

Построение графиков функции

Графические возможности системы MATLAB являются мощными и разнообразными. Изучим наиболее простые в использовании возможности (высокоуровневую графику).

Сформируйте два вектора x и y :

```
>> x=0:0.01:2;    y=sin(x);
```

Вызовите функцию: `>> plot(x,y)`

и вы получите на экране график функции (рис. 1).

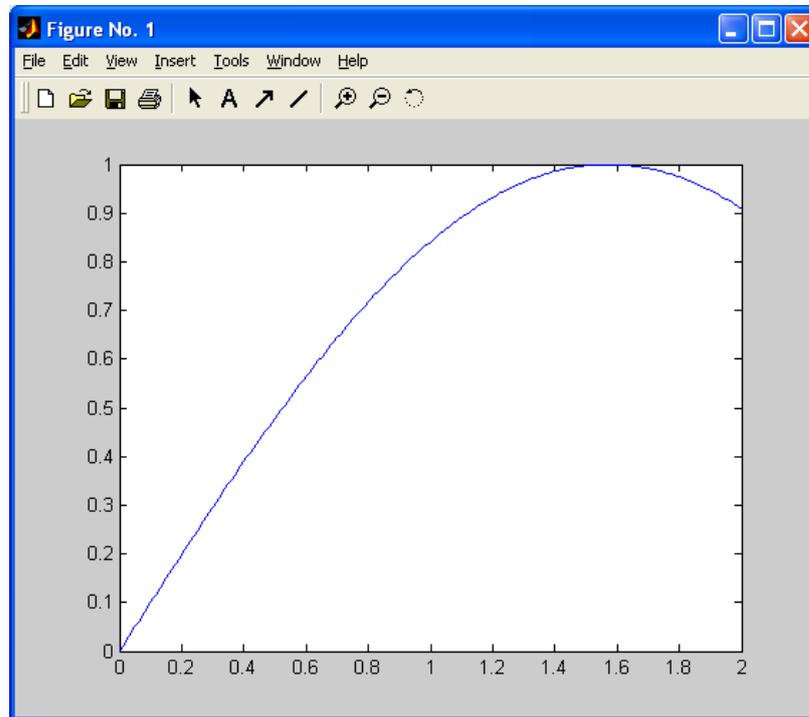


Рисунок 1 – График функции $y=\sin(x)$

MATLAB показывает графические объекты в специальных графических окнах, имеющих в заголовке слово *Figure*.

Не убирая с экрана дисплея первое графическое окно, введите с клавиатуры выражения

```
>> z=cos(x);
```

```
>> plot(x,z)
```

и получите новый график функции в том же самом графическом окне (при этом старые оси координат и график пропадают - этого также можно добиться командой `clf`; командой `cla` удаляют только график с приведением осей координат к их стандартным диапазонам от 0 до 1).

Если нужно второй график провести «поверх первого графика», то перед вторичным вызовом графической функции `plot` нужно выполнить команду `hold on`, которая предназначена для удержания текущего графического окна:

```
>> x=0:0.01:2;    y=sin(x);
```

```
>> plot(x,y)
```

```
>> z=cos(x);
```

```
>> hold on
```

```
>> plot(x,z)
```

Практически тоже самое получится (рис. 2), если набрать:

```
>> x=0:0.01:2;    y=sin(x);    z=cos(x);
```

```
>> plot(x,y,x,z)
```

Но в этом случае графики получаются разного цвета.

