

Введение в gnuplot

Пережогин А.С.

Создан: 1 января 2010 г. Изменен: 16 февраля 2012 г.

drew72156@yandex.ru

Содержание

1	Введение	1
2	Интерфейсы вывода gnuplot	3
3	Построения на плоскости. Двумерные графики	3
3.1	Параметрические функции	5
3.2	Полярные координаты	6
3.3	Построение нескольких графиков в одном окне	7
3.4	Добавление основной и дополнительной сетки	7
4	Построения в пространстве	8
5	Использование дискретных данных	10
6	Контрольные задания	12

1 Введение

Gnuplot (авторское написание – gnuplot, с прописной буквы) – программа для построения графиков функций и визуализации различных данных. Gnuplot используется во многих математических пакетах, таких как Matlab, Maxima, Scilab, GNU Octave и др.

Стоит отметить ключевые моменты, чтобы каждый пользователь решил для себя, стоит ли ему использовать в своей работе именно gnuplot, ведь существует большое количество других подобных программ, например, QtiPlot или SciDAVis, которые распространяются по лицензии GNU GPL. Итак, особенности gnuplot:

- программа использует собственную лицензию (несмотря на свое название, gnuplot не является частью проекта GNU): лицензия gnuplot допускает выпуск модифицированных версий в виде патчей к исходному коду, что открывает большие возможности по оптимизации системы для решения различных научных задач
- гибкий и эффективный встроенный язык программирования (в командных файлах возможна организация ветвлений, циклов и пр.)
- богатый набор встроенных команд и функций

- кроссплатформенная программа: gnuplot существует в версиях под ОС Linux, Windows, MacOS X

Дистрибутивы для различных ОС, а также примеры построения графиков можно загрузить по следующим ссылкам:

<http://sourceforge.net/projects/gnuplot/files/>

<http://gnuplot.sourceforge.net/demo/>

Работа в gnuplot осуществляется в следующих режимах:

- пакетный. Пользователь подготавливает файл, содержащий последовательный набор команд (новая команда начинается с новой строки). При работе в таком режиме командный файл должен находиться в одной директории с исполняемым файлом gnuplot.

В Linux после установки gnuplot обработка команд содержащихся в файле file.gnu выполняется следующей командой

```
$ gnuplot file.gnu
```

Если командный файл не указан, то после выполнения

```
$ gnuplot
```

программа автоматически переключается в интерактивный режим.

- интерактивный. Пользователь общается с программой с помощью командной строки в режиме реального времени

