1 gnuplot

gnuplot — свободная программа для создания двух- и трёхмерных графиков.

gnuplot имеет собственную систему команд, может работать интерактивно (в режиме командной строки) и выполнять скрипты, читаемые из файлов. Также используется в качестве системы вывода изображений в различных математических пакетах.

gnuplot выводит графики как непосредственно на экран (интерактивный режим), так и в файлы различных графических форматов (командный режим работы), таких как PNG, EPS, SVG, JPEG и множество других. Программа также может генерировать код на LaTeX, позволяя использовать шрифты и формулы LaTeX.

1.1 Установка

Официальный сайт проекта `http://www.gnuplot.info` содержит пункт "Download": бинарные установщики для Windows, исходные коды для сборки (необходим компилятор Си).

Для различных версий Linux (если `gnuplot` не установлен по-умолчанию) при наличии подключения к сети Internet достаточно обратиться к репозитарию.
Fedora Linux: `#yum install gnuplot`
Debian Linux: `#apt-get install gnuplot`

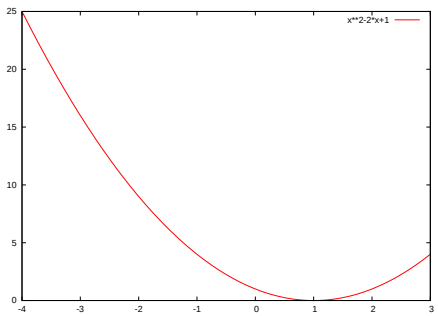
1.2 Примеры

1.2.1 Простейший график.

График, заданный формулой $y = x^2 - 2x + 1$ на отрезке $x \in [-4; 3]$.

```
$ gnuplot
gnuplot> plot [-4:3][*:*] x**2-2*x+1
```

Результат:



Примечание 1. Для получения данного рисунка стандартный вывод на экран был переопределен в файл с помощью команд:

```
set terminal postscript eps enhanced color
set output 'prim1.eps'
plot [-4:3][*:*] x**2-2*x+1
```

Примечание 2. Возведение в степень ** соответствует языку FORTRAN.

Строка
`set terminal postscript eps enhanced color`
устанавливает тип терминала для использования postscript–драйвера. Следующая строка
`set output 'prim1.eps'`
устанавливает вывод результата в заданный файл.

Используемые опции:
landscape и **portrait** устанавливают ориентацию печати, **eps** позволяет генерировать eps–файл (при этом вывод обязательно должен быть перенаправлен в файл);
enhanced — "расширенное" использование текста (в том числе верхние индексы `^`, нижние индексы `_`, греческие буквы $\alpha \rightarrow$`/Symbol a`);
color — позволяет использовать цвет в полученных рисунках.

Полный вариант использования postscript–драйвера в справочной информации описывается следующим образом:

```
set terminal postscript {default}
set terminal postscript {landscape | portrait | eps}
                        {enhanced | noenhanced}
                        {defaultplex | simplex | duplex}
                        {fontfile [add | delete] "<filename>"
                         | nofontfiles} {{no}adobeglyphnames}
                        {level1 | leveldefault}
                        {color | colour | monochrome}
                        {solid | dashed}
                        {dashlength | dl <DL>}
                        {linewidth | lw <LW>}
                        {rounded | butt}
                        {clip | noclip}
                        {palfuncparam <samples>{,<maxdeviation>}}
                        {size <XX>{unit},<YY>{unit}}
                        {blacktext | colortext | colourtext}
                        {{font} "fontname{,fontsize}" {<fontsize>}}
                        {fontscale <scale>}
```

По умолчанию график строится тонкой сплошной линией без выделения точек.

1.2.2 Несколько графиков в одних осях.

Построение нескольких графиков на одном рисунке по данным из файла.

