

Графика в MatLab

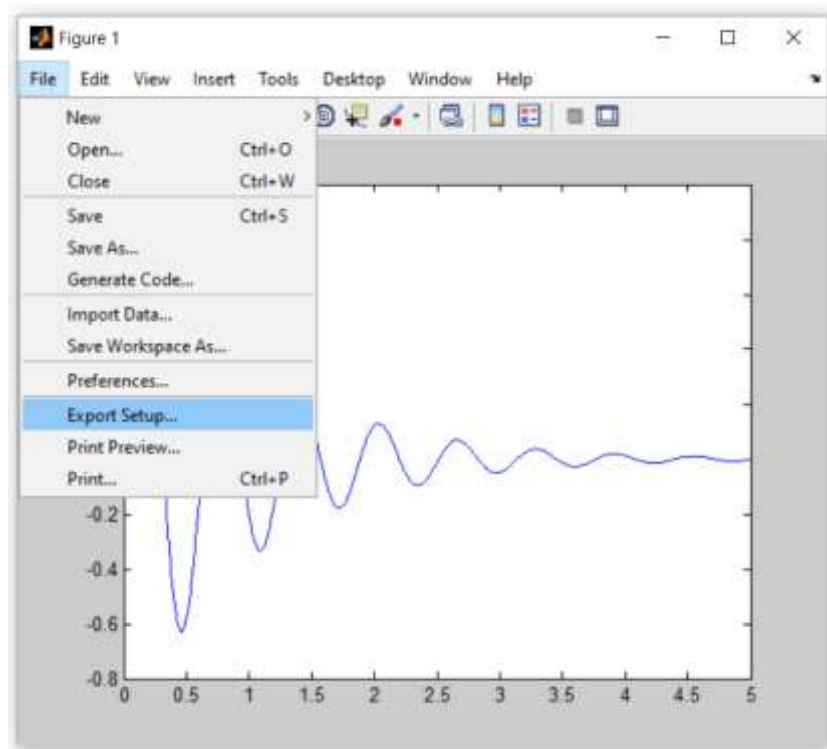
Система MATLAB предоставляет огромное количество графических средств. К ним относятся команды построения простых графиков функций, комбинированные и презентационные графики, элементы анимации и средства проектирования графического пользовательского интерфейса (GUI).

Построение графиков функций

Рассмотрим на примерах способы построения графиков в ML. Если заданы два вектора одинакового размера, хранящие координаты функции, то построить график функции можно с помощью команды **plot**.

Пример 1. Построение графика функции

```
% задание вектора x
x = [0:0.005:5];
% расчет значений функции
y = exp(-x).*sin(10*x);
% построение графика функции
plot(x, y)
```



Графическое окно пакета MatLab

После выполнения скрипта из примера #1 на экране возникнет графическое окно, выполняя команды меню этого графического окна File→Save as, File→Export Setup, можно сохранить построенный график в файл MatLab с расширением fig, либо экспортировать в графические форматы - png, eps, gif. Файл с расширением fig можно открыть в MatLab. Результатом открытия файла будет графическое окно.

Несколько графиков в одном графическом окне

Для того, чтобы построить еще один график в этом же графическом окне, то можно действовать двумя путями – воспользоваться командой hold on, либо добавить еще два аргумента в команду plot. С помощью функции delete производят удаление графика.

Пример 2. Два графика функции в одних осях с помощью hold on

```
x = [0:0.005:5];  
y1 = exp(-x).*sin(10*x);  
y2 = exp(-x).*cos(10*x);  
  
% построение первого графика функции  
plot(x, y1)  
% продолжать построение в этом же окне  
hold on  
% построение второго графика функции  
plot(x, y2)
```

Пример 3. Два графика функции в одних осях с помощью plot

```
x = [0:0.005:5];  
y1 = exp(-x).*sin(10*x);  
y2 = exp(-x).*cos(10*x);  
  
% построение сразу двух графиков функций  
plot(x, y1, x, y2)
```

Установка параметров графиков

Задать цвет и тип линии для графиков можно несколькими способами. Рассмотрим один из этих способов на примере.

