Компьютерные технологии в научных исследованиях и образовании

Юрий Анатольевич Богоявленский, заведующий кафедрой Информатики и математического обеспечения, к.т.н., доцент, ybgv

Построение графиков

Введение

Инструменты высокого уровня

Двумерные графики

Функция plot

Входные данные

Управление свойствами линий и легендами

Функция `plotyy` - две оси ординат

Перечень различных функций

Функция fplot — неявное задание данных

Изначально графики в `octave` строились с помощью пакета `gnuplot`, сейчас же, на базе `API OpenGL`, реализована еще поддержка инструментов `FLTK` и `Qt`. Вызов функции `graphics_toolkit` без параметров выводит информацию о текущем установленном инструменте, вызовы `graphics_toolkit ("gnuplot")`, `graphics_toolkit ("qt")` и `graphics_toolkit ("fltk")` включают соответствующий инструмент. Каждый график содержит данные об инструменте, с помощью которого он построен.

Предупреждение. Инструменты на основе `OpenGL` используют переменные с одинарной точностью точностью, которые ограничивают максимальное отображаемое значение величиной примерно 10^{38}. Если данные содержат большие значения, нужно использовать `gnuplot`, который поддерживает значения до 10^{308}. Аналогично, переменные одинарной

точности могут точно представлять только 6-9 десятичных цифр. Кроме того, если значение отличаются менее чем на 10^{-8}, они буду неотличимы для инструментов на базе OpenGL и нужно использовать gnuplot.

Примечание. Связь между octave и gnuplot осуществляется по однонаправленному конвейеру (pipe), что существенно снижает производительность и функциональные возможности.

Производительность значительно снижается, поскольку весь набор данных, который может составлять много мегабайт, должен быть передан в gnuplot по конвейеру. Функциональные возможности снижены т. к. у конвейера только одно направление — от octave к gnuplot, т.е. нет возможности передать в octave информацию о взаимодействии пользователя с окном графика (будь то изменение размера, перемещение, закрытие или что-либо еще). Рекомендуется не взаимодействовать с окном gnuplot, или закрывать его.

Отметим, что график в octave строится из следующих примитивных объектов figure (фигура), axes (оси), line (линия), text (текст), patch (накладка), scatter (рассеивание), surface (поверхность), text (текст), image (изображение) и light (свет). На каждый из них можно ссылаться с помощью дескриптора (handle), определяемого при создании объекта.

Замечание. В текущей версии (7.1.0) объект patch (накладка) определяется как закрашенный двумерный многоугольник.

Характеризации этих объектов и функции работы с ними даны в [Док, с. 419 — 508] и не будут рассмотрены нами а деталях. Ниже мы представим упрощенные инструменты высокого уровня, достаточные для решения многих практических задач построения графиков. В то же время, при описании этих инструментов эпизодически буду встречаться упоминания этих примитивных объектов и их свойств.

Инструменты высокого уровня

Octave предоставляет простые инструменты для создания множества различных типов двух и трех мерных графиков на основе функций с высоким уровнем абстракции от деталей машинной графики. В [Док, с. 489 — 508] представлены функции обеспечивающие детальное управление графиками.

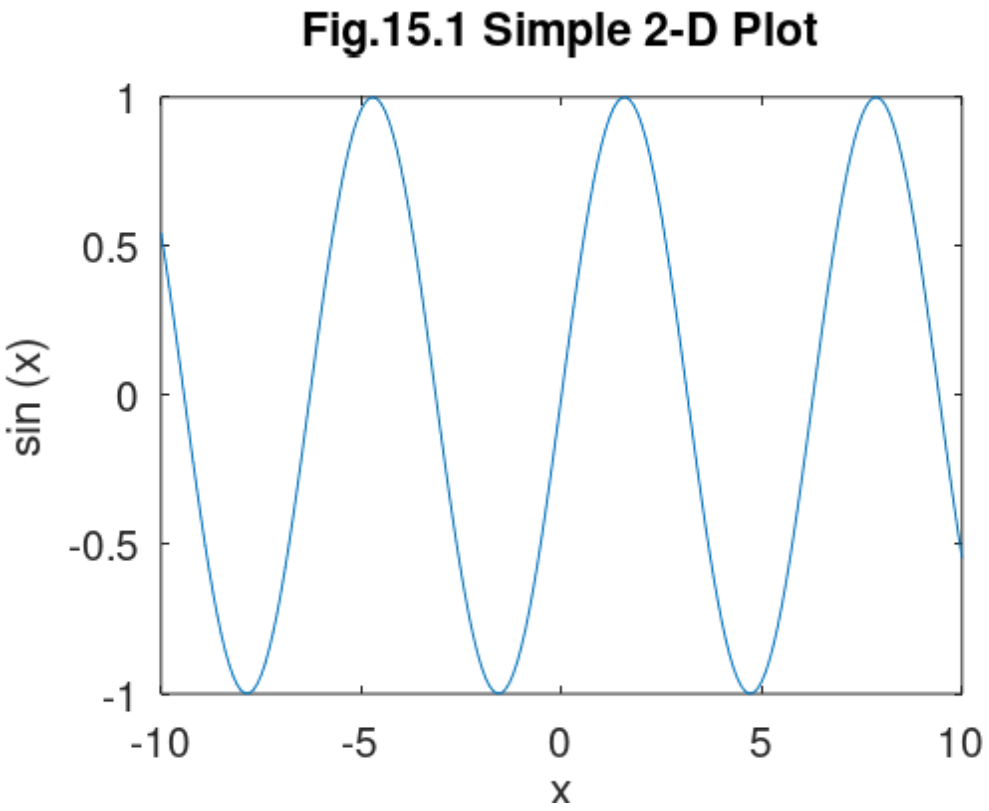
Функция plot

Функция plot позволяет создавать простые графики с линейными осями.

