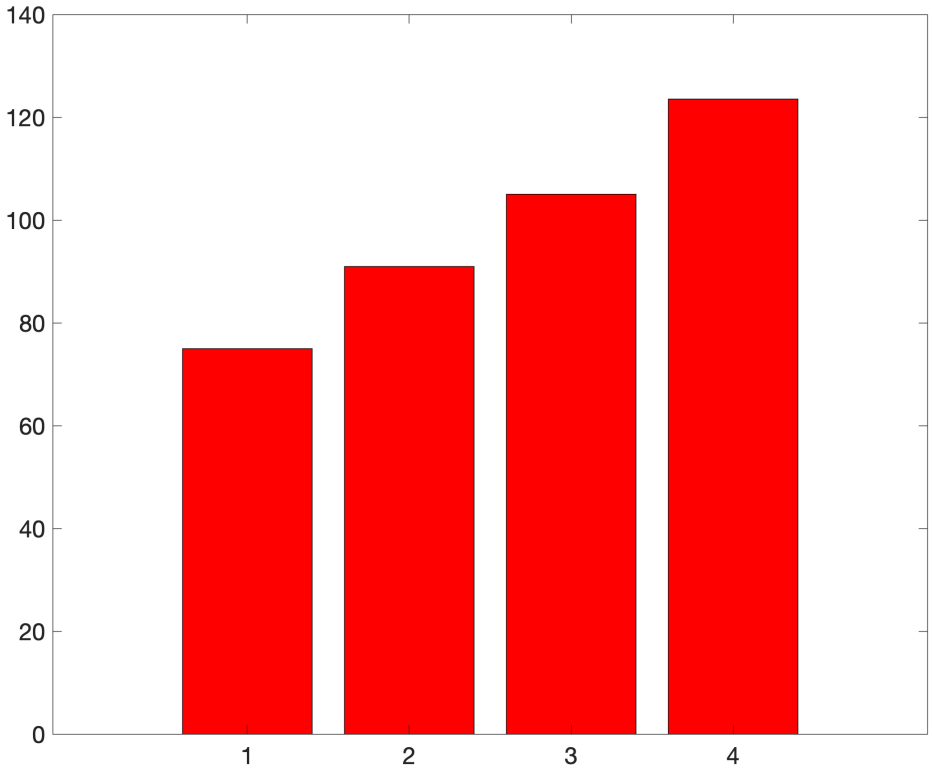
```
1 c = categorical({'
      apples','pears','
      oranges'});
2 prices = [1.23 0.99
      2.3];
3 bar(c,prices)
```



Столбчатые диаграммы

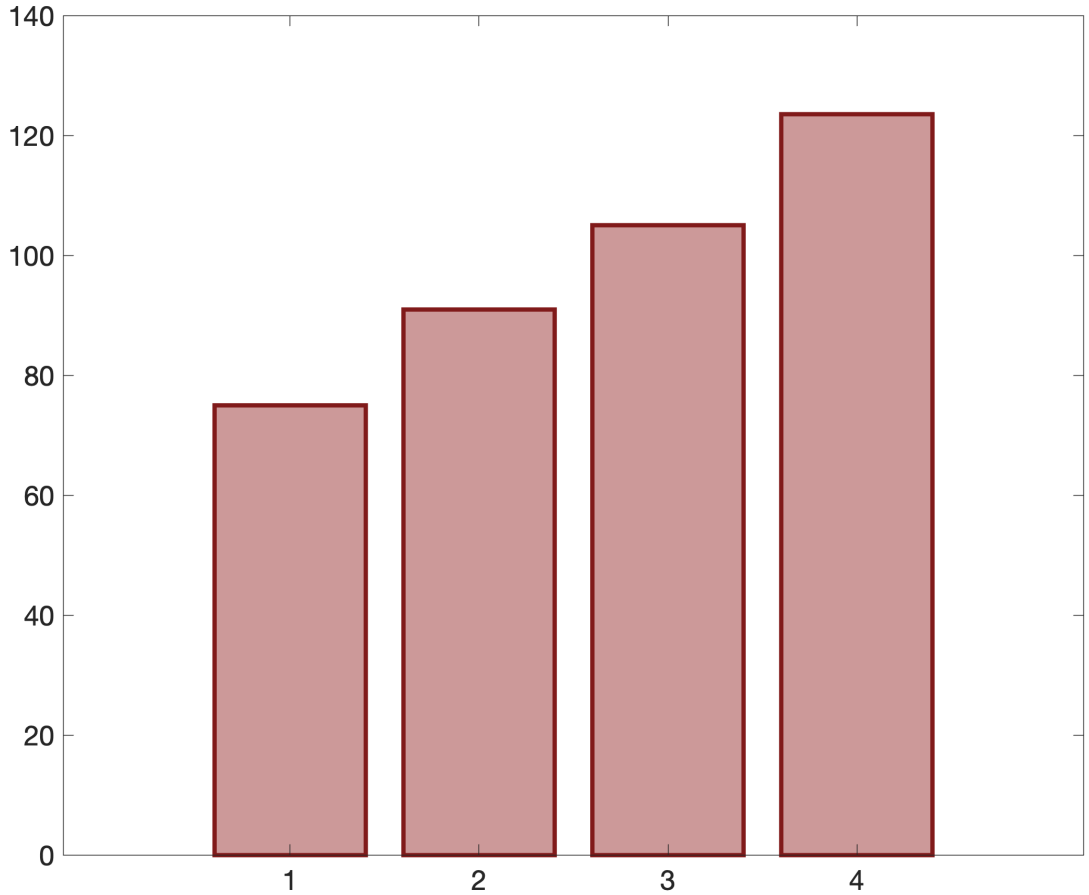
Цвет

```
1 y = [75 91 105 123.5];
2 bar(y,'r')
```