Пример 4. Задание цвета и типа линии для графика

```
x = [0:0.005:5];  
y = exp(-x).*sin(10*x);  
plot(x, y, 'r:')
```

Обозначения для типа линий, цветов и маркеров приведены в таблице ниже.

Цвет	
Y	Желтый
M	Розовый
C	Голубой
R	Красный
G	Зеленый
B	Синий
W	Белый
K	Черный
Линия	
-	Сплошная
:	Пунктирная
-.	штрих-пунктирная
--	Штриховая

Маркер	
.	Точка
o	Кружок
x	Крестик
+	знак "плюс"
*	звездочка
s	Квадрат
d	Ромб
v	Треугольник вершиной вниз
^	Треугольник вершиной вверх
<	Треугольник вершиной влево
>	треугольник вершиной вправо
p	пятиконечная звезда
H	шестиконечная звезда

Иногда необходимо создать несколько графических окон одновременно. В этом случае нужно использовать команду **figure** без параметров, или задать в качестве параметра целое число, соответствующее номеру окна.

Заголовок для графического окна задается с помощью функции **title**, ее параметром является текстовая строка.

Если заголовок используется не так часто - обычно его заменяет подписочная подпись, то использование легенды - **legend** всегда целесообразно, когда в одних осях отображается более одного графика.

Легенда позволяет определить взаимное соответствие графиков и функций. Для каждого из графиков в качестве параметров функции **legend** используют текстовую строку, описывающую соответствующий график. Последний необязательный аргумент определяет положение легенды в графическом окне. Также положение легенды можно менять уже после построения графика с помощью перетаскивания.

Расположение легенды	
-1	вне графика в правом верхнем углу графического окна
0	лучшее положение в пределах графика так, чтобы как можно меньше перекрывать сами графики
1	в верхнем правом углу графика (это положение используется по умолчанию)
2	в верхнем левом углу графика
3	в нижнем левом углу графика
4	в нижнем правом углу графика

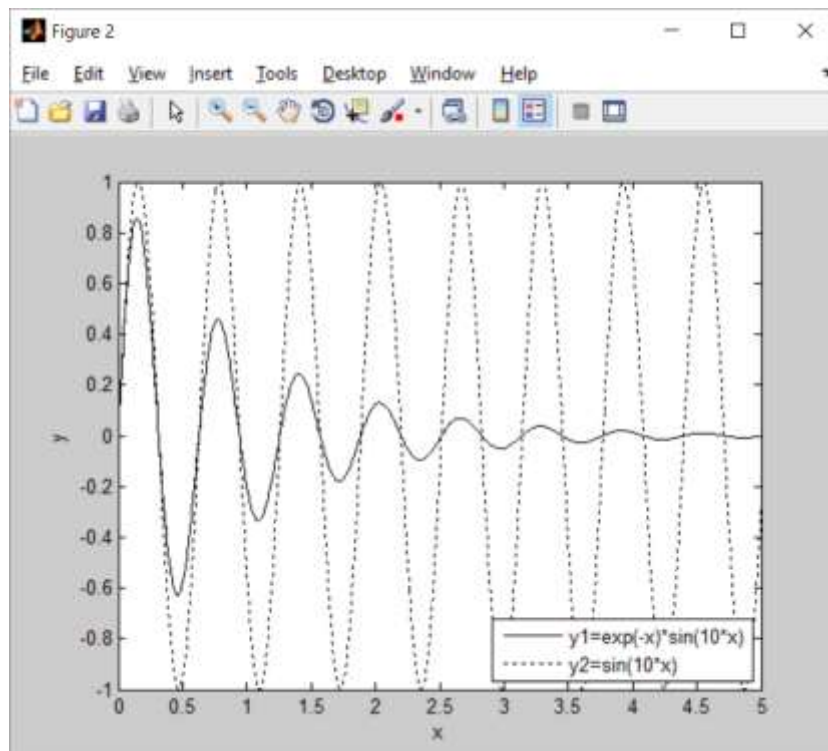
Поименование осей осуществляется посредством функций **xlabel** и **ylabel**. Обратите внимание на то, что дополнительные параметры для графических осей устанавливаются, когда оси уже созданы, то есть после команды **plot**.