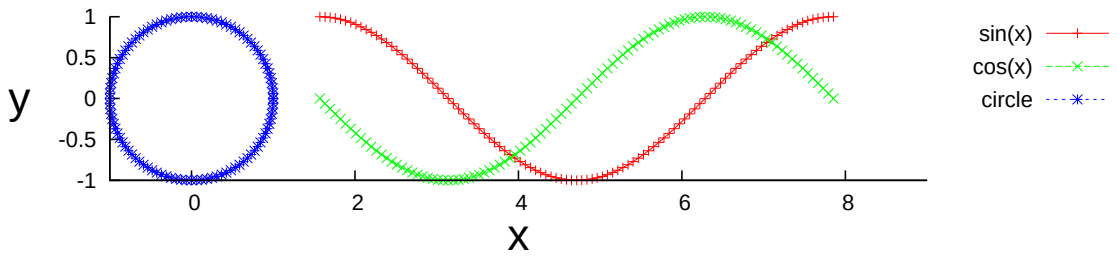
Данные в файле `data.txt` получены в результате работы следующей простой программы:

```
#include <stdio.h>
#include <math.h>
int main(void)
{
int i; double x,h=M_PI*0.02; FILE *fout=NULL;
fout=fopen("data.txt","w");
for(i=0;i<101;i++)
{ x=M_PI/2+i*h; fprintf(fout,"%lg %lg %lg\n",x,sin(x),cos(x)); }
return 0;
}
```

Используем командный файл:

```
# Устанавливаем обозначения осей, шрифт, размер,
# и запрет на вращение оси y:
set xlabel "x" font "Helvetica,28"
set ylabel "y" font "Helvetica,28" rotate by 0
# Устанавливаем явные границы (* - автовыбор):
set xrange [-1:9]
set yrange [-1:1]
# Устанавливаем отображение только левой и нижней осей:
set border 3
set xtics nomirror
set ytics nomirror
# Устанавливаем отображение легенды вне поля рисунка
set key outside spacing 1.5
# Устанавливаем явный масштаб соотношения осей
# (для корректного изображения окружности):
set size ratio 0.2
# Устанавливаем тип eps и явное имя файла
set terminal postscript eps enhanced color
set output 'prim2.eps'
# Печатаем три графика; символ \ служит
# для продолжения команды на следующей строке:
plot "data.txt" using 1:2 with linespoints title "sin(x)", \
"data.txt" using 1:3 w lp title "cos(x)", \
"data.txt" using 2:3 w lp title "circle"
```

Результат:



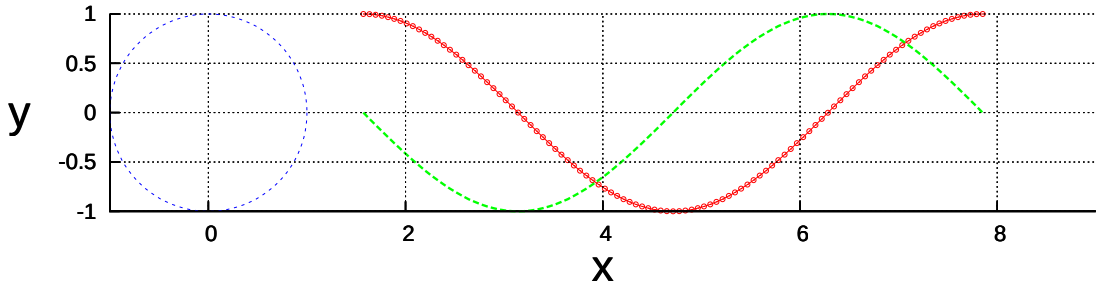
Основные комментарии содержатся в командном файле. Наиболее интересна последняя команда (три последних строки):
`using 1:2` обозначает использование первой и второй колонки;
`with linespoints` и `w lp` обозначает одно и то же — использовать линию с точками (цвет и форма точек выбираются по умолчанию);
`title "название"` задает название графика; построение графиков отделяется друг от друга запятой.

При замене конца предыдущего командного файла следующими строками (или добавлении этих строк в конец)

```
# Устанавливаем явный масштаб соотношения меток на осях
set size ratio -1
set output 'prim3.eps'
# Удаляем описание (легенду)
unset key
# Устанавливаем сетку по умолчанию
set grid
# Устанавливаем стили двух линий
set style line 1 lt 1 lc 1 pt 6 ps 0.5
set style line 2 lt 2 lc rgb "#00FF00" lw 3 pt 4 ps 0.4
# Другой способ печати нескольких графиков
set multiplot
```

```
plot "data.txt" using 1:2 w lp ls 1 # Используем линию стиля 1
plot "data.txt" using 1:3 w l ls 2
plot "data.txt" using 2:3 w l lt 3 lc 3 # Явно описываем линию
set nomultiplot
```

получаем результат (при добавлении будет создано два eps-файла):



Вообще `multiplot` может использоваться для построения значительно более интересных графиков
(см., например, `http://mydebianblog.blogspot.com/2008/02/gnuplot_18.html`).