Столбчатые диаграммы

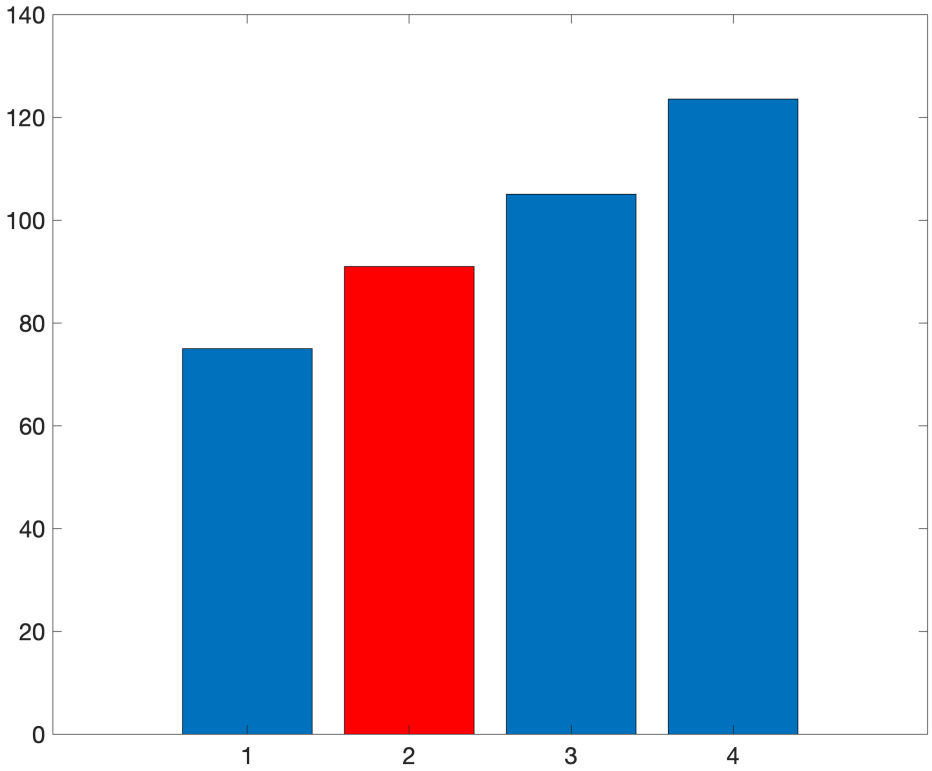
```
1 y = [75 91 105 123.5];
2 bar(y,'FaceColor',[0.8 0.6 0.6],...
3       'EdgeColor',[0.5 0.1 0.1],'LineWidth',3);
```



Столбчатые диаграммы

Изменение цвета отдельных столбцов

```
1 y = [75 91 105 123.5];
2 b = bar(y);
3 b.FaceColor = 'flat';
4 b.CData(2,:) = [1 0 0];
```

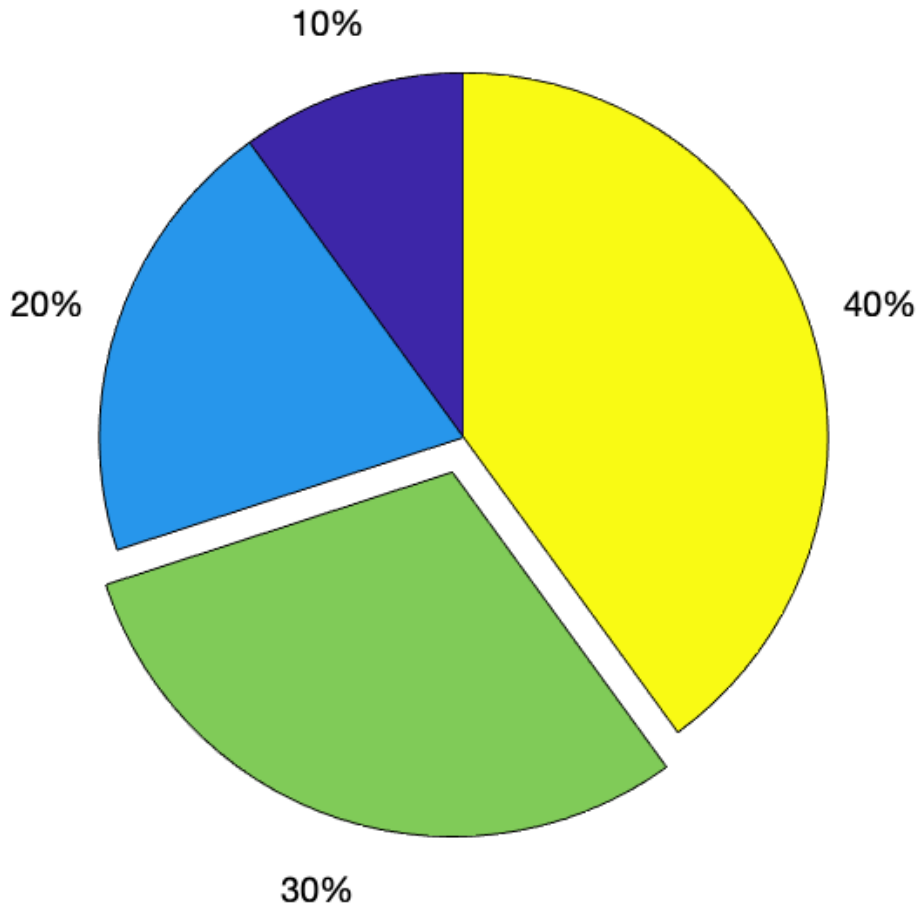


Круговые диаграммы

```
1 x=[1,2,3,4];
2 pie(x)
```

Второй аргумент – логический массив, указывающий на необходимость изображения соответствующего сектора отдельно от круговой диаграммы