Пример 5. Использование легенды и подписи осей

```
% создание графического окна
figure
x = [0:0.005:5];
y1 = exp(-x).*sin(10*x);
y2 = sin(10*x);
% построение графиков двух функций
plot(x, y1, 'k-', x, y2, 'k:')
```

```
% создание легенды
legend('y1=exp(-x)*sin(10*x)', 'y2=sin(10*x)', 4)

% создание подписей к осям
xlabel('x')
ylabel('y')
```



Использование легенды и подписи осей.

Построение графика неявно заданной функции

Кроме рассмотренной функции `plot` есть и другие, выполняющие построение графиков. Например, с помощью **`ezplot`** можно построить график, как явной функции, так и неявно заданной функции. По умолчанию функция **`ezplot`** выполняет построение графика на отрезке $[-2\pi, 2\pi]$. В качестве первого аргумента этой функции необходимо задать текстовую строку – выражение для функции. Вторым аргументом может быть отрезок, на котором выполняется построение функции.

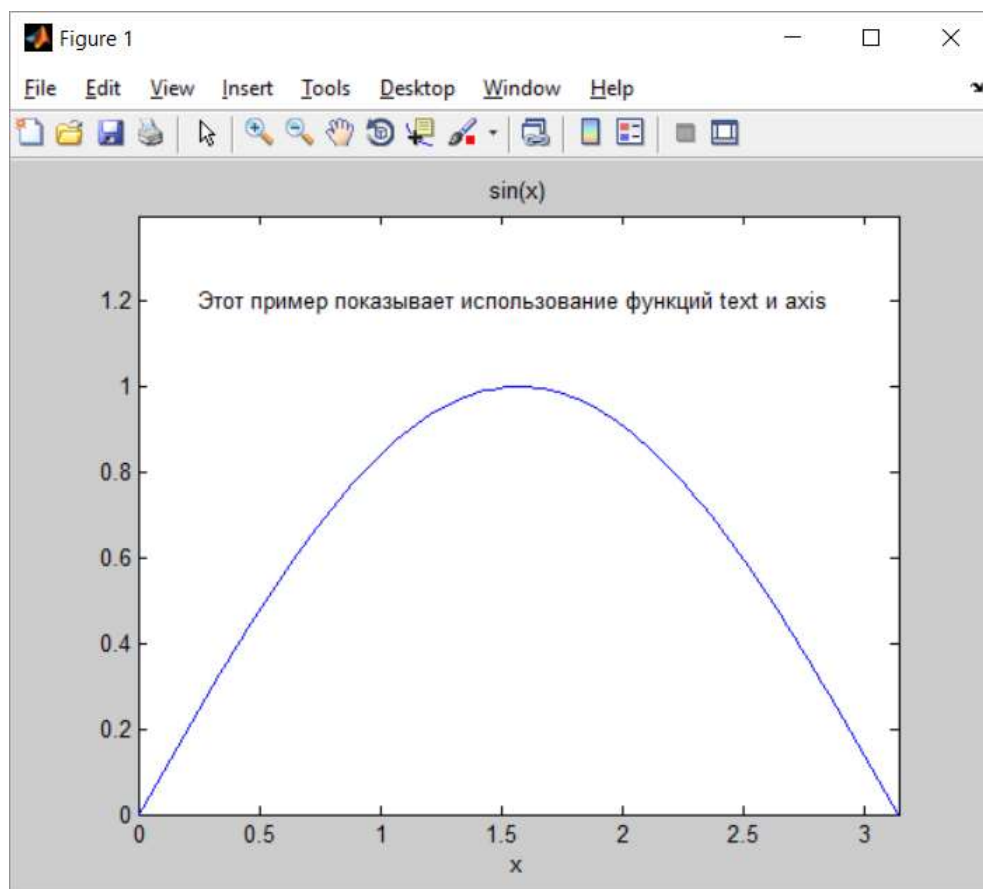
Пример 6. Использование функции ezplot

```
% построение графика сердечной функции, заданной неявно  
ezplot('x^2+(y-abs(x)^(1/2))^2=1')  
% построение графика функции, заданной явно  
% с заданием отрезка, на котором выполняется построение  
ezplot('sin(x)', [0 pi])
```

Задать максимальное и минимальное значение для осей можно с помощью функции

axis([xmin xmax, ymin ymax])

Это бывает необходимо, когда нужно расширить область осей, например, для расположения поясняющего текста. В примере 7 можно увидеть использование функций **axis** и **text**. Последняя Функция позволяет поместить в графические оси текстовую надпись. Ее аргументами являются координаты начала размещения текстовой строки и сама надпись.



