

# gnuplot FAQ

This document refers both to **gnuplot** version 4 and version 5.

FAQ version \$Revision: 1.56 \$, dated \$Date: 2014/12/31 23:35:35 \$.

## Contents

<b>0</b>	<b>Meta – Questions</b>	<b>3</b>
0.1	Where do I get this document? . . . . .	3
0.2	Where do I send comments about this document? . . . . .	3
<b>1</b>	<b>General Information</b>	<b>3</b>
1.1	What is <b>gnuplot</b> ? . . . . .	3
1.2	How did it come about and why is it called <b>gnuplot</b> ? . . . . .	3
1.3	What does <b>gnuplot</b> offer? . . . . .	3
1.4	Is <b>gnuplot</b> suitable for scripting? . . . . .	4
1.5	Can I run <b>gnuplot</b> on my computer? . . . . .	4
1.6	Legalities . . . . .	4
1.7	Does <b>gnuplot</b> have anything to do with the FSF and the GNU project? . . . . .	4
1.8	Where do I get further information? . . . . .	4
<b>2</b>	<b>Setting it up</b>	<b>4</b>
2.1	What is the current version of <b>gnuplot</b> ? . . . . .	4
2.2	Where can I get <b>gnuplot</b> ? . . . . .	4
2.3	Where can I get current development version of <b>gnuplot</b> ? . . . . .	5
2.4	How do I get <b>gnuplot</b> to compile on my system? . . . . .	5
2.5	What documentation is there, and how do I get it? . . . . .	5
2.6	Worked examples . . . . .	5
2.7	How do I modify <b>gnuplot</b> , and apply patches? . . . . .	6
2.8	How do I determine which options are compiled into <b>gnuplot</b> ? . . . . .	6
<b>3</b>	<b>Working with it.</b>	<b>6</b>
3.1	How do I get help? . . . . .	6
3.2	How do I print out my graphs? . . . . .	6
3.3	How do I include my graphs in <word processor>? . . . . .	7
3.4	How do I edit or post-process a <b>gnuplot</b> graph? . . . . .	7
3.5	How do I change symbol size, line thickness and the like? . . . . .	8
3.6	Can I animate my graphs? . . . . .	8
3.7	How do I plot implicit defined graphs? . . . . .	8
3.8	How to fill an area between two curves . . . . .	8
3.9	Pm3d splot from a datafile does not draw anything . . . . .	8
3.10	Drawing 2D projection of 3D data . . . . .	9
3.11	How to overlay dots/points scatter plot onto a pm3d map/surface . . . . .	9
3.12	How to produce labeled contours . . . . .	9
3.13	How to overlay contour plot over pm3d map/surface . . . . .	9
3.14	Color facets with pm3d . . . . .	9
3.15	Palette for printing my color map on color as well as black&white printer? . . . . .	10

<b>4</b>	<b>Wanted features</b>	<b>10</b>
4.1	What's new in <b>gnuplot</b> 4.6, 5.0 etc?	10
4.2	Does <b>gnuplot</b> support a driver for <graphics format>?	10
4.3	Does <b>gnuplot</b> have hidden line removal?	10
4.4	Does <b>gnuplot</b> support bar-charts/histograms/boxes?	10
4.5	Does <b>gnuplot</b> support pie charts? quarterly time charts?	11
4.6	Can I put multiple pages on one page?	11
4.7	Does <b>gnuplot</b> support multiple y-axes on a single plot?	11
4.8	Can I put both commands and data into a single file?	11
4.9	Can I put Greek letters and super/subscripts into my labels?	11
4.10	How do I include accented characters	11
4.11	Can I do 1:1 scaling of axes?	12
4.12	Can I put different text sizes into my plots?	12
4.13	How do I skip data points?	12
4.14	How do I plot every nth point?	12
4.15	How do I plot a vertical line?	12
4.16	How do I plot data files	12
4.17	How do I replot multiplot drawing	12
<b>5</b>	<b>Miscellaneous</b>	<b>12</b>
5.1	I've found a bug, what do I do?	12
5.2	Can I use <b>gnuplot</b> routines for my own programs?	13
5.3	What extensions have people made to <b>gnuplot</b> ? Where can I get them?	13
5.4	I need an integration, fft, iir-filter,...!	13
5.5	Can I do heavy-duty data processing with <b>gnuplot</b> ? or What is beyond <b>gnuplot</b> ?	13
5.6	How to use hotkeys in my interactive terminals	14
5.7	I have ported <b>gnuplot</b> to another system, or patched it. What do I do?	14
5.8	I want to help in developing the next version of <b>gnuplot</b> . What can I do?	14
5.9	Open questions for inclusion into the FAQ?	14
<b>6</b>	<b>Making life easier</b>	<b>14</b>
6.1	How do I plot two functions in non-overlapping regions?	14
6.2	How do I run my data through a filter before plotting?	14
6.3	How do I save and restore my current settings?	14
6.4	How do I plot lines (not grids) using <code>splot</code> ?	14
6.5	How do I plot a function $f(x,y)$ that is bounded by other functions in the x-y plane?	15
6.6	How do I call <b>gnuplot</b> from my own programs?	15
6.7	What if I need h-bar (Planck's constant)?	15
6.8	What if I need the Solar mass symbol?	15
6.9	How do I produce blank output page?	15
6.10	How do I give exact positions for the graph borders on the page?	16
<b>7</b>	<b>Common problems</b>	<b>16</b>
7.1	Help! None of my fonts work.	16
7.2	<b>Gnuplot</b> does not open a plot window on VMS. Why?	16
7.3	Why does <b>gnuplot</b> ignore my very small numbers?	17
7.4	When I run <b>gnuplot</b> from the shell or a script, the resulting plot flashes by on the screen and then disappears	17
7.5	My formulas (like 1/3) are giving me nonsense results! What's going on?	17
7.6	My output files are incomplete!	17
7.7	When using the $\text{\LaTeX}$ -terminal, there is an error during the $\text{\LaTeX}$ -run!	17
7.8	Calling <b>gnuplot</b> in a pipe or with a <b>gnuplot</b> -script doesn't produce a plot!	18
<b>8</b>	<b>Credits</b>	<b>18</b>