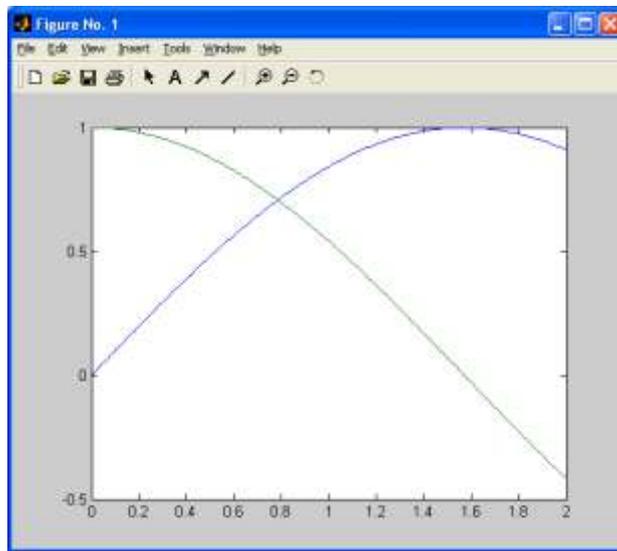


Рисунок 2 – Графики функций $y=\sin(x)$, $z=\cos(x)$, построенные в одном графическом окне

Если нужно одновременно визуализировать несколько графиков так, чтобы они не мешали друг другу, то это можно сделать двумя способами. Первым решением является построение их в разных графических окнах. Для этого перед вторичным вызовом функции `plot` следует набрать команду `figure`, которая создает новое графическое окно и заставляет все последующие за ней функции построения графиков выводить туда. Попробуйте построить $y=\sin(x)$, $z=\cos(x)$ в разных окнах.

Вторым решением показа нескольких графиков без конфликта диапазонов осей координат является использование функции `subplot`. Эта функция позволяет разбить область вывода графической информации на несколько подобластей, в каждую из которых можно вывести графики различных функций.

Например, для ранее выполненных вычислений с функциями `sin` и `cos` постройте графики этих двух функций в первой подобласти, а график функции $\exp(x)$ - во второй подобласти одного и того же графического окна (рис. 3):

```
>> w=exp(x);
>> subplot(1,2,1); plot(x,y,x,z)
>> subplot(1,2,2); plot(x,w)
```

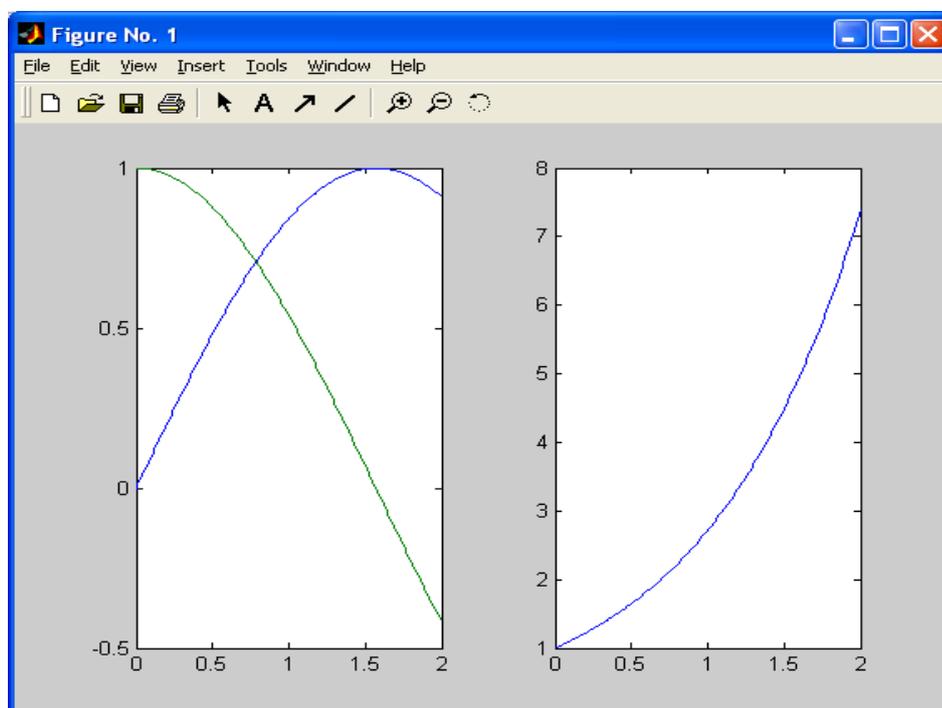


Рисунок 3 – Графики функций $y=\sin(x)$, $z=\cos(x)$ и $w=\exp(x)$, построенные в двух подобластях одного графического окна