Например программа

```
x = -10:0.1:10;
plot (x, sin (x));
xlabel ("x");
ylabel ("sin (x)");
title ("Simple 2-D Plot");
```

построит график



который, в большинстве систем, откроется в отдельном окне.

Функцию plot, строящуя двухмерные графики, можно задавать в следующих формах.

```
plot (y)
plot (x, y)
plot (x, y, fmt)
plot ( . . . , property, value, . . . )
plot (x1, y1, . . . , xn, yn)
plot (hax, . . . )
h = plot ( . . . )
```

Входные данные

Возможно много различных комбинаций фактических параметров. Простейшей формой является:

```
plot (y)
```

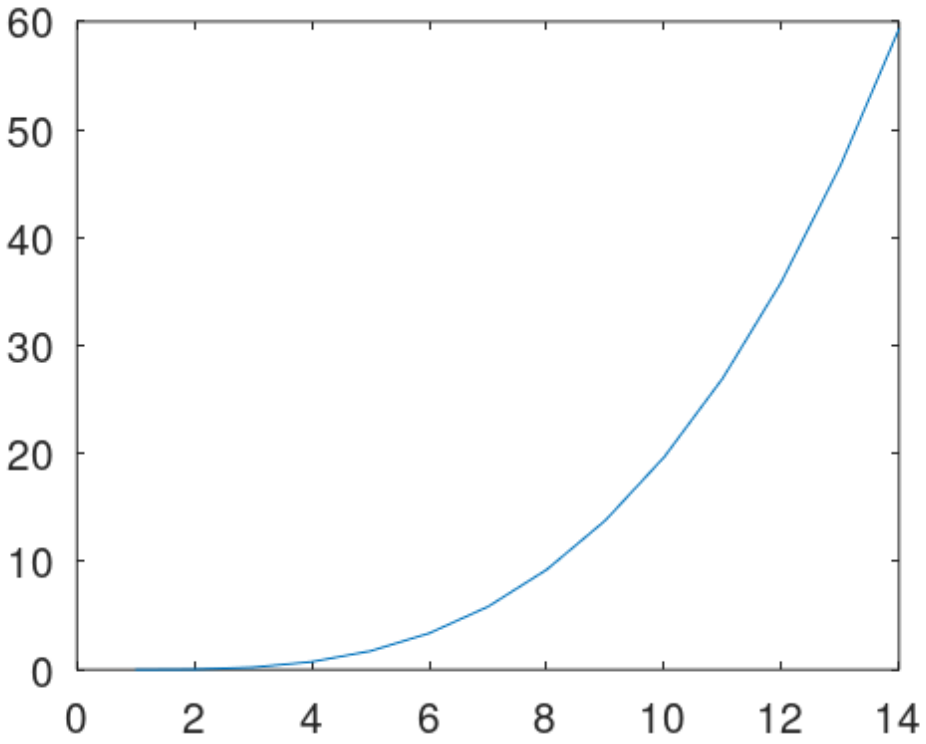
когда по оси ординат выводятся значения элементов y, а соответствующие им значения по оси абсцисс берутся из диапазона 1:numel (y), где функция numel возвращает количество элементов своего параметра.

Пример. Программа

```
ybgv@ybgv-home:~/MyOct> octave  -q -p ~/MyOct
octave:1>
octave:1> cd ~/MyOct/Plotting
octave:2> pwd
ans = /home/ybgv/MyOct/Plotting
octave:3>
octave:3> x=0:0.3:4
x =

 Columns 1 through 7:

         0    0.3000    0.6000    0.9000    1.2000    1.5000
1.8000

 Columns 8 through 14:

    2.1000    2.4000    2.7000    3.0000    3.3000    3.6000
3.9000

octave:4> Y=[x.^3]'
Y =

         0
    0.0270
    0.2160
    0.7290
    1.7280
```

```
     3.3750
     5.8320
     9.2610
    13.8240
    19.6830
    27.0000
    35.9370
    46.6560
    59.3190

octave:5> numel(Y)
ans = 14
octave:6> plot(Y)
octave:7> print figure1.png
octave:8>
```

дает график:



Если в вызове `plot` задано более одного фактического параметра, они могут интерпретироваться как:

- `plot (y, property, value, ...)`
- `plot (x, y, property, value, ...)`
- `plot (x, y, fmt, ...)`

и т.п. Может быть задано любое количество фактических параметров.