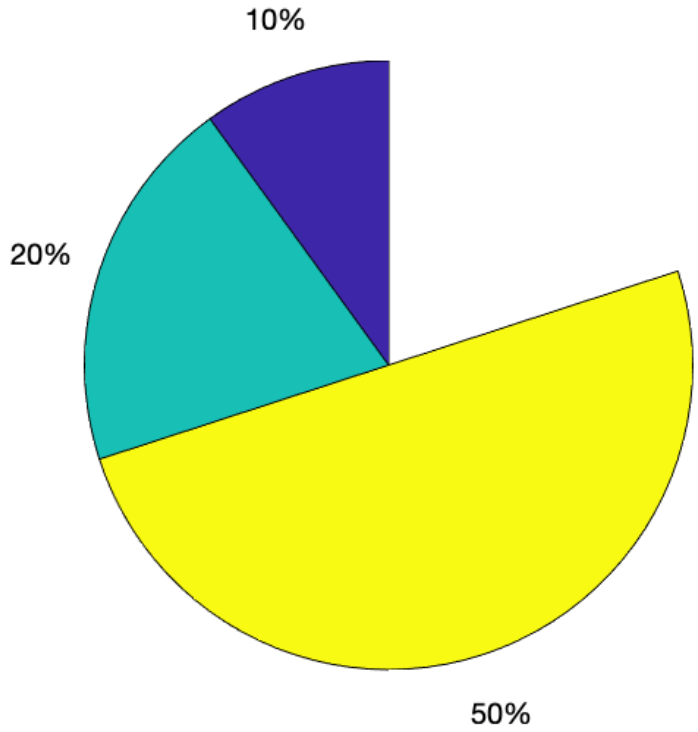
```
1 y=[0,0,1,0];
2 pie(x,y)
```



Круговые диаграммы

Если сумма элементов массива данных больше или равна единице, то эта сумма принимается за 100%, в противном случае строится диаграмма с пропущенным сектором.

```
1 x=[0.1,0.2,0.5];
2 pie(x)
```



Круговые диаграммы

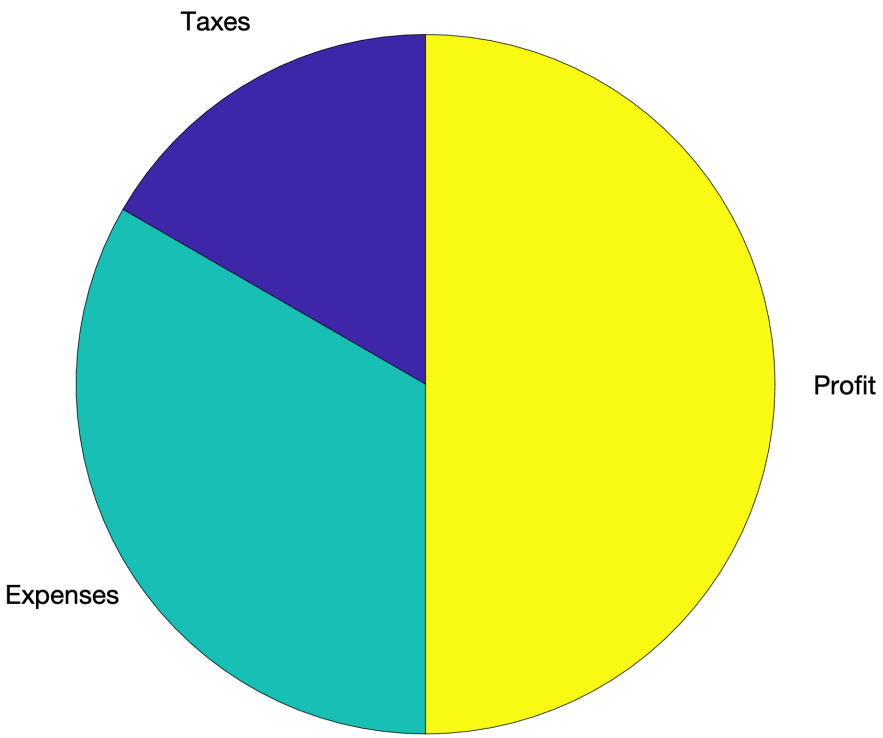
```
X = 1:3;
labels = {'Taxes','Expenses','Profit'};
hp = pie(X,labels)

>> hp
  1×6 graphics array:

    Patch     Text      Patch     Text      Patch     Text
```

Круговые диаграммы

```
1 set(hp(2:2:6),'FontSize',18);
```

или

```
1 set(findobj(hp,'type','text'),'FontSize',18);
```



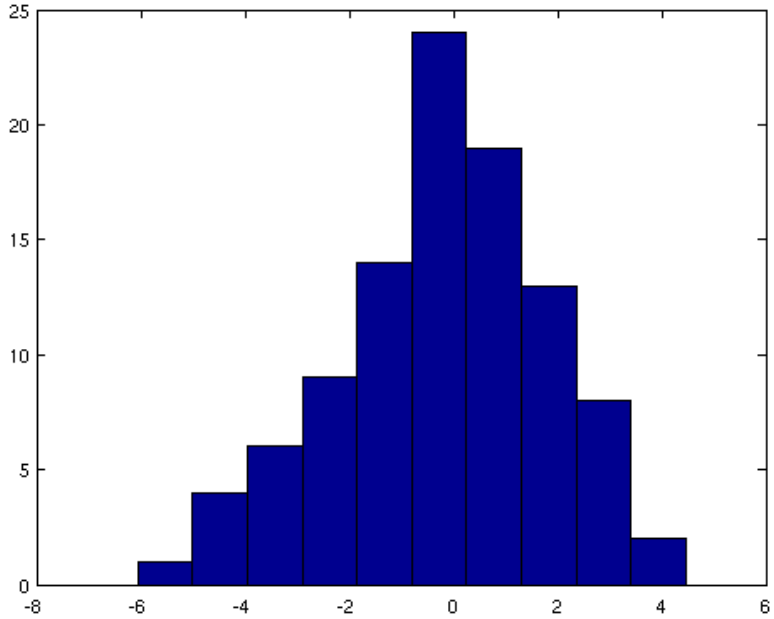
Гистограммы

Вектор-строка 90 случайных чисел, распределённых по нормальному закону с $m = 0$ и $\sigma = 2$

```
1 y=random('norm',0,2,1,90);
```

Функция `hist` делит интервал y на 10 частей и строит гистограмму:

```
1 hist(y)
```