Стили линий:

`lt` — тип линии (1 — непрерывная);

`lc` — цвет линии: номер (линии 1 и 3) или RGB-цвет (линия 2);

`lw` — толщина линии;

`pt` — тип точек;

`ps` — размер точек.

Следует обратить внимание, на линию 2: хотя в описании линии точки присутствуют, на графике они не изображаются (при определении графика вместо `lp` используется `l`).

1.2.3 Простая диаграмма.

Командный файл:

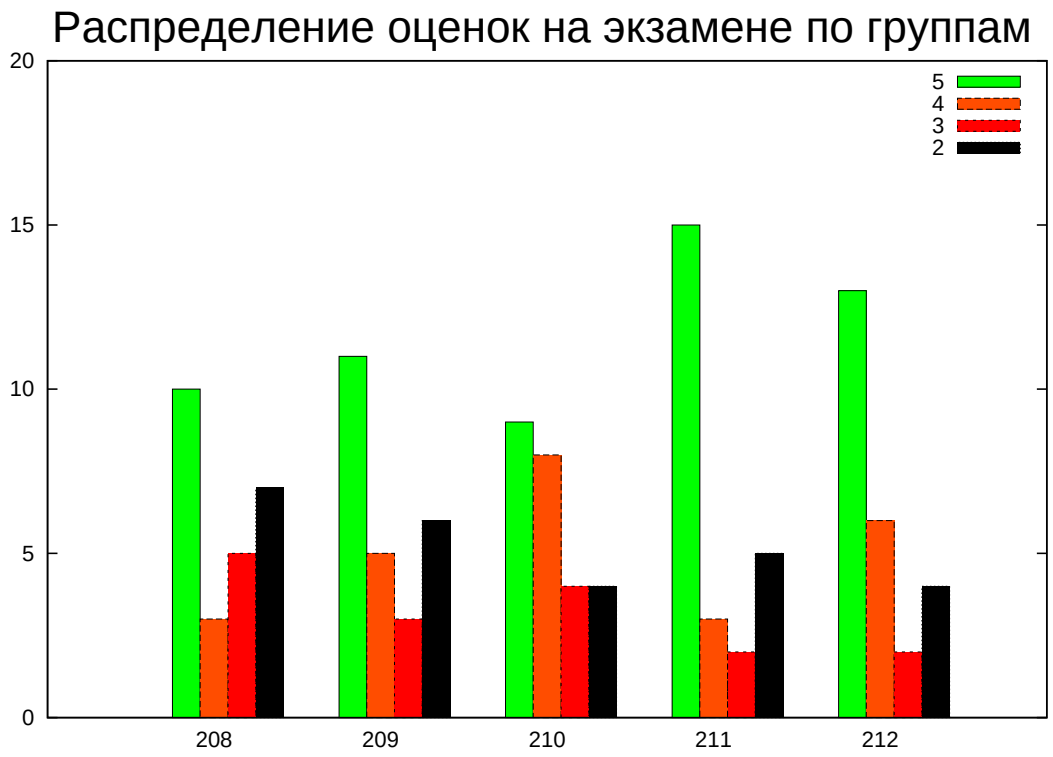
```
set terminal postscript eps enhanced color
```

```
set output 'diag1.eps'
# С utf8, кажется, пока проблемы...
set encoding koi8r
set title "Распределение оценок на экзамене по группам" \
font "Helvetica,28"
# Запрет на установку подписей к осям
# Точнее подписи состоят из одного пробела
set ylabel " "
set xlabel " "
# Установка диапазона
set yrange [0:20]
# Данные являются диаграммой
set style data histogram
# solid - столбики закрашенные
set style fill solid 2 border -1
set xtic rotate by 0 scale 0
# Напечатать данные из файла 'diag2.dat'
# В качестве меток использовать столбец 1
# Для столбцов переопределяются цвета
plot 'diag1.dat' using 2:xtic(1) ti col lc 2, '' u 3 ti col lc 8, \
'' u 4 ti col lc 1, '' u 5 ti col lc 0
```

Файл данных `diag1.dat`:

```
name "5" "4" "3" "2"
"208" 10 3 5 7
"209" 11 5 3 6
"210" 9 8 4 4
"211" 15 3 2 5
"212" 13 6 2 4
```

Полученный результат:



1.2.4 Диаграмма с двумя осями.