Значения `x` и `y` интерпретируются следующим образом.

- Если задан единственный параметр данных, он считается набором значение ординат, а за значения абсцисс принимаются индексы элементов, начиная с 1, см. пример выше.
- Если `x` и `y` скаляры, то на графике отображается одна точка.
- Если `x` и `y` векторы, то значения элементов `y` считаются ординатами точек графика, а значения элеменов `x` — их абсциссами. Количества элементов `x` и `y` должны совпадать.
- Если `x` вектор, а `y` матрица, то для каждого ее столбца (строки) строится отдельный график. При этом значения элементов столбцов (или строк) считаются значениями ординат точек графика, а значения элементов `x` — их абсциссами. Количества элементов в `x` и в стобцах (или строках) `y` должны совпадать. Выбор между строками и столбцами `y` происходит автоматически.
- Если `x` матрица, а `y` вектор, то для каждого столбца (строки) `x` строится отдельный график. При этом значения элементов столбцов (или строк) считаются значениями абсцисс точек графика, а значения элементов `x` — их ординатами. Количества элементов в `y` и в столбцах (или строках) `x` должны совпадать. Выбор между строками и столбцами `x` происходит автоматически.
- Если `x` и `y` матрицы, то для каждой пары их соответствующих столбцов строится отдельный график. При этом значения элементов столбцов `y` считаются значениями ординат точек графика, а значения элементов

соответствующих столбцов x — их абсциссами. Обе матрицы должны иметь одинаковое количество строк и столбцов.

Пример программы, строящей графики для описанных комбинаций фактических параметров данных в виде векторов и матриц.

```
ybgv@ybgv-home:~> octave  -q -p ~/MyOct -p ~/MyOct/Plotting/
octave:1> Ymatrix
warning: fopen: '/home/ybgv/MyOct/Plotting/Ymatrix.m' found by searching load path
warning: called from
    fileread at line 42 column 7
    Ymatrix at line 3 column 1

ans = # Задание ординат и абсцисс матрицами 2x11 и 11x2

fileread Ymatrix.m
echo on
x=0:10

Y1=ones(2,11)

Y=x.*Y1

Yrow=Y

Yrow(2,:)=Y(1,:).^1.5

Ycol=Yrow'

X=Y'

whos
```

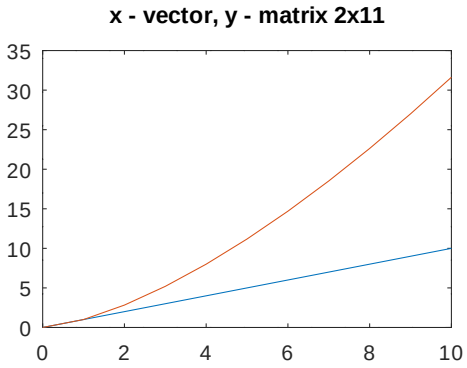
```
cd ~/Teaching-Process/COURSES/Komp-teh-obr-nauka/Lekcii/Plot/

f = figure('position', [10 50 1200 800]);
subplot (3, 2, 1)
plot(x,Yrow);title("x - vector, y - matrix 2x11")
subplot (3, 2, 2)
plot(x,Ycol);title("x - vector, y - matrix 11x2")
subplot (3, 2, 3)
plot(Yrow,x);title("x - matrix 2x11, y - vector")
subplot (3, 2, 4)
plot(Ycol,x);title("x - matrix 11x2, y - vector")
subplot (3, 2, 5)
plot(X,Ycol);title("x - matrix 11x2, y - matrix 11x2")
subplot (3, 2, 6)
plot(Ycol,X);title("x - matrix 11x2, y - matrix 11x2")
print XiY-ViM.pdf
print XiY-ViM.png

# ---------- Конец программы



+ echo on
+ x=0:10
x =

    0    1    2    3    4    5    6    7    8    9   10

+
+ Y1=ones(2,11)
Y1 =
```

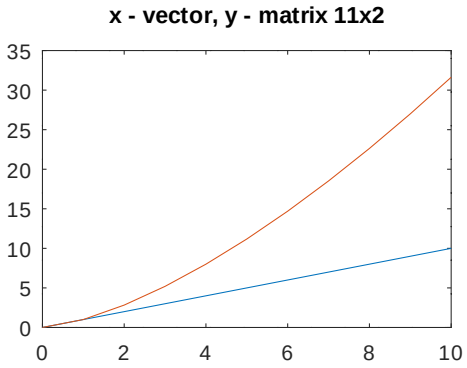
```
     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1

  +
  + Y=x.*Y1
  Y =

      0     1     2     3     4     5     6     7     8     9    10
      0     1     2     3     4     5     6     7     8     9    10

  +
  + Yrow=Y
  Yrow =

      0     1     2     3     4     5     6     7     8     9    10
      0     1     2     3     4     5     6     7     8     9    10

  +
  + Yrow(2,:)=Y(1,:).^1.5
  Yrow =

   Columns 1 through 7:

            0     1.0000     2.0000     3.0000     4.0000     5.0000
6.0000
            0     1.0000     2.8284     5.1962     8.0000    11.1803
14.6969

   Columns 8 through 11:

      7.0000     8.0000     9.0000    10.0000
```