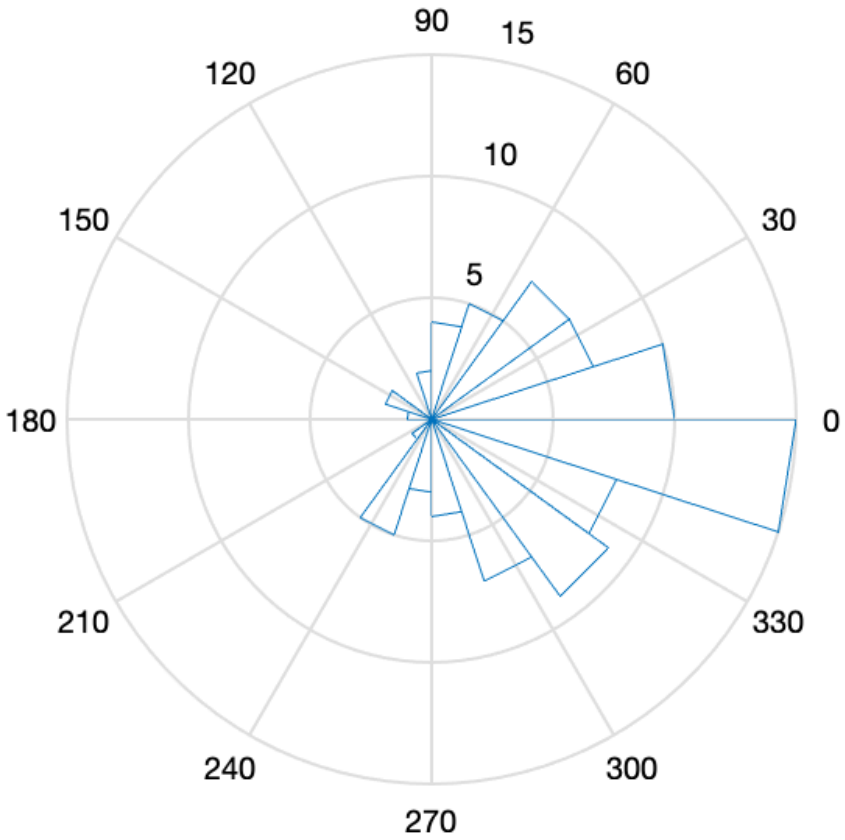
Круговая гистограмма

Генерируем вектор-строку 90 случайных чисел, распределённых по нормальному закону с $m = 0$ и $\sigma = 1$

```
1 y=random('norm',0,1,1,90);
```

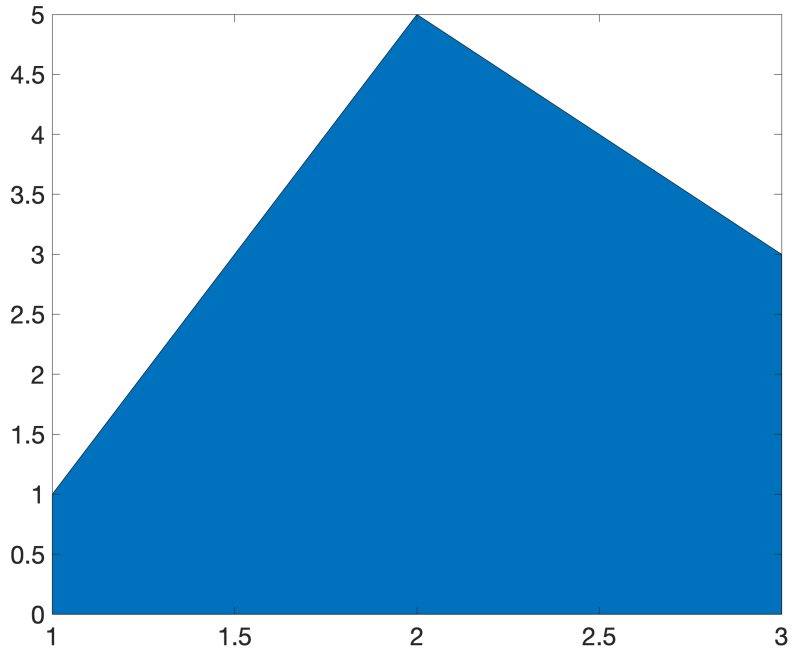
Функция `rose` строит гистограмму в полярных координатах, предполагая, что элементы массива `y` заданы в радианах.

```
1 rose(y)
```



area

```
1 Y = [1, 5, 3];
2 area(Y);
```



area

```
1 Y = [1, 5, 3;
2       3, 2, 7;
3       1, 5, 3;
4       2, 6, 1];
5 area(Y);
```

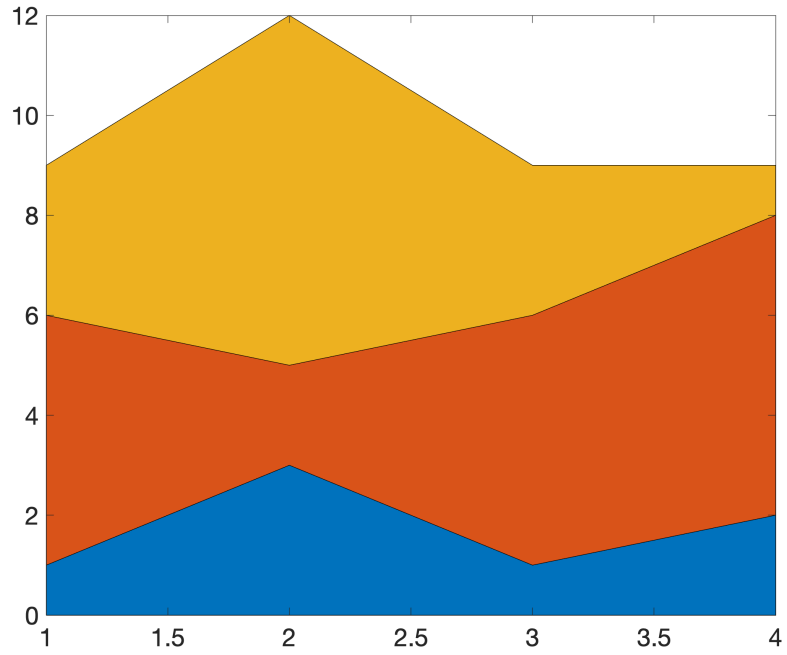


График с оценкой погрешности

```
1 x = 0:10:40;
2 y = [20 30 45 40 60];
3 err = [1 3 5 3 5];
4 errorbar(x,y,err,...
5 'LineWidth',2);
```