Интерфейс окна для интерактивного режима представлен на рис. 1

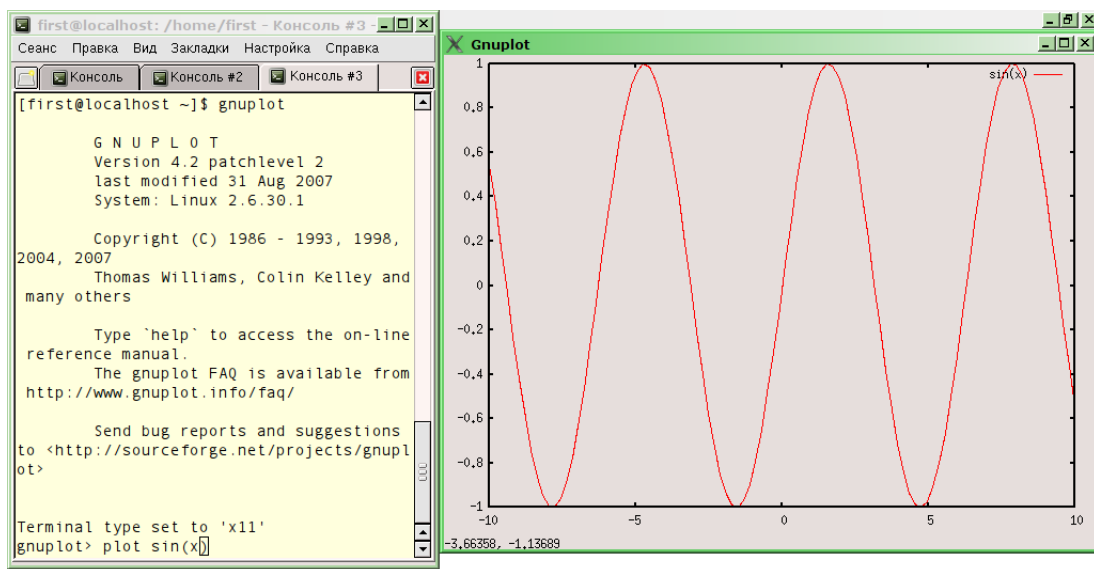


Рис. 1: Интерактивный ввод

```
save '<...\gnuplot\bin\имя файла>'
```

Перечислим некоторые базовые команды gnuplot:

help - вывести справку

load '<имя файла>' - загрузить командный файл

Наравне с load существует команда call, которая позволяет загрузить файл, указав дополнительные параметры (всего до 10 опций). Описание всех параметров можно посмотреть в справочном материале.

Вызываем командой gnuplot интерактивный режим и последовательно выполним команды для построения графика $y = \sin(x)$ и вывод его в файл "plot1.jpg".

```
set terminal jpeg                # вывод в JPEG-файл
set output "plot1.jpg"         # назначаем выходной файл
set xrange [-2*pi:2*pi]       # изменение по Ox, Oy
set yrange [-2:2]
plot sin(x)                    # строим график
```

|| **Замеч.:** gnuplot чувствителен к регистру вводимых команд ||

2 Интерфейсы вывода gnuplot

Терминалом в gnuplot является то устройство (или файл), в которое будет осуществляться вывод полученного результата. Таковым может быть монитор, принтер или же файл с расширением png, jpg, eps и др., а также latex-файл. По умолчанию построения выводятся на монитор, но при необходимости можно изменить устройство вывода командой:

```
set terminal <тип терминала>
```

Ниже приведены некоторые терминалы gnuplot:

windows - вывод данных на дисплей в ОС Windows

X11 - вывод данных на дисплей в ОС Linux

png - вывод данных в файл формата png (растровый формат)

jpeg - вывод данных в файл формата jpeg (растровый формат)

postscript eps - вывод данных в файл формата eps (векторный формат)

latex - вывод данных в файл формата L^AT_EX

Пример вывода в файл (по умолчанию файл создается в директории \gnuplot\bin\):

```
set terminal postscript eps     #устанавливаем тип терминала
set output "plot1.eps"        #устанавливаем имя выходного файла
```

(для просмотра файлов формата EPS существует утилита GSview, которую можно скачать с ресурса: <http://www.cs.wisc.edu/~ghost/gsvie/>. В Linux доступны стандартные программы для просмотра eps-файлов Okular, evince)

3 Построения на плоскости. Двумерные графики

Для построения графика функции на плоскости используется команда plot:

```
plot [<изменение аргумента>] [<изменение функции>] <функция> <доп. параметры>
```

В качестве примера построим график синусоиды, пусть x изменяется от -2π до 2π :

```
plot [-2*pi:2*pi] sin(x)
```

Область изменения значений аргумента/функции можно задать до построения, для этого нужно ввести:

```
set xrange [<нач. значение>:<конечн. значение>]
set yrange [<нач. значение>:<конечн. значение>]
```

При выводе gnuplot позволяет устанавливать различные визуальные параметры для графика. Так, для отрисовки самого графика существует 8 простых и 14 расширенных стилей. Для их использования в команде `plot` после объявления функции следует ввести:

```
with <стиль графика> // linestyle <тип, целое число (комбинация стиль+цвет)>
```

Некоторые простые стили:

```
lines (по умолч.) - линии
points - точки
lines and points - линии с точками
dots - очень маленькие точки
impulses - дискретные прямые
steps - ломаная под прямым углом линия
```

Для изменения цвета и толщины линии графика в команде `plot` нужно указать:

```
linestyle <цвет, целое число> // linewidth <толщина линии, pt>
```

|| **Замеч.:** при вводе многие команды gnuplot можно сокращать: например, вместо `linestyle` набирать `ls` (см. справку gnuplot) ||