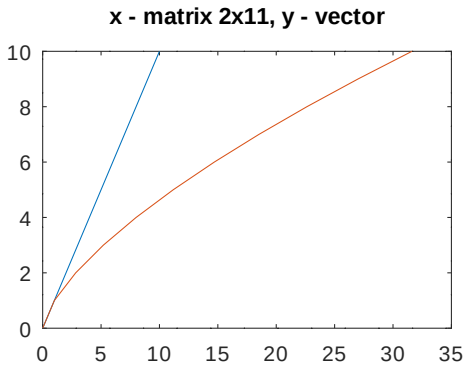
```
   18.5203   22.6274   27.0000   31.6228

+
+ Ycol=Yrow'
Ycol =

         0         0
    1.0000    1.0000
    2.0000    2.8284
    3.0000    5.1962
    4.0000    8.0000
    5.0000   11.1803
    6.0000   14.6969
    7.0000   18.5203
    8.0000   22.6274
    9.0000   27.0000
   10.0000   31.6228

+
+ X=Y'
X =

    0    0
    1    1
    2    2
    3    3
    4    4
    5    5
    6    6
    7    7
    8    8
    9    9
```

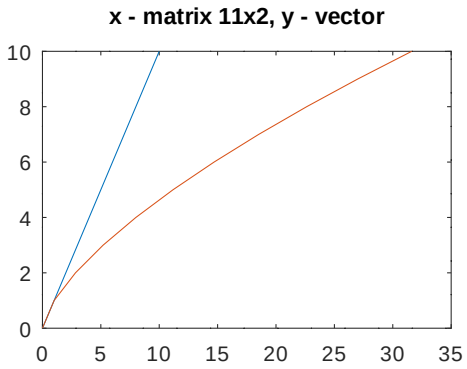
```
   10   10


+
+ whos
Variables visible from the current scope:

variables in scope: top scope

  Attr   Name        Size                     Bytes  Class
  ====   ====        ====                     =====  =====
         X          11x2                       176  double
         Y           2x11                      176  double
         Y1          2x11                      176  double
         Ycol       11x2                       176  double
         Yrow        2x11                      176  double
         ans         1x802                     802  char
         x           1x11                       24  double

Total is 923 elements using 1706 bytes


+
+ cd ~/Teaching-Process/COURSES/Komp-teh-obr-nauka/Lekcii/Plot/
+
+ f = figure('position', [10 50 1200 800]);
+ subplot (3, 2, 1)
+ plot(x,Yrow);title("x - vector, y - matrix 2x11")
+ subplot (3, 2, 2)
+ plot(x,Ycol);title("x - vector, y - matrix 11x2")
+ subplot (3, 2, 3)
+ plot(Yrow,x);title("x - matrix 2x11, y - vector")
+ subplot (3, 2, 4)
+ plot(Ycol,x);title("x - matrix 11x2, y - vector")
```

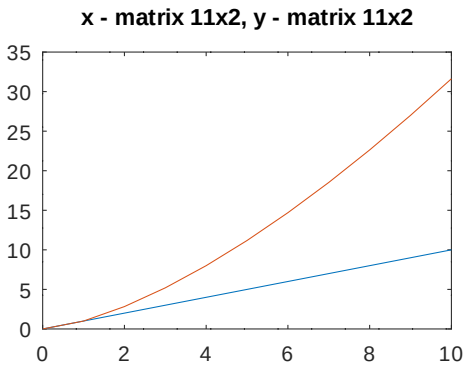
```
+ subplot (3, 2, 5)
+ plot(X,Ycol);title("x - matrix 11x2, y - matrix 11x2")
+ subplot (3, 2, 6)
+ plot(Ycol,X);title("x - matrix 11x2, y - matrix 11x2")
+ print XiY-ViM.pdf
+ print XiY-ViM.png
+
+ # ---------- Конец программы
+
octave:2>
```

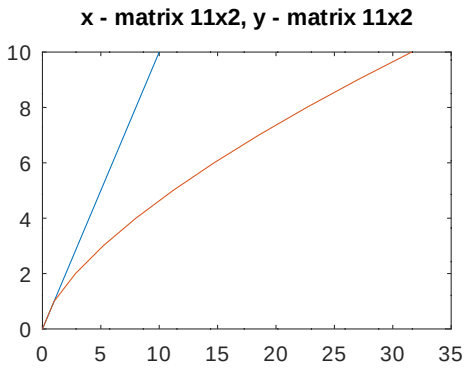












Управление свойствами линий и легендами

В форматах: `plot (y, property, value, …)` и `plot (x, y, property, value, …)` может быть задано несколько пар фактичесих

параметров вида property, value (свойство, значение). Эти параметры описывают свойства отображаемых линий (которые, напомним, в octave рассматриваются как объекты). Это, например, такие свойства как "linestyle", "linewidth", "color", "marker", "markersize", "markeredgecolor", "markerfacecolor". Полный список свойств описан в [Док, с. 448 — 451], в разделе 15.3.3.5. Line properties.