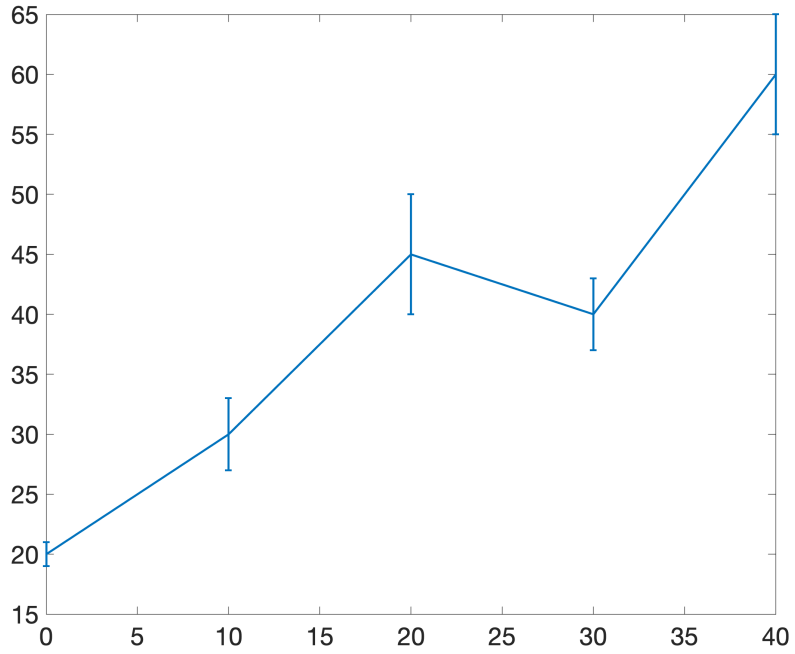


График с оценкой погрешности

```
1 x = 0:10:40;
2 y = [20 30 45 40 60];
3 err = [1 3 5 3 5];
4 errorbar(x,y,err,...
5 'horizontal',...
6 'LineWidth',2);
```

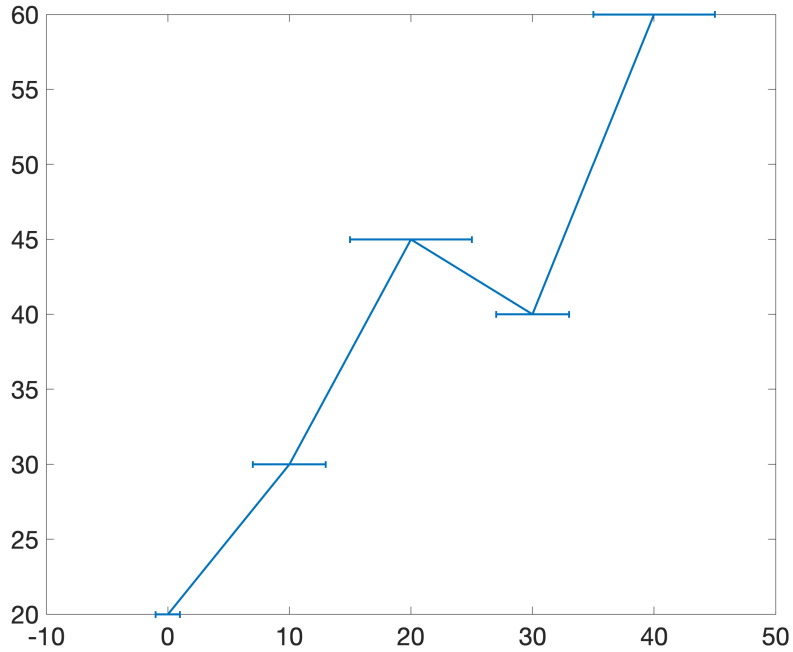
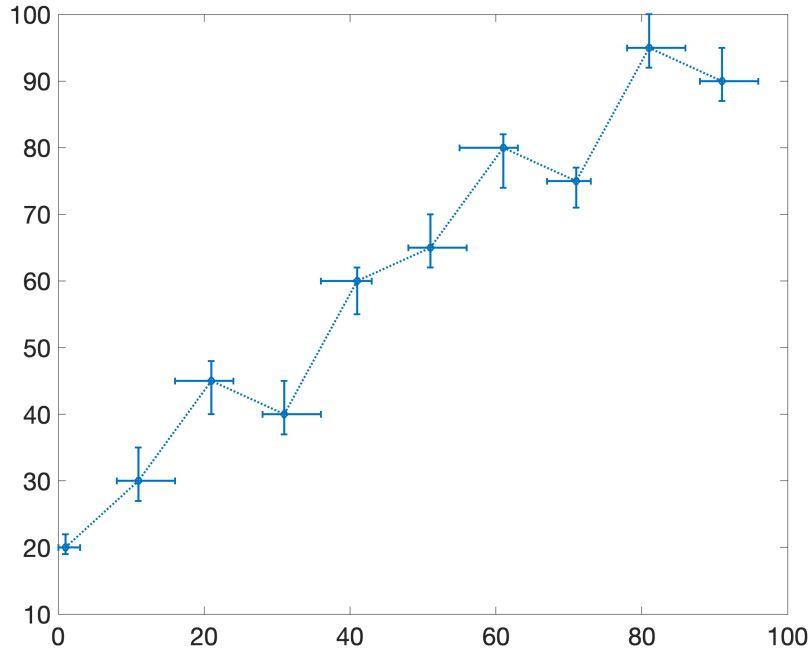


График с оценкой погрешности

```
x = 1:10:100;
y = [20 30 45 40 60 65...
     80 75 95 90];
yneg = [1 3 5 3 5 3 6 4 3 3];
ypos = [2 5 3 5 2 5 2 2 5 5];
xneg = [1 3 5 3 5 3 6 4 3 3];
xpos = [2 5 3 5 2 5 2 2 5 5];
errorbar(x,y,yneg,ypos,...
         xneg,xpos,'o:');
```