Диапазоны изменения переменных на осях координат этих подобластей независимы друг от друга. Функция `subplot` принимает три числовых аргумента, первый из которых равен числу рядов подобластей, второй равен числу колонок подобластей, а третий аргумент - номеру подобласти (номер отсчитывается вдоль рядов с переходом на новый ряд по исчерпанию). Снять действие функции `subplot` можно командой:

```
>> subplot(1,1,1)
```

Если для одиночного графика диапазоны изменения переменных вдоль одной или обеих осей координат слишком велик, то можно воспользоваться функциями построения графиков в *логарифмических масштабах*. Для этого предназначены функции `semilogx`, `semilogy` и `loglog`.

Построить график функции в полярных координатах (рис. 4) можно с помощью графической функции `polar`.

```
>> phi=0:0.01:2*pi; r=sin(3*phi);  
>> polar(phi,r)
```

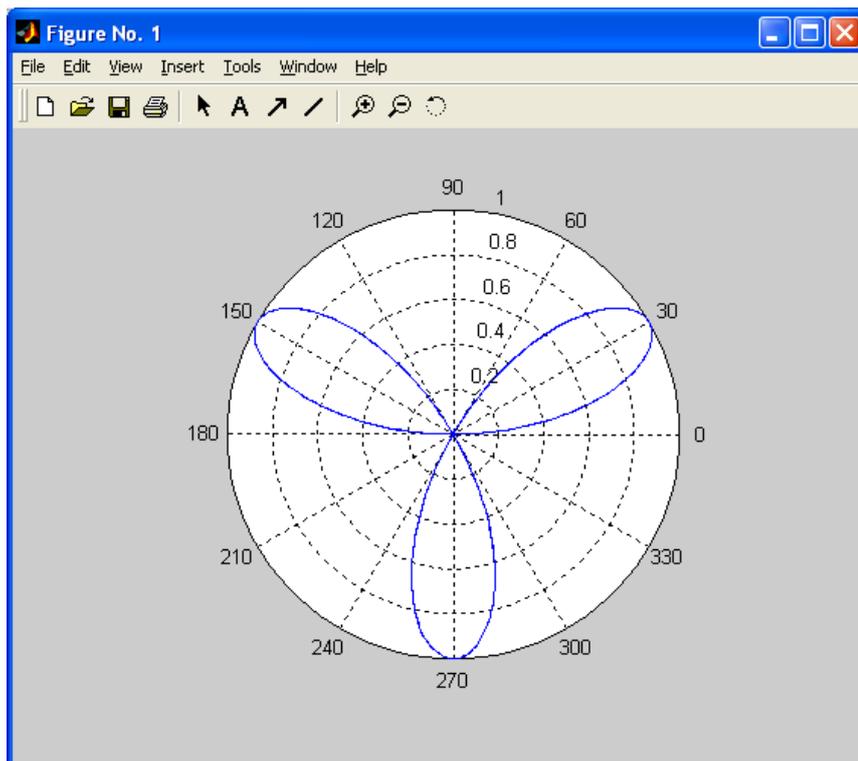


Рисунок 4 – График функции $r=\sin(3*\phi)$ в полярных координатах

Рассмотрим дополнительные возможности, связанные с управлением внешним видом графиков - задание цвета и стиля линий, а также размещение различных надписей в пределах графического окна. Например, команды

```
>> x=0:0.1:3; y=sin(x);  
>> plot(x,y,'r-',x,y,'ko')
```

позволяют придать графику вид красной сплошной линии (рис. 5), на которой в дискретных вычисляемых точках проставляют черные окружности. Здесь функция `plot` дважды строит график одной и той же функции, но в двух разных стилях. Первый из этих стилей отмечен как `'r-'`, что означает проведение линии красным цветом (буква `r`), а штрих означает проведение сплошной линии. Второй стиль, помечен как `'ко'`, означает проведение черным цветом (буква `k`) окружностей (буква `o`) на месте вычисляемых точек.