Фактические параметры вида fmt также используются для управления стилем отображения и задаются как текстовая строка, состоящая из четырех необязательных частей:

"<linestyle><marker><color><;displayname;>"

Если присутствует <marker>, но отсутствует <linestyle>, то будут отображаться только маркеры без линий. Аналогично, если присутствует <linestyle>, но отсутствует <marker>, то будут отображаться только линии. Если присутствует оба, то будут отображаться и линии и маркеры. Если не заданы ни параметры вида fmt ни пары property, value (свойство, значение), то по умолчанию будут отображаться сплошные линии без маркеров и цвет, определяемый свойством "colororder" текущих осей.

Возможные значения частей параметра вида fmt:

<linestyle>

Значение	Тип линии
'-'	сплошная (по умолчанию)
'--'	штриховая
':'	пунктирная
'-.'	штрихпунктирная

<marker>

Значение	Название
‘+’	перекрестие
‘o’	кружок
‘*’	звездочка
‘.’	точка
‘x’	крестик
‘|’	вертикальная линия
‘_’	горизонтальная линия
‘s’	квадрат
‘d’	ромб
‘^’	треугольник, обращенный вверх
‘v’	Треугольник , обращенный вниз
‘>’	треугольник, обращенный вправо
‘<’	треугольник, обращенный влево
‘p’	треугольная пентаграмма
‘h’	гексаграмма

<color>

Значение	Название
‘k’	черный
‘r’	красный
‘g’	зеленый
‘b’	синий
‘y’	желтый
‘m’	пурпурный
‘c’	голубой

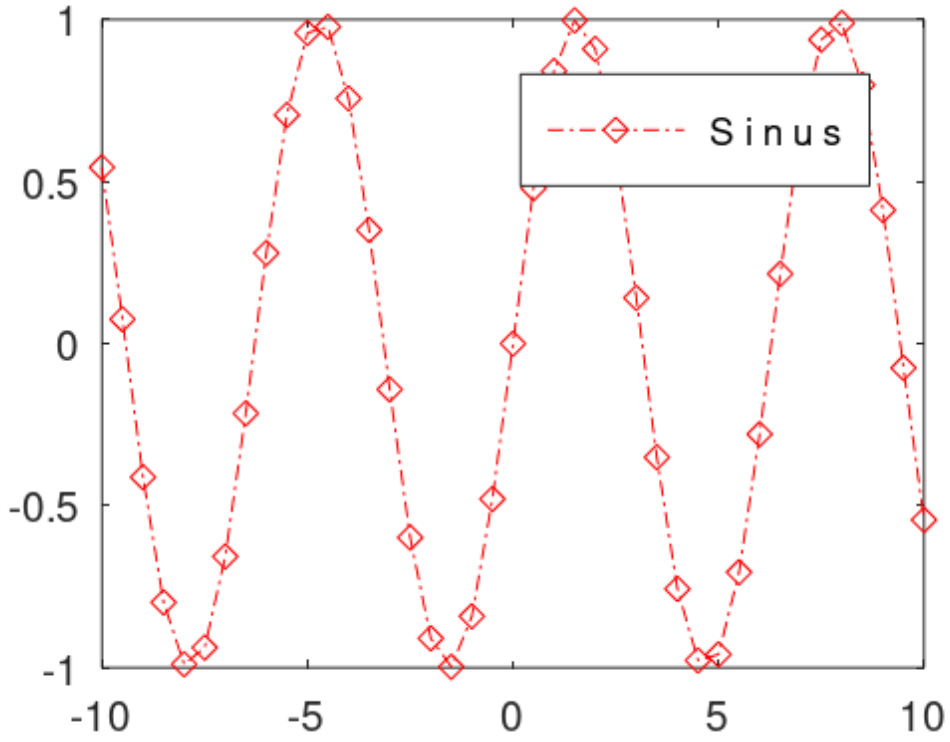
Значение	Название
‘w’	белый

Текст между точками с запятой используется для установки свойства "displayname", значение которго есть текстовая строка, выводимая на графике в рамке легенды соответствующей кривой.

Пример.

```
x=-10:0.5:10
plot (x, sin (x),"-.dr;S i n u s;");print sin.pdf;print sin.png
```