Выполним то же построение, но с использованием полного набора команд:

```
set terminal postscript eps enhanced color # вывод в eps-файл
set output "plot1.eps" # назначаем выходной файл
set ylabel "label_y" font "Helvetica,18" # ось OY, шрифт
set xlabel "label_x" font "Helvetica,18" # ось OX, шрифт
set bmargin 4 # отступ снизу,
set lmargin 10 # отступ слева
set rmargin 10 # отступ справа
set tmargin 4 # отступ сверху
set xtics ("10"0,"20"1,"30"2) # метки по осям Oх,Oу
set ytics ("-30"-2,"-20"-1,"0"0,"20"1,"30"2)
set xrange [-2*pi:2*pi] # изменение по Oх, Oу
set yrange [-2:2]
plot sin(x) ln 1 lw 2 # построение, цвет, толщина
```

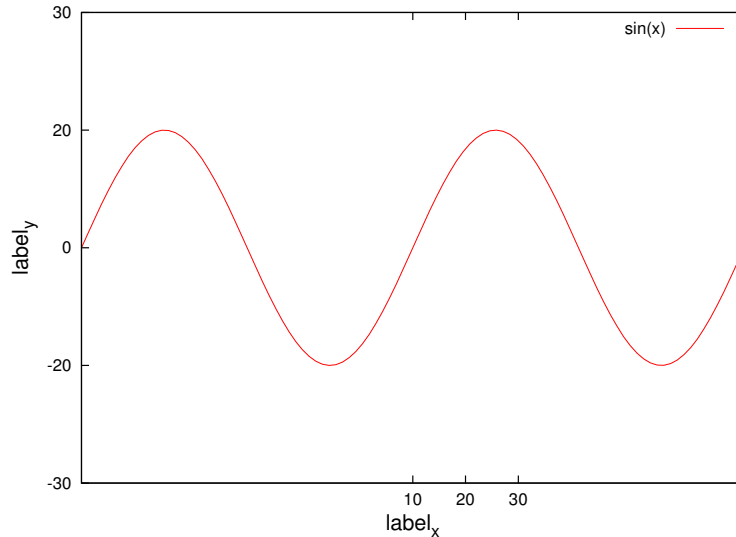


Рис. 2: График функции $y = \sin(x)$

|| [Замеч.:](#) команды с синтаксисом `set <>` можно отменить путем ввода `unset <>` ||

3.1 Параметрические функции

Для построения графиков параметрических функций необходимо ввести:

```
set parametric
```

В `gnuplot` при подобных построениях параметром является переменная `t`. Построим эллипс, заданный в параметрическом виде:

```
set terminal postscript eps enhanced color
set output "plot2.eps"
set parametric                               # параметрический график
set size square                               # квадратные пропорции рисунка
set xrange [-1:1]
set yrange [-1:1]
set trange [0:2*pi]                          # интервал параметра t
set sample 300                               # число точек, разбивающих интервал t
f(x)=sin(x)                                  # объявляем функции
g(x)=cos(x)
plot f(t), g(t) with points ls 3             # построение, точечный
```

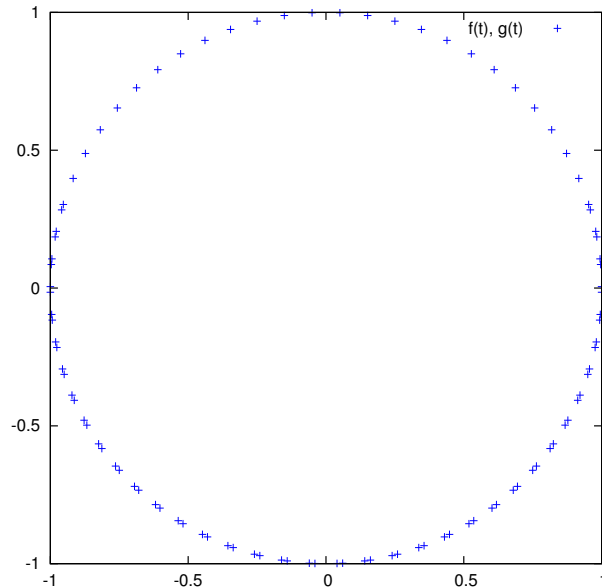


Рис. 3: Эллипс, заданный параметрически

3.2 Полярные координаты

По умолчанию gnuplot производит построения в декартовой системе координат. Для переключения на полярную систему координат нужно ввести:

```
set polar
```