Командный файл:

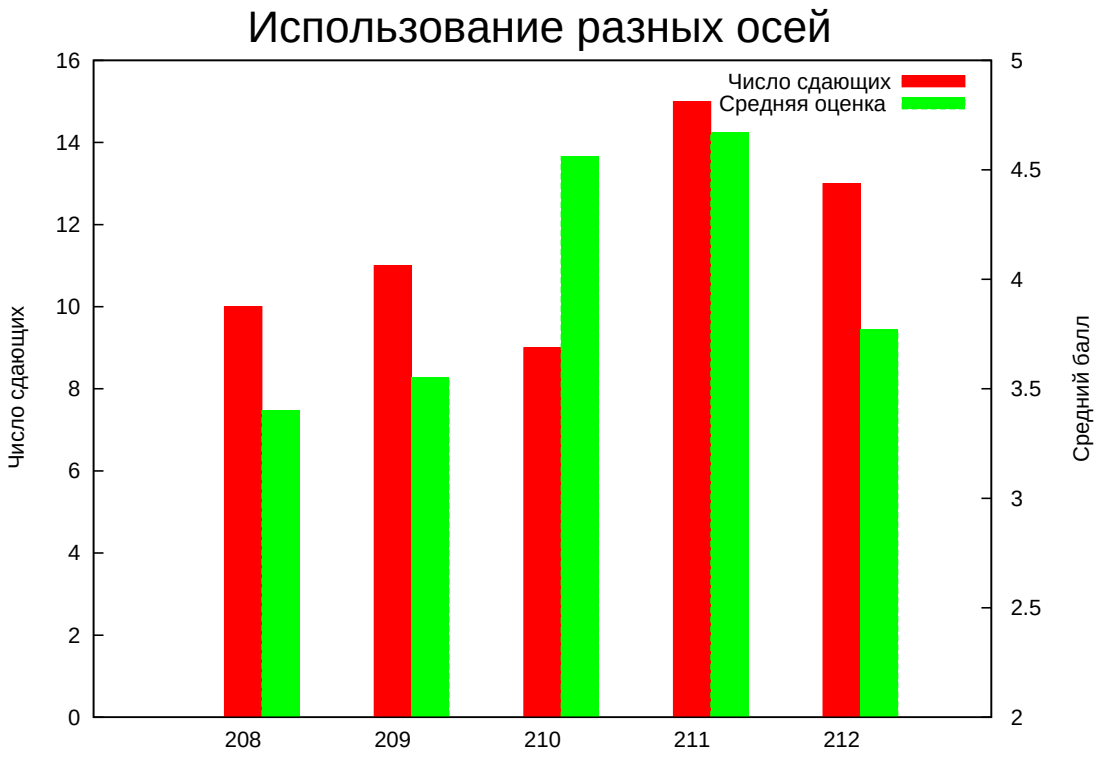
```
#! /usr/bin/gnuplot
set terminal postscript eps enhanced color
set output 'diag2.eps'
# С utf8, кажется, пока проблемы...
set encoding koi8r
set title "Использование разных осей" font "Helvetica,28"
set ylabel "Число сдающих"
set y2label "Средний балл"
# Нижняя ось не подписывается
set xlabel " "
# Явное задание диапазонов на левой и правой осях (* - автоматически)
set yrange [0:*]
set y2range [2:5]
# Задание диаграммы (если строку убрать будет выведен график точками)
set style data histogram
# Заполненные столбики в диаграмме
set style fill solid
# горизонтальные отметки на нижней оси x
set xtic rotate by 0 scale 0
# Запрет на создание зеркальной копии меток на правой оси
```

```
set ytic nomirror
# Создание своеей разметки на правой оси
set y2tic
# Напечатать данные из файла 'diag2.dat'
# В качестве меток использовать столбец 1
# Для третьего столбца использовать нижнюю и правую ось x1y2
plot 'diag2.dat' using 2:xtic(1) ti col, '' u 3 ti col axes x1y2
```

Файл данных `diag1.dat`:

```
name "Число сдающих" "Средняя оценка"
"208" 10 3.40
"209" 11 3.55
"210" 9 4.56
"211" 15 4.67
"212" 13 3.77
```

Полученный результат:



1.2.5 Построение простой поверхности.