## 0 Meta – Questions

### 0.1 Where do I get this document?

The newest version of this document is on the web at <http://www.gnuplot.info/faq/>.

### 0.2 Where do I send comments about this document?

Send comments, suggestions etc via email to the developer mailing list <mailto:gnuplot-beta@lists.sourceforge.net>. Please contribute your suggestions with respect to the file `faq.tex` available from <http://gnuplot.cvs.sourceforge.net/viewvc/gnuplot/faq/>.

## 1 General Information

### 1.1 What is gnuplot?

**gnuplot** is a command-driven interactive function plotting program. It can be used to plot functions and data points in both two- and three-dimensional plots in many different formats. It is designed primarily for the visual display of scientific data. **gnuplot** is copyrighted, but freely distributable; you don't have to pay for it.

### 1.2 How did it come about and why is it called gnuplot?

The authors of **gnuplot** are: Thomas Williams, Colin Kelley, Russell Lang, Dave Kotz, John Campbell, Gershon Elber, Alexander Woo and many others.

The following quote comes from Thomas Williams:

I was taking a differential equation class and Colin was taking Electromagnetics, we both thought it'd be helpful to visualize the mathematics behind them. We were both working as sys admin for an EE VLSI lab, so we had the graphics terminals and the time to do some coding. The posting was better received than we expected, and prompted us to add some, albeit lame, support for file data.

Any reference to GNUplot is incorrect. The real name of the program is "**gnuplot**". You see people use "**Gnuplot**" quite a bit because many of us have an aversion to starting a sentence with a lower case letter, even in the case of proper nouns and titles. **gnuplot** is not related to the GNU project or the FSF in any but the most peripheral sense. Our software was designed completely independently and the name "**gnuplot**" was actually a compromise. I wanted to call it "llamaplot" and Colin wanted to call it "nplot." We agreed that "newplot" was acceptable but, we then discovered that there was an absolutely ghastly pascal program of that name that the Computer Science Dept. occasionally used. I decided that "**gnuplot**" would make a nice pun and after a fashion Colin agreed.

### 1.3 What does gnuplot offer?

- Plotting two-dimensional functions and data points in many different styles (points, lines, error bars)
- Plotting three-dimensional data points and surfaces in many different styles (contour plot, mesh)
- Algebraic computation in integer, float and complex arithmetic
- Data-driven model fitting using Marquardt-Levenberg minimization
- Support for a large number of operating systems, graphics file formats and output devices
- Extensive on-line help
- T<sub>E</sub>X-like text formatting for labels, titles, axes, data points
- Interactive command line editing and history (most platforms)

## 1.4 Is gnuplot suitable for scripting?

Yes. Gnuplot can read in files containing additional commands during an interactive session, or it can be run in batch mode by piping a pre-existing file or a stream of commands to stdin. Gnuplot is used as a back-end graphics driver by such higher-level mathematical packages as Octave, and can easily be wrapped in a cgi script for use as a web-driven plot generator.

## 1.5 Can I run gnuplot on my computer?

**Gnuplot** is in widespread use on many platforms, including MS Windows, linux, unix, and OSX. The current source code retains supports for older systems as well, including VMS, Ultrix, OS/2, and MS-DOS. 16-bit platforms are no longer supported.