При этом аргументом является переменная t , как и в случае параметрических функций. Посторим полярные графики функций $\rho = 3\cos(5\varphi)$ и $\rho_1 = 3\cos(3\varphi)$ (в этом случае для того, чтобы линии выглядели плавными, установим параметр `set samples`, отвечающий за кол-во точек для отрисовки графика):

```
set terminal postscript eps enhanced color
set output "plot3.eps"
set polar # полярные координаты
set grid polar # полярная коорд. сетка
unset key # откл. легенды
set samples 300 # кол-во точек построения
set xrange [0:2*pi]
set yrange [-pi:pi]
plot 3*cos(5*t), 3*cos(3*t) ls 3,1 # построение
```

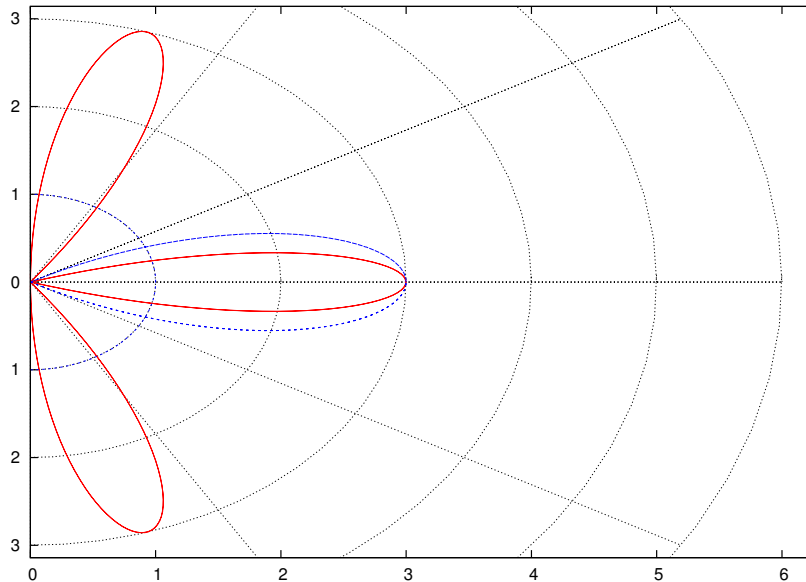


Рис. 4: Графики функций $\rho = 3\cos(5\varphi)$ и $\rho_1 = 3\cos(3\varphi)$ в полярных координатах

3.3 Построение нескольких графиков в одном окне

Командой `set multiplot` программа `gnuplot` входит в режим `multi-plot`, командная строка `Gnuplot` обращается в `multiplot>`. В этом режиме новая фигура накладывается на старую. Следующий пример показывает наложение трех функций: $y=x$, $y = x^2$ и $y = x^3$.

```
gnuplot> set multiplot
multiplot> plot x
multiplot> plot x*x
multiplot> plot x*x*x
multiplot> set nomultiplot
```

Чтобы избежать разных подписей по координатным осям необходимо добавить

```
gnuplot> set xrange [-10:10]
gnuplot> set yrange [-10:10]
gnuplot> set nokey
gnuplot> set multiplot
multiplot> plot x
multiplot> plot x*x
multiplot> plot x*x*x
multiplot> set nomultiplot
```

3.4 Добавление основной и дополнительной сетки

Команда `set grid` создает сетку на главных позициях делений. Для того чтобы создать их на дополнительных делениях:

```
gnuplot> set mxtics 5
gnuplot> set mytics 5
gnuplot> set grid xtics ytics mxtics mytics
```

Это дает пять координатных сеток на дополнительных делениях между главными. Можно конкретизировать позицию сетки опциями `xtics ... mytics` команды `set grid`. Если использовать `set grid noxtics noytics mxtics mytics`, координатные сетки появятся только на дополнительных делениях.