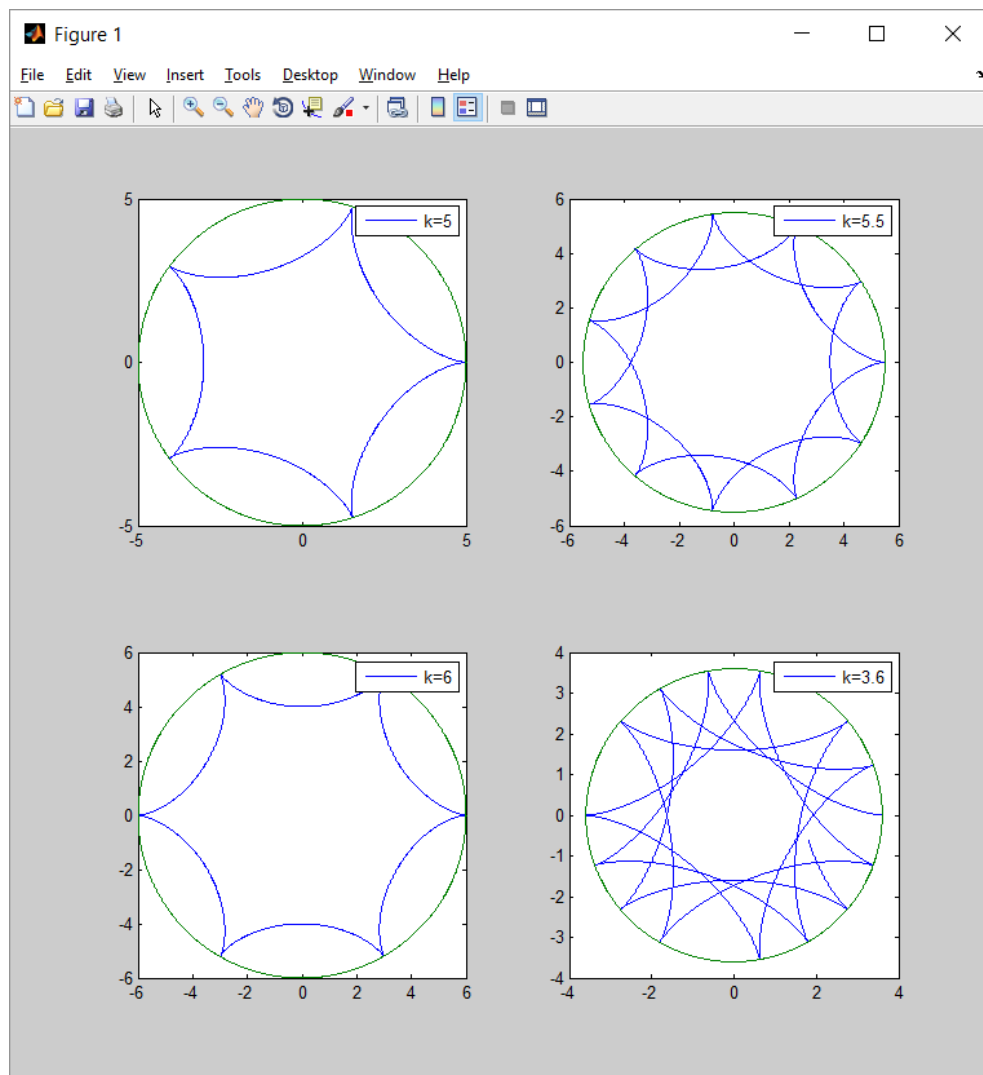
Использование функций axis и text.