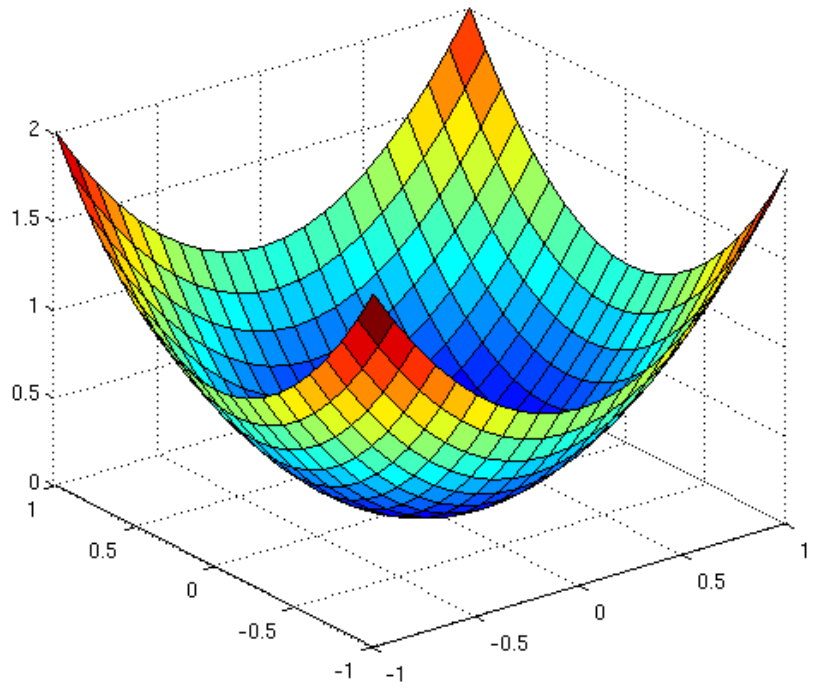


Сетки и поверхности

```
1 pp=-1:0.1:1;
2 [X,Y]=meshgrid(pp,pp);
3 Z=X.^2+Y.^2;
4 mesh(X,Y,Z);
5 colorbar;
```

или (раскрашенная сеточная поверхность)

```
1 surf(X,Y,Z);
```

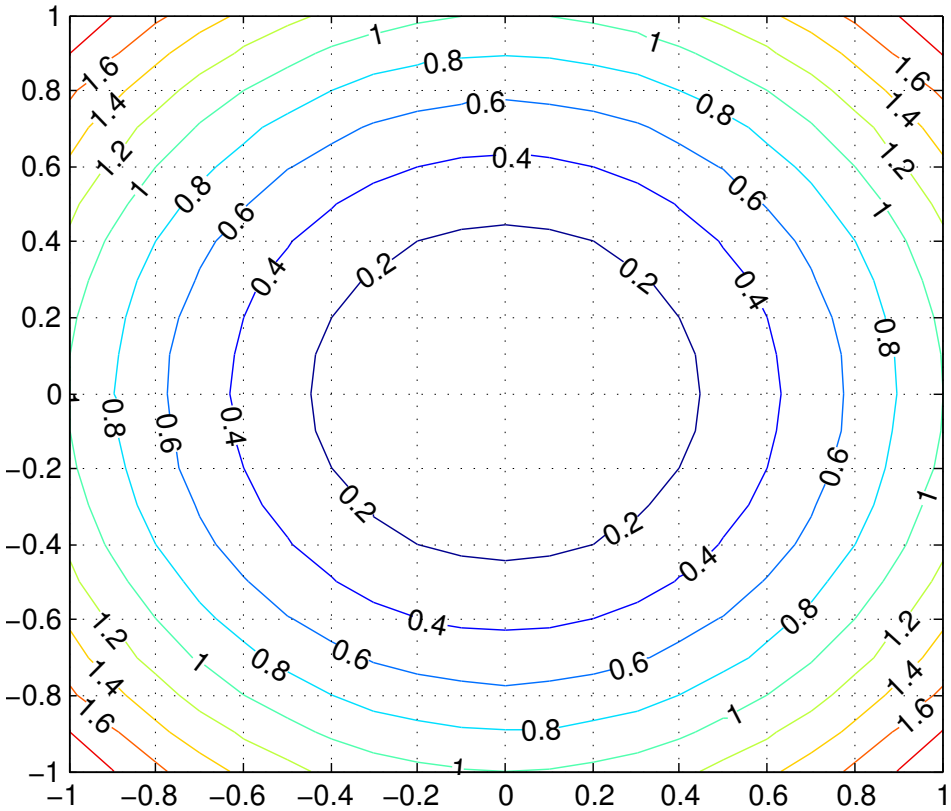


Функция meshgrid

```
[x, y] = meshgrid(0:0.1:0.4,   0:0.2:0.4);

x =

         0    0.1000    0.2000    0.3000    0.4000
         0    0.1000    0.2000    0.3000    0.4000
         0    0.1000    0.2000    0.3000    0.4000


y =

         0         0         0         0         0
    0.2000    0.2000    0.2000    0.2000    0.2000
    0.4000    0.4000    0.4000    0.4000    0.4000
```

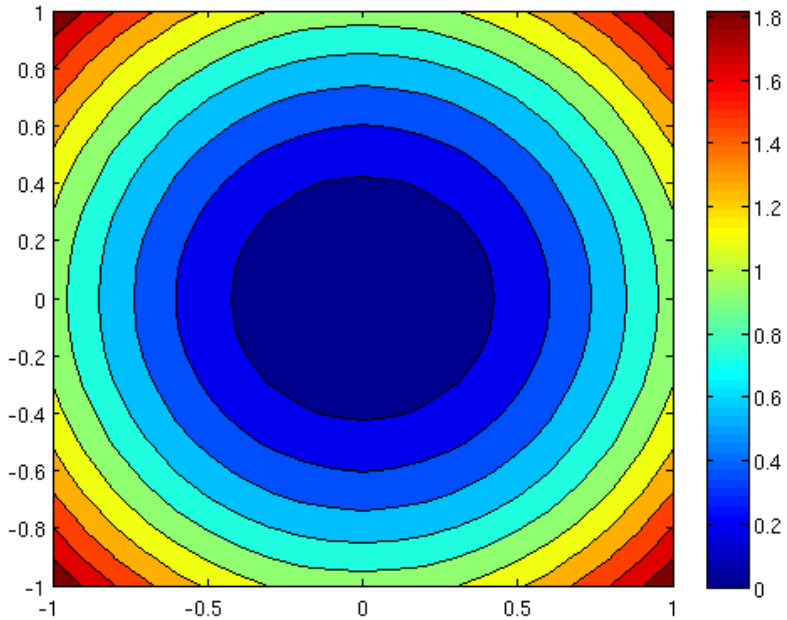
Контурные графики

```
1 [X,Y]=meshgrid(-1:0.1:1,-1:0.1:1);
2 Z=X.^2+Y.^2;
3 contour(X,Y,Z);
4 [c,h]=contour(X,Y,Z));
5 clabel(c,h); grid on;
```