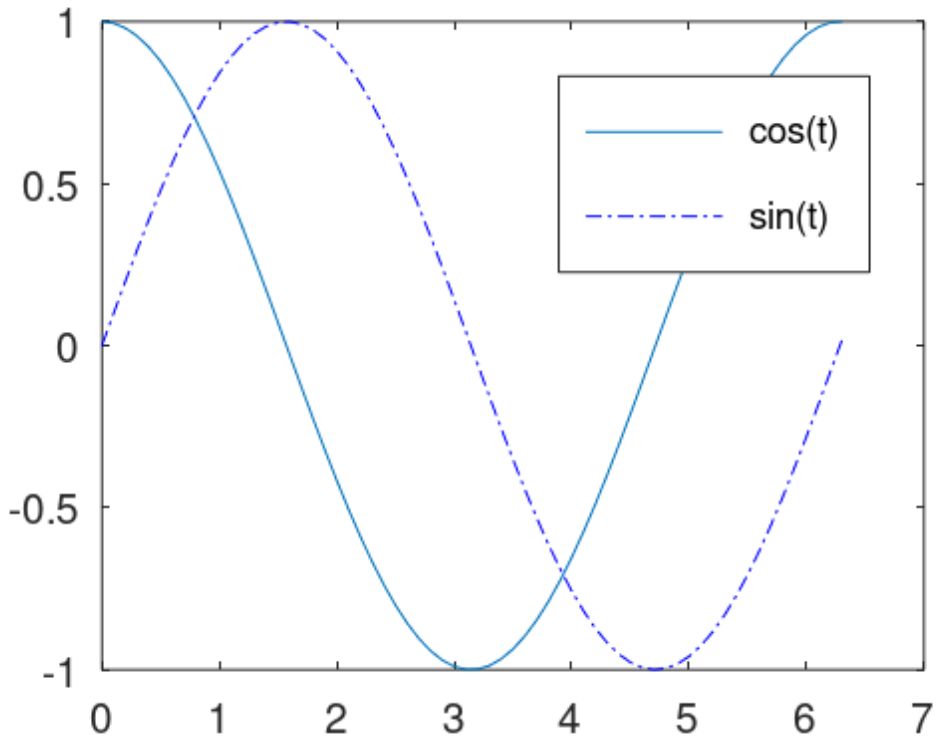
Этот код построит кривую штрихпунктирной линией с маркерами в виде ромбов, красным цветом, с указанием в рамке легенды для этой кривой с показом заданного вида линии и текста: S i n u s. График будет выведен в файлы `sin.pdf` и `sin.png` в текущем каталоге.



При размещении на одном графике нескольких кривых параметр вида `fmt` позволяет вывести легенды для каждой из них, с указанием вида и цвета линии, отображающей кривую.

Пример.

```
echo on
t = 0:0.1:6.3;
plot (t, cos(t), "-;cos(t);",...
      t, sin(t),"-.b;sin(t);")
print Legsincos.pdf; print Legsincos.png
```



В форме вызова `plot (hax, . . . )` первым фактическим параметром является значение дескриптора объекта `axes (оси)`. При этом график будет строиться в осях, определенных этим объектом, а не в определенных текущим объектом `axes (оси)`, ссылка на дескриптор которго задается функцией `gca`.

В форме вызова h = plot (. . .) возвращаемое значение h есть вектор дескрипторов созданных объектов.

Функция plotyy - две оси ординат

Функция построения графика с двумя независимыми осями ординат и общей осью абсцисс plotyy может быть задана в формах:

```
plotyy (x1, y1, x2, y2)
plotyy ( . . . , fun)
plotyy ( . . . , fun1, fun2)
plotyy (hax, . . . )
[ax, h1, h2] = plotyy ( . . . )
```

Параметры x1 и y1 определяют данные для первого графика, а x2 и y2 — для второго. По умолчанию графики строятся с помощью вызова функции plot конструкцией аналогичной feval (@plot, x, y). Для каждого из графиков можно задать собственную функцию его построения: параметром fun1 для данных x1 и y1 и параметром fun2 для данных x2 и y2. При этом графики генерируются конструкцией, аналогичной feval (fun, x, y). Параметры fun1 и fun2 могут быть дескриптором функции, встроенной функцией или строкой, содержащей имя функции.

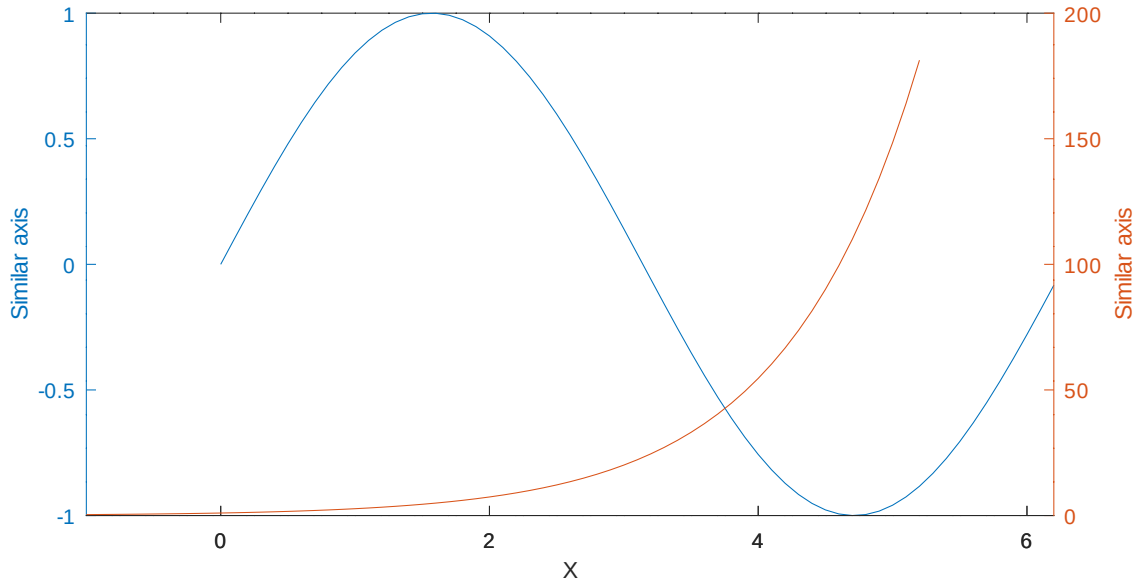
Первым параметром - hax можно задать дескриптор объекта оси для основных осей, по которым выполняется построение данные x1 и y1. Это также может быть двухэлементный вектор с маркерами осей для первичной и вторичной осей.