Пример 7. Использование функций axis и text

```
ezplot('sin(x)')  
axis([0 pi 0 1.4])  
text(0.25, 1.2,...  
'Этот пример показывает использование функций text и  
axis')
```

Создание нескольких графических окон

Пример #8 соответствует случаю, когда в одном окне **figure** размещается несколько графических окон для визуализации графиков функций (а, вообще говоря, любой графической информации).



Использование функции subplot.

Структура таких окон и выбор активного окна осуществляются процедурой **subplot**. Первый и второй аргументы **subplot** задают матричную структуру окон-осей, их количество по строкам и столбцам, а третий аргумент – номер активных осей. Здесь выбираются графики функций, заданных параметрически.

Пример 8. Использование функции subplot

```
% Гипоциклоиды
clear, clc
% Задание вектора-параметра t
t=0:0.001:8*pi;
% Данные и вектора для 1-го окна subplot
k=5;
x11=(k-1)*(cos(t)+cos((k-1)*t)/(k-1));
y11=(k-1)*(sin(t)-sin((k-1)*t)/(k-1));
x12=k*cos(t);
y12=k*sin(t);
% Данные и вектора для 2-го окна subplot
k=5.5;
x21=(k-1)*(cos(t)+cos((k-1)*t)/(k-1));
y21=(k-1)*(sin(t)-sin((k-1)*t)/(k-1));
x22=k*cos(t);
y22=k*sin(t);
% Данные и вектора для 3-го окна subplot
k=6;
x31=(k-1)*(cos(t)+cos((k-1)*t)/(k-1));
y31=(k-1)*(sin(t)-sin((k-1)*t)/(k-1));
x32=k*cos(t);
y32=k*sin(t);
% Данные и вектора для 4-го окна subplot
k=3.6;
x41=(k-1)*(cos(t)+cos((k-1)*t)/(k-1));
y41=(k-1)*(sin(t)-sin((k-1)*t)/(k-1));
x42=k*cos(t);
y42=k*sin(t);

% Построение графиков в 1-м окне subplot
```



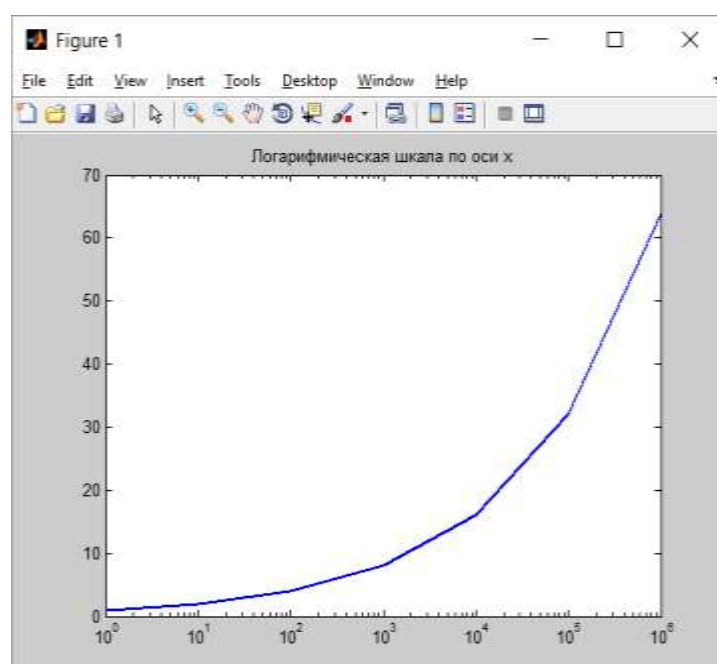
```

subplot(2,2,1)
plot(x11,y11,x12,y12)
legend('k=5')
% Построение графиков во 2-м окне subplot
subplot(2,2,2)
plot(x21,y21,x22,y22)
legend('k=5.5')
% Построение графиков в 3-м окне subplot
subplot(2,2,3)
plot(x31,y31,x32,y32)
legend('k=6')
% Построение графиков в 4-м окне subplot
subplot(2,2,4)
plot(x41,y41,x42,y42)
legend('k=3.6')