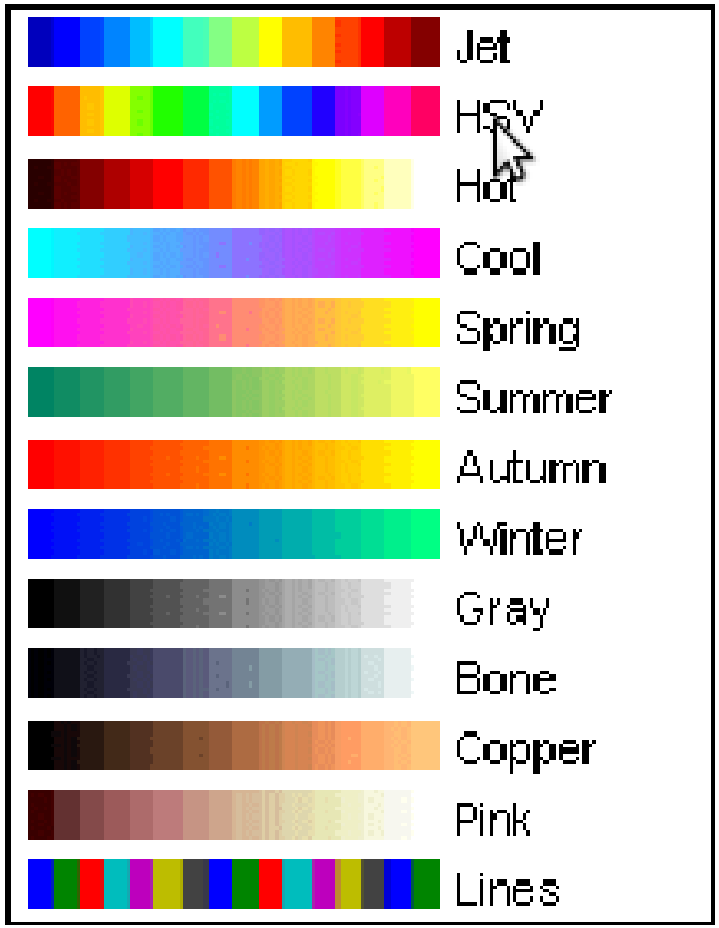
Контурный график с заливкой

```
1 contourf(X,Y,Z,10);
2 colorbar;
```



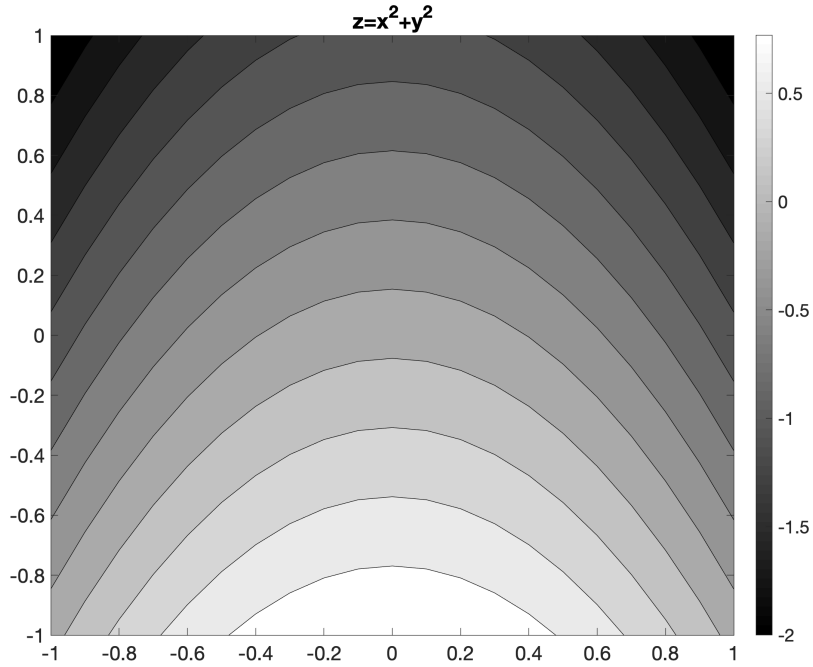
Палитра

```
1 contourf(X,Y,Z,10);
2 colorbar;
```



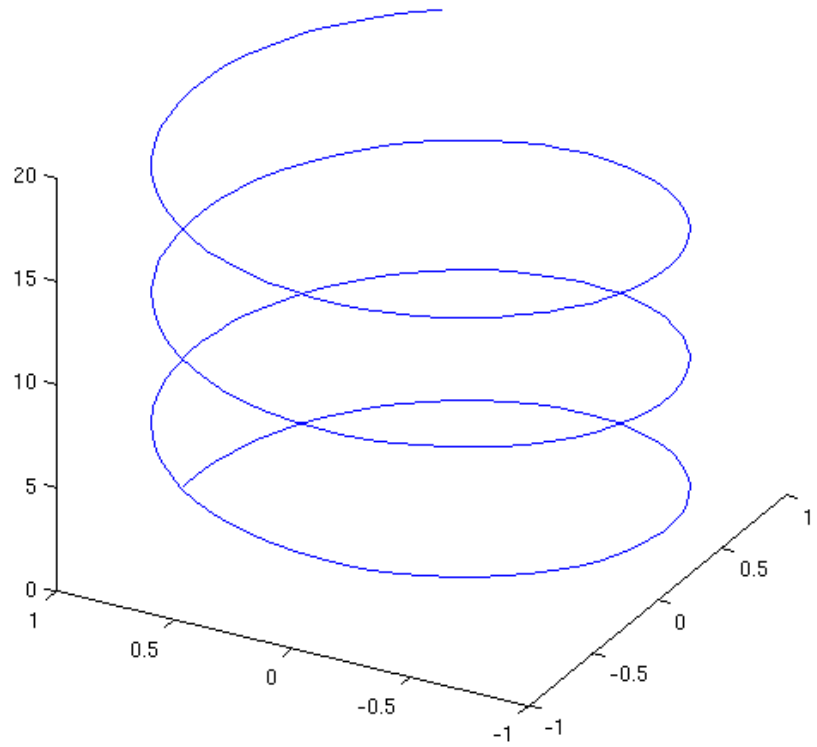
Палитра

```
1 pp=-1:0.1:1;
2 [X,Y]=meshgrid(pp,pp);
3 Z=-X.^2-Y;
4 contourf(X,Y,Z,12);
5 colorbar;
6 colormap(gray);
7 title('z=x^2+y^2');
```