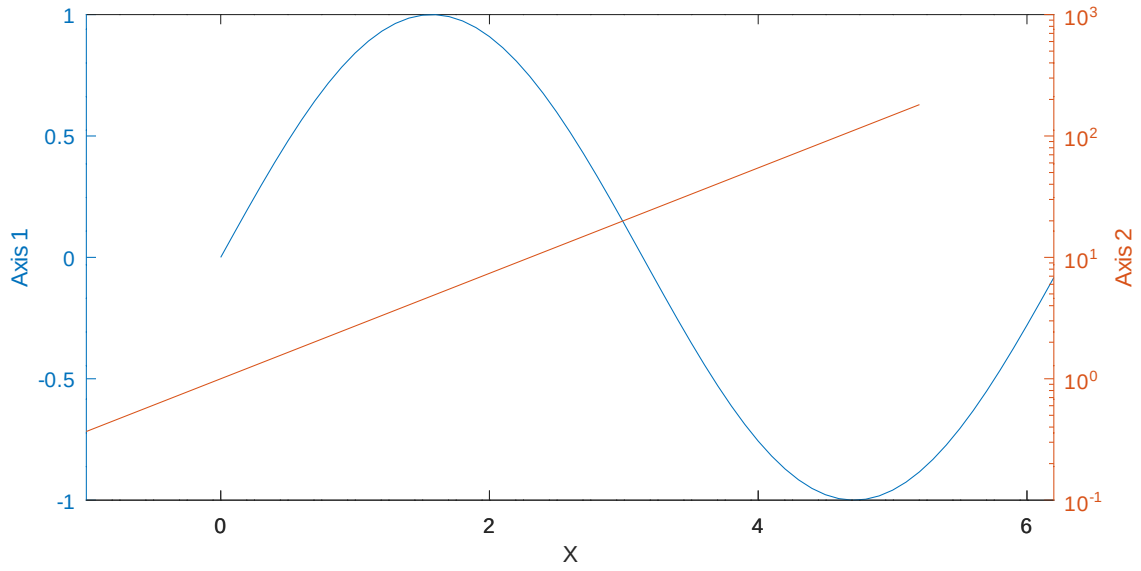
Возвращаемое значение ax представляет собой вектор, содержащий два дескриптора объектов ось для двух осей ординат, сгенерированных командами построения графиков.

В примере ниже вызов ax = plotyy (x, y1, x - 1, y2); по умолчнию строит верхний график рисунка с помощью функции plot с одинаковыми для обоих наборов данных линейными осями. В свою очередь

вызов `ax = plotyy (x, y1, x - 1, y2, @plot, @semilogy);` строит на нижнем графике рисунка данные набора `x, y1` с помощью функции `plot` с линейными осями, а данные набора `x - 1, y2` — с помощью функции `semilogy`, которая во всем аналогична функции `plot`, кроме того, что применяет логарифмическую ось ординат.

```
octave:6> Dvefun
+
+ echo on
+ echo on
+ x = 0:0.1:2*pi;
+ y1 = sin (x);
+ y2 = exp (x - 1);
+
+ f = figure('position', [10 10 1300 1500]);
+
+ subplot(2,1,1)
+ ax = plotyy (x, y1, x - 1, y2);
+ xlabel ("X");
+ ylabel (ax(1), "Similar axis");
+ ylabel (ax(2), "Similar axis");
+
+ subplot(2,1,2)
+ ax = plotyy (x, y1, x - 1, y2, @plot, @semilogy);
+ xlabel ("X");
+ ylabel (ax(1), "Axis 1");
+ ylabel (ax(2), "Axis 2");
+ print 2axes.pdf;print 2axes.png
+
octave:7>
```


Similar axis
Similar axis
X
1
0.5
0
-0.5
-1
200
150
100
50
0
0
2
4
6


Axis 1
Axis 2
X
1
0.5
0
-0.5
-1
10^3
10^2
10^1
10^0
10^{-1}
0
2
4
6

Перечень различных функций

Далее в [Док, с. 317 — 346] приведены функции для построения различных популярных графиков, которые мы не будем описывать в деталях а приведем ниже их список. Отметим, что некоторые их них не строят графики а подготавливают данные для работы других функций. Желающие, при необходимости, смогут воспользоваться ими опираясь на описания уже рассмотренных функций.