You should be able to compile the **gnuplot** source more or less out of the box in any reasonably standard (ANSI/ISO C, POSIX) environment.

## 1.6 Legalities

**Gnuplot** is authored by a collection of volunteers, who cannot make any legal statement about the compliance or non-compliance of **gnuplot** or its uses. There is no warranty whatsoever. Use at your own risk. Citing from the README of a mathematical subroutine package by R. Freund:

For all intent and purpose, any description of what the codes are doing should be construed as being a note of what we thought the codes did on our machine on a particular Tuesday of last year. If you're really lucky, they might do the same for you someday. Then again, do you really feel \*that\* lucky?

## 1.7 Does gnuplot have anything to do with the FSF and the GNU project?

**Gnuplot** is neither written nor maintained by the FSF. At one time it was distributed by the FSF but this is no longer true. **Gnuplot** as a whole is not covered by the GNU General Public License (GPL).

**Gnuplot** is freeware in the sense that you don't have to pay for it. However it is not freeware in the sense that you would be allowed to distribute a modified version of your **gnuplot** freely. Please read and accept the modification and redistribution terms in the *Copyright* file.

## 1.8 Where do I get further information?

See the main gnuplot web page <http://www.gnuplot.info>.

Some documentation and tutorials are available in other languages than English. See <http://gnuplot.sourceforge.net/help.html>, section "Localized learning pages about gnuplot", for the most up-to-date list.

# 2 Setting it up

## 2.1 What is the current version of gnuplot?

The current released version of **gnuplot** is 5.0, released in January 2015. Incremental versions (patchlevel 1, 2, ...) are typically released every six months. The development version of **gnuplot** is currently 5.1.

## 2.2 Where can I get gnuplot?

The best place to start is <http://www.gnuplot.info>. From there you find various pointers to other sites, including the project development site on SourceForge <http://sourceforge.net/projects/gnuplot>.

The source distribution ("gnuplot-4.6.0.tar.gz" or a similar name) is available from the official distribution site <http://sourceforge.net/projects/gnuplot>.

## 2.3 Where can I get current development version of gnuplot?

The development version of gnuplot is available as a cvs source tree online for direct browsing from <http://sourceforge.net/projects/gnuplot>, section "CVS". You can download all current sources according to the documentation therein; for example by a sequence of commands like

```
cvs -d:pserver:anonymous@gnuplot.cvs.sourceforge.net:/cvsroot/gnuplot login
cvs -z3 -d:pserver:anonymous@gnuplot.cvs.sourceforge.net:/cvsroot/gnuplot co -P gnuplot
```

or (in bash)

```
export CVSROOT=:pserver:anonymous@gnuplot.cvs.sourceforge.net:/cvsroot/gnuplot
cvs login
cvs -z3 checkout gnuplot
```

Hit `<return>` when asked for a password.

Further, before the `./configure` command of gnuplot compilation phase, you have to execute `./prepare` to create the up-to-date configure files.

There are no official preliminary binary releases of gnuplot: you have to compile it yourself. However, you may find unofficial binary releases for some platforms, like OS/2, Windows or Macintosh.

Important note: questions related to the development version should go to <mailto://gnuplot-beta@lists.sourceforge.net>.

## 2.4 How do I get gnuplot to compile on my system?

As you would any other installation. Read the files *README.1ST*, *README*, and *INSTALL*.

- For Unix, use `./configure` (or `./configure --prefix=$HOME/usr` for an installation for a single user), `make` and finally `make install` or `make install-strip`, the latter for smaller executables without debugging information. If you want to make a RPM package, then replace the latest step by `checkinstall` or `checkinstall make install-strip`, supposing the package `checkinstall` on your machine.
- On Windows, makefiles can be found in `config/mingw`, `config/msvc`, `config/watcom`, and `config/cygwin`. Update the options in the makefile's header and run the appropriate `make` tool in the same directory as the makefile. Additional instructions can be found in the makefiles.
- For DOS, if you are using bash and DJGPP, you can just run `djconfig.sh`.
- For other platforms, copy the relevant makefile (e.g. `makefile.os2` for OS/2) from `config/` to `src/`, optionally update options in the makefile's header, then change directory to `src` and run `make`.

## 2.5 What documentation is there, and how do I get it?

Full documentation is included in the source distribution. Individual sections can be browsed from inside a gnuplot session by typing `help` keyword. Look in the docs and tutorial subdirectories, where you'll find files to produce PDF or HTML versions of the User Manual.

Online copies are available at <http://gnuplot.sourceforge.net/documentation.html>.

## 2.6 Worked examples

There is a directory of worked examples in the the source distribution. These examples, and the resulting plots, may also be found at <http://gnuplot.sourceforge.net/demo/>.

## 2.7 How do I modify gnuplot, and apply patches?

For this, you will need to recompile **gnuplot**.