4 Построения в пространстве

Для построения поверхностей в `gnuplot` применяется команда `splot`. Обращение к ней происходит аналогично команде `plot`, за исключением некоторых особенностей. Выделим некоторые команды, применяемые исключительно к объемным построениям (их описание см. в построениях):

```
set zrange [<нач. значение>:<конечн. значение>]
set surface
set pm3d
set palette <параметры настройки цвета>
set view <поворот по x> <поворот по z> <масштаб> <масштаб по z>
set contour base|surface
set isosamples <целое число>
```

Следует отметить, что при параметрических построениях в пространстве в качестве параметров выступают переменные `u, v`. В остальном все так же, как и в случае плоскости.

Выполним некоторые построения.

```
set terminal postscript eps enhanced color
set output "splot3.eps"
set grid
set xrange [-10:10]
set yrange [-10:10]
splot x*y
```

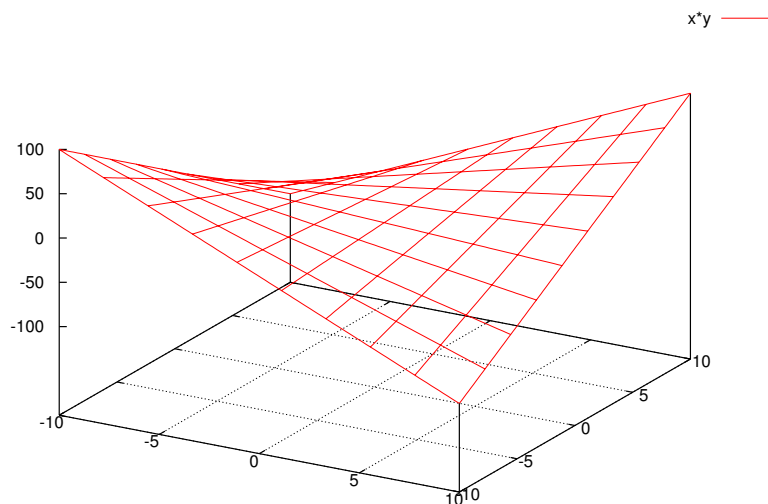


Рис. 5: Поверхность, заданная функцией $z = x * y$

```
set terminal postscript eps enhanced color
```



```

set output "splot4.eps"
set grid xtics ytics ztics      # вкл. сетку по всем осям
set xlabel "x"
set ylabel "y"
set key top                     # поместить легенду вверх
set border 4095                 # отобразить границы бокса
set xrange [-15:15]
set yrange [-15:15]
set zrange [-0.25:1]           # вид по оси Oz
set pm3d                       # заливка поверхности палитрой
set palette defined ( 1 "blue", 1 "yellow", 2 "red" ) # настройка палитры RGB
splot sin(sqrt(x**2+y**2))/sqrt(x**2+y**2) ls 6