```

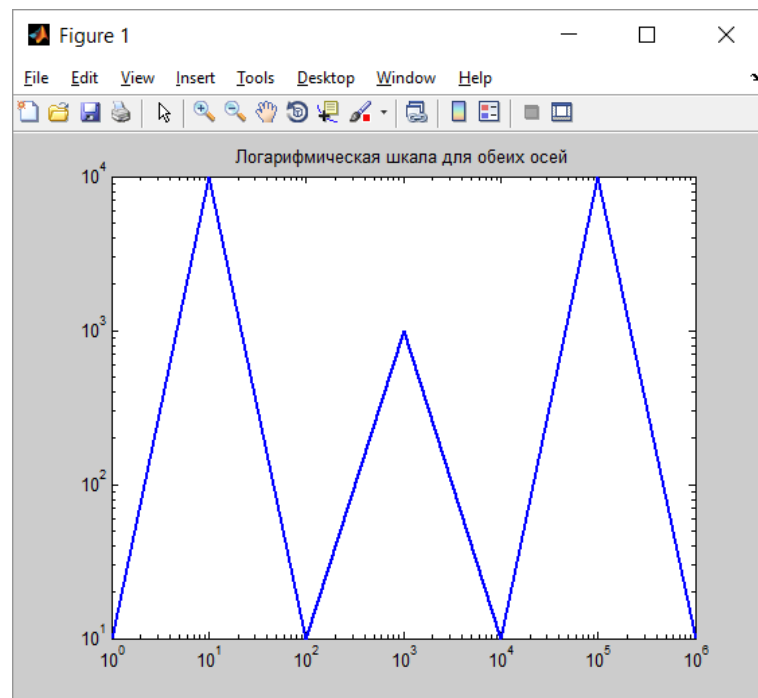
Использование логарифмической шкалы

Следующие примеры охватывают случаи, когда для одной или обеих осей стоит выбрать логарифмический масштаб, для больших диапазонов значений или для зависимостей логарифмического типа. Функции, которые строят графики в логарифмическом масштабе, это: **semilogx**, **semilogy** и **loglog**.



Пример 9. Использование логарифмической шкалы для оси x

```
clear, clc
x=[1e+0,1e+1,1e+2,1e+3,1e+4,1e+5,1e+6]
y=[1,2,4,8,16,32,64];
p=semilogx(x,y), set(p,'LineWidth',2);
title('Логарифмическая шкала по оси x')
```



Пример 10. Использование логарифмической шкалы для обеих осей

```
clear, clc
x=[1e+0,1e+1,1e+2,1e+3,1e+4,1e+5,1e+6]
y=[1e+1,1e+4,1e+1,1e+3,1e+1,1e+4,1e+1];
p=loglog(x,y)
set(p,'LineWidth',2);
title('Логарифмическая шкала для обеих осей')
grid on % задание сетки
```

3. Постройте графики функций:

$$y_1 = \sin(x)$$

$$y_2 = \cos(x)$$

Подпишите оси. Добавьте сетку.

6. Постройте график функции $y = \sin(x)$. Добавьте подписи координатных осей, сетку, команду задания границ для осей. Добавьте подпись ' $\leftarrow \sin(x)$ ' в точке (3.05,0.16).

8. Постройте графики функций

$$f = \log(0.5 * x)$$

$$g = \sin(\log(x))$$

в логарифмическом масштабе по оси x . Добавьте легенду на графики.

11. Постройте в полярных координатах следующие кривые: окружность, спираль Архимеда, сердце, бабочка, полярная роза.

15. Задайте три вектора. Первый вектор содержит номера дней месяца. Второй и третий список содержат значения курса акций в эти дни, для двух разных месяцев. Построить графики зависимости курса акций от номера дня. Добавить заголовок, подписи осей, сетку и легенду, цвета для линий графиков.

17. Постройте график зависимости времени возведения матрицы в квадрат от ее порядка. Матрицу наполнять случайными целыми числами в диапазоне от 1 до 10. Построить такие же графики для верхней и нижней треугольных матриц. Все графики строить в одних осях. Добавить сетку и легенду. Сделать вывод.