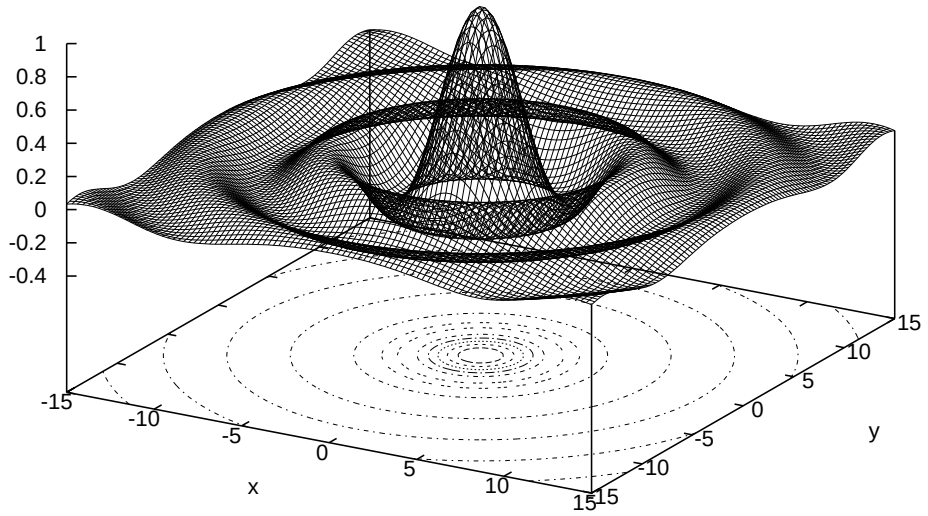
Достаточно стандартный пример построения поверхности

$$z = \frac{\sin\left(\sqrt{x^2+y^2}\right)}{\sqrt{x^2+y^2}}.$$

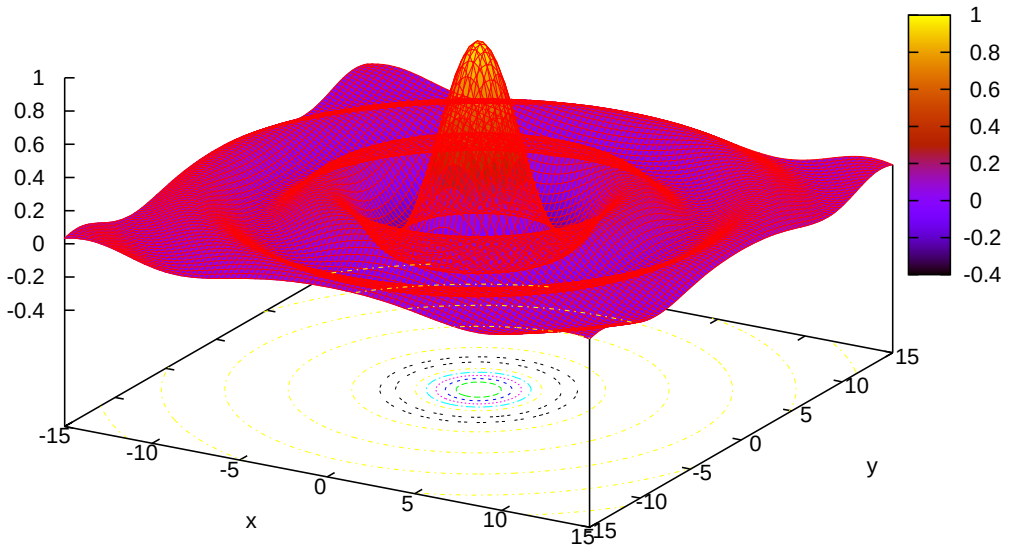
Командный файл:

```
set xlabel "x"
set ylabel "y"
unset key
set xrange [-15:15]
set yrange [-15:15]
set zrange [*:*]
set surface
set contour base
set view 60,30,1.0,1.0
set isosamples 100
set cntrparam levels auto 10
set terminal postscript eps enhanced
set output 'prim4.eps'
splot sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
```

Результат:



Добавление команды `set pm3d` поверхность раскрашивает.