```

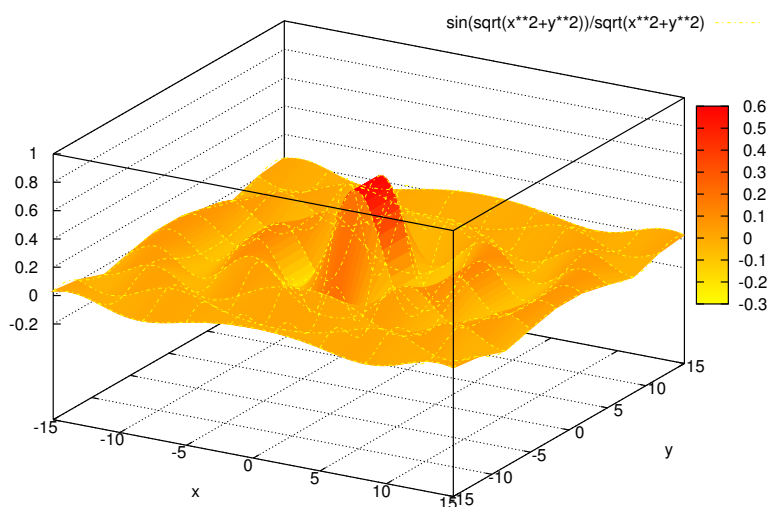


Рис. 6: Поверхность, заданная функцией $z = \frac{\sin\sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}$

```

set terminal postscript eps enhanced color
set output "splot4_1.eps"
unset key
unset pm3d                    # откл. заливку поверхности
unset surface                 # откл. сетку поверхности
unset colorbox               # откл. вывод colorbox-а (см. рис.)
set contour surface          # линии уровня на поверхности
unset xlabel
unset ylabel
unset ztics
set zrange [-0.25:1]
set samples 300
set view 0,0,1,1            # вид сверху
splot sin(sqrt(x**2+y**2))/sqrt(x**2+y**2) lw 2

```

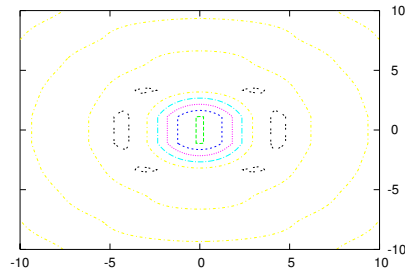


Рис. 7: Линии уровня поверхности, заданной функцией $z = \frac{\sin\sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}$

```

set terminal postscript eps enhanced color
set output "splot6.eps"
unset key
unset pm3d
set grid xtics ytics ztics
set samples 100
set parametric
splot [-2*pi:2*pi] cos(u)*cos(v),cos(u)*sin(v),sin(u)

```

Рис. 8: Эллипсоид, заданный параметрически

5 Использование дискретных данных

Одной из отличительных особенностей gnuplot является то, что формировать данные для построения графиков можно не только задавая промежутки изменения переменных, но и делая это вручную, причем можно как использовать заранее подготовленный файл с данными, так и вводить их в интерактивном режиме 'на лету'. Рассмотрим первый вариант (данные из файла).

В качестве примера зададим матрицу значений в текстовом файле, после чего произведем построение в gnuplot. Итак, создаем в директории `\gnuplot\bin\` файл `mtx1` и в нем:

```

1 1 5          # данные в виде матрицы ixj, i=1,j=1
1 2 -3        # i=1,j=2...
1 3 -8
2 1 -4

```

```
2 2 6
2 3 1
```

Затем в командной строке gnuplot:

```
set terminal postscript eps enhanced color
set output "splot7.eps"
splot "mtrx1" lw 3          # строим график по дискретным данным
```

Рис. 9: Точечный график: визуализация данных из файла

В том случае если данные представлены в текстовом файле mtrx2 в виде матрицы

```
1 2 1 3
-1 3 1 -4
1 2 -3 4
5 1 1 5
```

..то команда для построения графика будет иметь вид:

```
set terminal postscript eps enhanced color
set output "splot8.eps"
splot "mtrx2" matrix lw 3      # строим график по дискретным данным
```

6 Контрольные задания

1. Перечислить терминалы вывода графиков. Указать растровые и векторные форматы.
2. Выполнить построение графика функции $y = 3 \cos(2x + 4)$, $x \in (-2, 2)$. Вывести график в формат eps, png, jpeg.
3. Построить линию, заданную параметрически
$$\begin{cases} x(t) = 3 \cos(t) \\ y(t) = 4 \sin(t) \end{cases}$$
 $t \in (0, 2\pi)$.
4. Построить в полярной системе координат график функции $\rho = \sin(t)$, $t \in (0, 2\pi)$. Задать цвет графика – синий.
5. Построить график функции $z = \cos(x + y)$, $x \in (-2, 2)$, $y \in (-2, 2)$.
6. Для матрицы размерности 5x5 построить её график. Значения элементов матрицы считываются из файла.