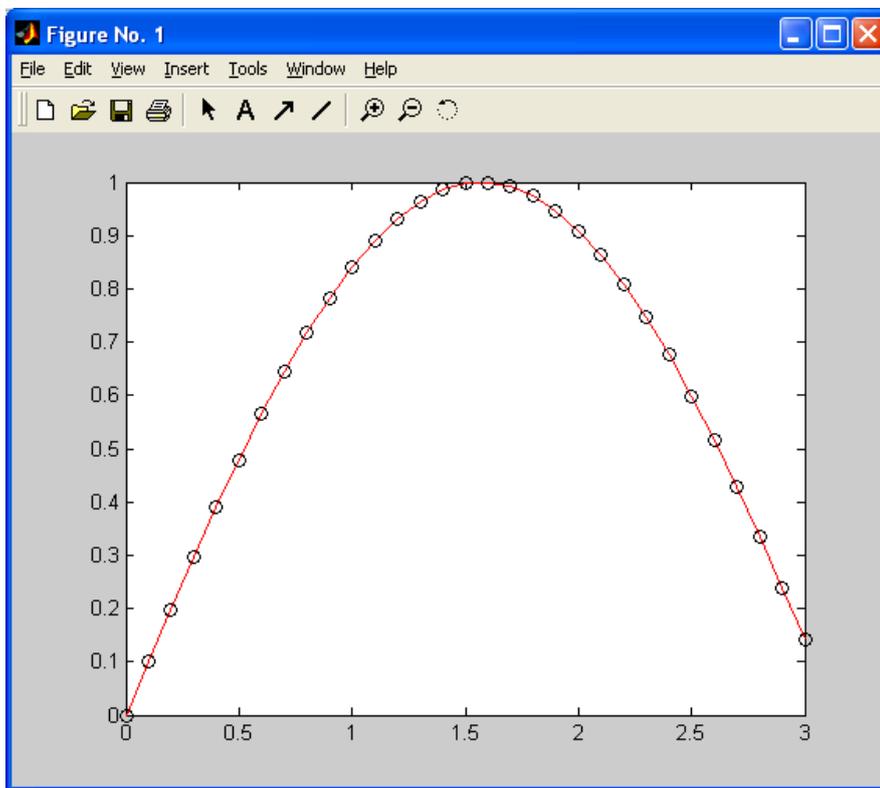


Рисунок 5 – Построение графика функции $y=\sin(x)$ в двух разных стилях

В общем случае функция `plot(x1, y1, s1, x2, y2, s2, ...)` позволяет объединить в одном графическом окне несколько графиков функций $y_1(x_1)$, $y_2(x_2)$, ... проведя их со стилями s_1 , s_2 , ... и т. д.

Стили s_1, s_2, \dots задаются в виде набора трех *символьных маркеров*, заключенных в одиночные кавычки (апострофы). Один из этих маркеров задает тип линии (Таблица 3). Другой маркер задает цвет (Таблица 4). Последний маркер задает тип проставляемых «точек» (Таблица 5). Можно указывать **не все** три маркера. Тогда используются маркеры, установленные по умолчанию. Порядок, в котором указывают маркеры, не является существенным, то есть 'r+-' и '-+r' приводит к одинаковому результату.

Таблица 3 - Маркеры, задающие тип линии

Маркер	-	--	:	-.
Тип линии	Непрерывная	Штриховая	Пунктирная	Штрихпунктирная

Таблица 4 - Маркеры, задающие цвет линии

Маркер	Цвет линии	Маркер	Цвет линии
c	Голубой	g	Зеленый
m	Фиолетовый	b	Синий
y	Желтый	w	Белый
r	Красный	k	Черный

Таблица 5 - Маркеры, задающие тип точки

Маркер	.	+	*	o	x
Тип точки	Точка	Плюс	Звездочка	Кружок	Крестик

Если в строке стиля поставить маркер на тип точки, но не проставить маркер на тип линии, то тогда отображаются только вычисляемые точки, а непрерывной линией они не соединяются.