1.2.6 Поверхность по расчетным данным.

Подготовка расчетных данных.

Строить поверхности в `gnuplot`-е можно не только для известных вычисляемых функций но и по заранее подготовленным данным. Стандартный формат входного файла в этом случае содержит три столбца чисел, соответствующих x, y и z (разумеется, их можно переопределить на любые нужные номера). При этом значения x и y должны полностью заполнять сетку и идти в определенном порядке. А именно: выбираются некоторые заданные значения $x_0 < x_1 < x_2 < \ldots < x_M$ и $y_0 < y_1 < y_2 < \ldots < y_N$. Затем для выбранного x_0 последовательно строчка за строчкой приводятся тройки чисел для всех выбранных y_n $n \in \{0, \ldots, N\}$, следующий блок для x_1 отделяется пустой строкой, и так до последнего блока чисел для x_M:

$x_0\ y_0\ z_{00}$

$x_0\ y_1\ z_{01}$

$x_0\ y_2\ z_{02}$

$\ldots$

$x_0\ y_N\ z_{0N}$

"пустая строка"

$x_1\ y_0\ z_{10}$

$x_1\ y_1\ z_{11}$

$x_1\ y_2\ z_{12}$

$\ldots$

$x_1\ y_N\ z_{1N}$

"пустая строка"

...

...

"пустая строка"

$x_M\ y_0\ z_{M0}$

$x_M\ y_1\ z_{M1}$

$x_M\ y_2\ z_{M2}$

...

$x_M\ y_N\ z_{MN}$

"конец файла данных"

Любой пропуск какой-либо тройки чисел или их перестановка приводит к появлению диагностики "График не может быть построен" и предложению проинтерполировать недостающие данные.

Файл данных `data2.txt` готовится следующей несложной программой (нарочито выбирается мало точек по x и y):

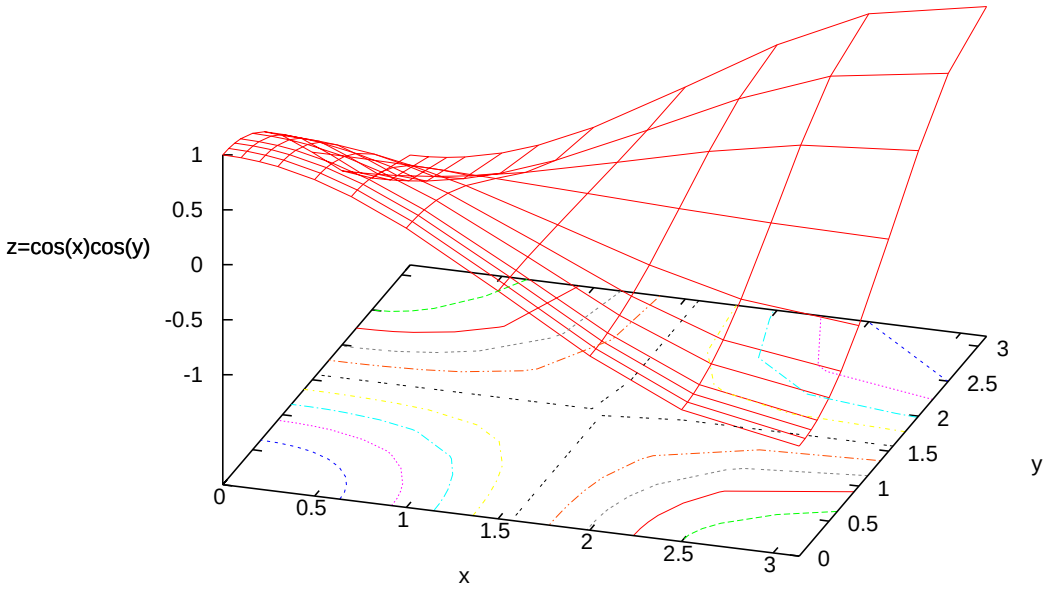
```
#include <stdio.h>
#include <math.h>
int main(void)
{
int i,j;
double x[]={0,0.1,0.2,0.3,0.5,0.7,1.0,1.5,2.0,2.5,M_PI},
y[]={0,0.1,0.2,0.3,0.5,0.7,1.0,1.5,2.0,2.5,M_PI};
FILE *fout=NULL;
fout=fopen("data2.txt","w");
for(i=0;i<11;i++) {
for(j=0;j<11;j++)
fprintf(fout,"%lg %lg %lg\n",x[i],y[j],cos(x[i])*cos(y[j]) );
fprintf(fout," \n");
}
return 0;
}
```