№	Имя функции	Назначение
1.	`semilogx`	логарифмическая шкала абсцисс
2.	`semilogy`	логарифмическая шкала ординат
3.	`loglog`	логарифмические шкалы обоих осей
4.	`bar`	различные виды столбчатый диаграмм
5.	`barh`	горизонтальные столбчатые диаграммы
6.	`hist`	гистограммы
7.	`stemleaf`	диаграмма стебель-листья
8.	`printd`	вспомогательная функция для temleaf
9.	`stairs`	ступенчатая диаграмма
10.	`stem`	диаграмма стебель
11.	`stem3`	трехмерная диаграмма стебель
12.	`scatter`	диаграмма рассеивания
13.	`plotmatrix`	диаграмма рассеивания для двух матриц
14.	`pareto`	диаграмма Парето
15.	`rose`	угловая гистограмма
16.	`contour`	линии уровня
17.	`contourf`	линии уровня залитые цветом
18.	`contourc`	вычисление данных для функций contour, contourf и contour3
19.	`contour3`	трехмерные линии уровня

№	Имя функции	Назначение
20.	errorbar	линия со столбцами погрешности
21.	semilogxerr	линия со столбцами погрешности с логарифмическим масштабом по оси абсцисс
22.	semilogyerr	линия со столбцами погрешности с логарифмическим масштабом по оси ординат
23.	loglogerr	линия со столбцами погрешности с логарифмическим масштабом по обоим осям
24.	polar	двумерный график в полярных координатах
25.	pie	круговая диаграмма
26.	pie3	трехмерная круговая диаграмма
27.	quiver	векторное поле
28.	quiver3	трехмерное векторное поле
29.	streamribbon	трехмерный потоковый график ленты из векторных данных об объеметрубы вдоль линий тока, масштабируемые расходимостью векторного поля
30.	streamtube	визуализация трубки вдоль линий потока, масштабируемых дивергенцией векторного поля
31.	ostreamtube	визуализация расширения потока и сжатия векторного поля из-за локальной междупотоковой дивергенции
32.	streamline	визуализация потоков двумерных и трехмерных векторных полей
33.	stream2	вычисление линий потоков двумерного векторного поля
34.	stream3	вычисление линий потоков трехмерного векторного поля
35.	compass	визуализация компонент векторного поля, исходящих из начала графика в полярных координатах
36.	feather	визуализация компонентов векторного поля, исходящих из равноудаленных точек на оси абсцисс
37.	pcolor	двумерный график плотности
38.	area	цветовое заполнение под кривыми
39.	fill	заполненные цветом двумерные многоугольники
40.	fill3	заполненные цветом трехмерные многоугольники
41.	comet,	простая анимация в стиле кометы

№	Имя функции	Назначение
	comet3	

В [Док, с. 346 — 351] приведены функции конфигурации объекта axes (оси), которые мы не рассматриваем.

Функция fplot — неявное задание данных

Функция fplot строит двумерные графики с линейными осями, используя имя функции и ограничения для диапазона независимой переменной вместо явных заданных пар значений абсцисс и ординат.

Например,

fplot (@sin, [-10, 10], 201);

создает график, эквивалентный приведенному выше, но также включает легенду, отображающую

название отображаемой функции.

Функцию можно задавать в следующих формах.

```
fplot (fn)
fplot (fn, limits)
fplot ( . . . , tol)
fplot ( . . . , n)
fplot ( . . . , fmt)
fplot ( . . . , property, value, . . . )
fplot (hax, . . . )
[x, y] = fplot ( . . . )
```

Параметры задаются следующим образом.

- fn — задает функцию для построения графика как анонимную или в виде дескриптора или как текстовую строку;

- limits — границы диапазона изменения независимой переменной. Задаются двумя — [xlo, xhi] или четырьмя значениями [xlo, xhi, ylo, yhi], где x обозначает независимую переменную, y — значение функции, lo — нижнюю границу диапазона, hi — верхнюю; значение по умолчанию — [-5, 5].
- tol — относительный допуск, используемый для построения графика, значение по умолчанию он $2 \cdot 10^{-3}$ (0.2%).
- n — минимальное количество точек графика. Если n указано, то максимальный размер шага вычисляется по формуле (xhi - xlo) / n. Отмнтим, что для соответствия значению относительного допуска может использоваться более чем n точек.
- fmt — полностью аналогичен такому же параметру функции plot управляет стилем линии и др. Может быт также задан в виде пар вида property, value (свойство, значение).
- hax — необязятельный параметр — значение дескриптора объекта axes (оси). При этом график будет строиться в осях, определенных этим объектом, а не в определенных текущим объектом axes (оси), ссылка на дескриптор которго задается функцией gca.

Параметры tol, n и fmt являются необязательными, любое их количество может быть задано в любом порядке.

Если вызов задан в формате [x, y] = fplot (. . .) , то левая часть [x, y] получает значения данных для построения графика, по котрым он может быть построен функцией plot(x,y). В противном случае выводится файл графика.

Пример:

fplot (@cos, [0, 2*pi])

Примечания по программированию:

fplot лучше всего работает с непрерывными функциями. Функции с разрывами вряд ли будут хорошо отображаться.

Это ограничение может быть снято в будущем.