Теперь перейдем к **оформлению осей координат, к надписям на осях**. Система MATLAB устанавливает пределы на горизонтальной оси равными тем значениям, что указаны пользователем для независимой переменной. Для зависимой переменной по вертикальной оси MATLAB самостоятельно вычисляет диапазон изменения значений функции. Если мы хотим отказаться от этой особенности масштабирования при построении графиков в системе MATLAB, то мы должны явным образом навязать свои пределы изменения переменных по осям координат. Это делается с помощью функции

```
axis ([ xmin, xmax, ymin, ymax ])
```

Для проставления различных надписей на полученном рисунке применяют функции `xlabel`, `ylabel`, `title` и `text`. Функция `xlabel` создает подпись у горизонтальной оси, функция `ylabel` - тоже для вертикальной оси (причем эти надписи ориентированы вдоль осей координат). Если требуется разместить надпись в произвольном месте рисунка, применяем функцию `text`. Общий заголовок для графика создается функцией `title`. Кроме того, используя команду `grid on`, можно нанести измерительную сетку на всю область построения графика. Например (рис. 6):

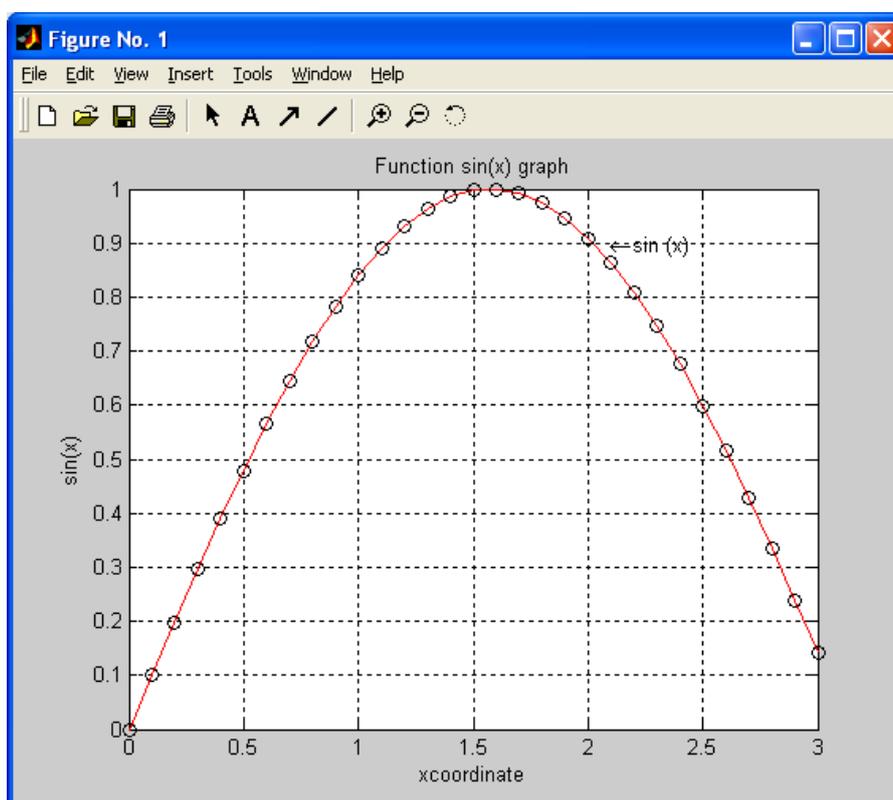


Рисунок 6 – График функции $y=\sin(x)$, построенный в двух стилях с надписями на координатных осях и на рисунке

```
>> x=0:0.1:3; y=sin(x);  
>> plot(x,y,'r-',x,y,'ko')  
>> title('Function sin(x) graph')  
>> xlabel('xcoordinate'); ylabel('sin(x)')  
>> text(2.1,0.9,'\leftarrow sin (x)'); grid on
```

Надпись функцией `text` помещается начиная от точки с координатами, указанными первыми двумя аргументами. По умолчанию координаты задаются в тех же единицах измерения, что и координаты, указанные на горизонтальной и вертикальной осях. Специальные управляющие символы вводятся внутри текста после символа `\` (обратная косая черта).