Командный файл:

```
set xlabel "x"
set ylabel "y"
set zlabel "z=cos(x)cos(y)"
unset key
set xrange [0:3.14159265]
set yrange [0:3.14159265]
set zrange [*:*]
```

```
set surface
set contour base
set view 55,18,1.0,1.0
set cntrparam levels auto 12
set terminal postscript eps enhanced color
set output 'prim6.eps'
splot 'data2.txt' w l
```

Результат:



1.2.7 Построение пространственных кривых.

Если `splot` получает файл данных с одним блоком (без пустых строк) он интерпретирует их как последовательность точек трехмерного пространства и соединяет их (по–видимому, сглаживание тоже допустимо).

Файл данных `data3.txt` готовится следующей несложной программой:

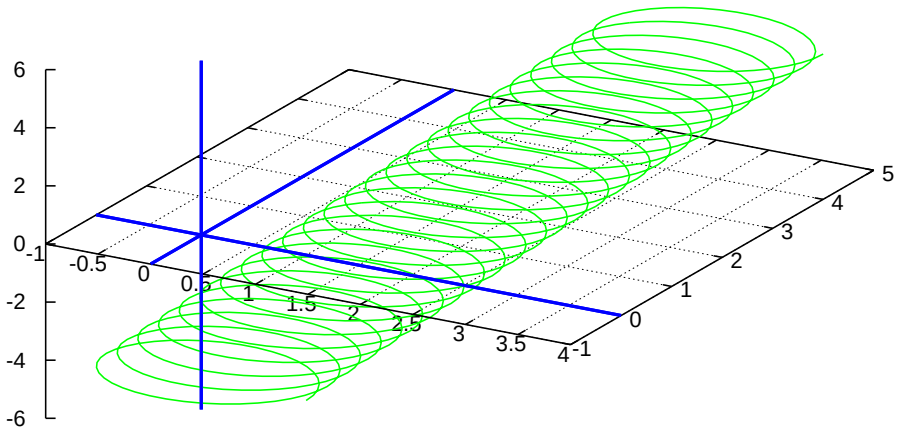
```
#include <stdio.h>
#include <math.h>
int main(void)
{
int i; double x,y,z,h_x=0.003,h_y=0.004,h_z=0.01;
FILE *fout=NULL;
fout=fopen("data3.txt","w");
for(i=0;i<1001;i++)
```

```
{
 x=i*h_x+cos(M_PI*0.05*i);
 y=i*h_y+sin(M_PI*0.05*i);
 z=i*h_z-5;
fprintf(fout,"%lg %lg %lg\n",x,y,z); }
return 0;
}
```

Командный файл:

```
unset key
set xzeroaxis lt 1 lw 4 lc 3
set yzeroaxis lt 1 lw 4 lc 3
set zzeroaxis lt 1 lw 4 lc 3
set xyplane at 0
set grid
set view 60,30,1.0,1.0
set terminal postscript eps enhanced color
set output 'prim7.eps'
splot 'data3.txt' w l lw 2 lc 2
```

Результат:



1.2.8 Анимированные графики.

Пример взят с http://www.cs.karelia.ru/studies/gnuplot/Examples.html

В примере используется два файла.

Файл `gp_rot.txt`:

```
a=0
xrot=60
zrot=0
load "move.rot"
```

Файл `move.rot`:

```
a=a+1
zrot=(zrot+10)%360
set view xrot,zrot
splot x**2+y**2
pause 1 # ускорено по сравнению с оригиналом
if (a<50) reread
```

Запуск: `gnuplot gp_rot.txt`