Параметрические кривые в пространстве

```
1 t=0:0.1:20;
2 x=cos(t);
3 y=sin(t);
4 z=t;
5 plot3d(x,y,z);
```



Поворот пространственного графика

Задать новое положение наблюдателя:

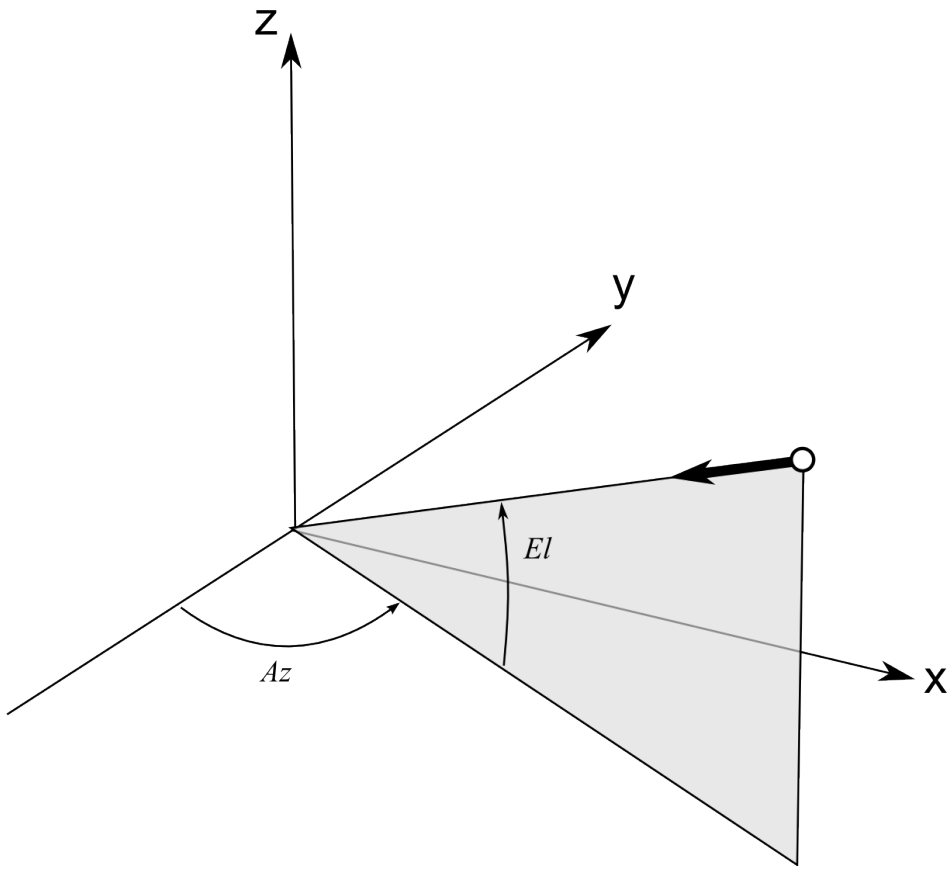
```
1 view(az, el);
```

Определить положение наблюдателя

```
1 [Az, El]=view;
```

Взгляд направлен вдоль оси y

```
1 view(0,0);
```



Освещенная поверхность

Отобразить поверхность с заданным азимутом и возвышением источника света:

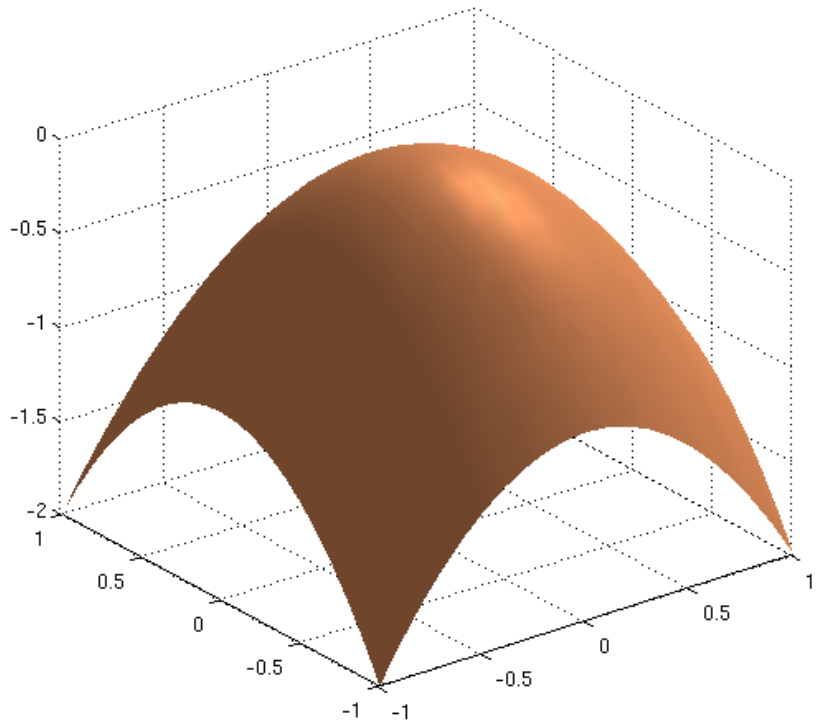
```
1 surfl(X,Y,Z,[Az,El]);
```

Палитра copper:

```
1 colormap('copper');
```

Сглаживать цвета:

```
1 shading interp;
```



Анимированные графики. Функция comet

На плоскости:

```
t=0:0.01:10;
x=t-sin(t);
y=1-cos(t);
comet(x,y);
```

В пространстве:

```
t=0:0.01:10;
x=cos(t);
y=sin(t);
z=t;
comet3(x,y,z);
```

Экспорт в файл

Экпорт в растровый формат (png)

```
t=0:0.01:10;
x=cos(t);
y=sin(t);
plot(x,y);

print('filename','-dpng','-r300');
```

Экпорт в векторный формат (pdf)

```
figure('PaperUnits','Centimeters','PaperSize',[16,10]);
t=0:0.01:10;
x=cos(t);
y=sin(t);
plot(x,y);

print('plot_figure','-dpdf','-fillpage');
```