Modifications people make are either done by replacing files, such as terminal drivers, or by patching. If a file is a replacement, it will probably tell you in its README or in the lines at the beginning.

To patch a file, you need the *patch* utility, and possibly also the *automake* and *autoconf* tools. A typical command for applying a patch is `patch -p0 <newfunctionality.diff`.

There is repository of contributed patches in the "Patches" section on gnuplot's sourceforge site <http://sourceforge.net/p/gnuplot/patches/>.

## 2.8 How do I determine which options are compiled into gnuplot?

Given that you have a compiled version of **gnuplot**, you can use the `show` command to display the list of compile options (a.k.a. compilation options, or build options) that were used to build your copy.

```
gnuplot> show version long
```

# 3 Working with it.

## 3.1 How do I get help?

Read this document.

Give the `help` command at the initial prompt. After that, keep looking through the keywords. Good starting points are `plot` and `set`.

Read the manual, if you have it.

Look through the demo subdirectory; it should give you some ideas.

Ask your colleagues, the system administrator or the person who set up **gnuplot**.

If all these fail, please upgrade to the newest version of **gnuplot** or urge your system-administrator to do so. Then post a question to <news://comp.graphics.apps.gnuplot> or send mail to the gatewayed mailing list <mailto://gnuplot-info@lists.sourceforge.net>. Please note that, due to the overwhelming amount of spam it would otherwise receive, you have to subscribe before you can post to it. Subscription instructions are in the main gnuplot manual. Do not forget to cite the version number and the operating system. If you want to subscribe to the mailing list, visit the URL <http://lists.sourceforge.net/lists/listinfo/gnuplot-info>. But please don't use the mailing list if you can read <news://comp.graphics.apps.gnuplot> directly. If you post a question there, it is considered good form to solicit e-mail replies and post a summary.

## 3.2 How do I print out my graphs?

The kind of output produced is determined by the `set terminal` command; for example, `set terminal postscript` will produce the graph in PostScript format. Output can be redirected using the `set output` command.

As an example, the following first plots a graph of  $\sin(x)$  to the screen and then redraws that same plot as a PostScript output file.

```
gnuplot> plot [-6:6] sin(x)
gnuplot> set terminal postscript
Terminal type set to 'postscript'
Options are 'landscape monochrome "Courier" 14'
gnuplot> set output "sin.ps"
gnuplot> replot
gnuplot> unset output           # set output back to default
gnuplot> unset terminal         # ditto for terminal type
gnuplot> system("print sin.ps") # print PS File (site dependent)
gnuplot>
```

Using the platform-independent way of restoring terminal by `set term push/pop` commands, do it by

```

gnuplot> set terminal postscript eps color lw 15 "Helvetica" 20
gnuplot> set out 'a.eps'
gnuplot> replot
gnuplot> set term pop

```

The command `set term pop` without a previous corresponding `set term push` switches the terminal back to the startup terminal. The same thing happens if you say `unset terminal`.

Some interactive terminal types (`win`, `wxt`, `qt`) provide a printer icon on the terminal's toolbar. This tool prints the current plot or saves it to file using generic system tools rather than by using a different gnuplot terminal type. That is, the file you get by selecting "save to png" in the print menu will be different than the file you get from `set term png; replot;`.

### 3.3 How do I include my graphs in <word processor>?

Basically, you save your plot to a file in a format your word processor can understand (using `set term` and `set output`, see above), and then you read in the plot from your word processor. Vector formats (PostScript, emf, svg, pdf,  $\TeX$ ,  $\LaTeX$ , etc) should be preferred, as you can scale your graph later to the right size.

Details depend on which word processor you use; use `set term` to get a list of available file formats.

Many word processors can use Encapsulated PostScript (\*.eps) for graphs. You can select eps output in **gnuplot** using the `set terminal postscript eps` command. Note that it is a good idea to check and correct the bounding box of the graphs in the eps files (manually or by the `fixbb` script from gnuplot webpage), as you have to correct this box for any eps figure produced by whichever program. Some (most?) word processors do not preview the actual image in the eps file, and you have to add the preview image yourself. You can use the GSView viewer for this (available for OS/2, Windows and X11), or some Unix ps tool.

Some Windows office applications, including OpenOffice.org, can handle vector images in EMF format. These can be either produced by the emf terminal, or by selecting 'Save as EMF...' from the toolbar of the graph window of the windows terminal.

OpenOffice.org can also read SVG, as well as AutoCAD's dxf format.

There are many ways to use gnuplot to produce graphs for inclusion in a  $\TeX$  or  $\LaTeX$  document. Some terminals produce \*.tex fragments for direct inclusion; others produce \*.eps, \*.pdf, \*.png output to be included using the `\includegraphics` command. The epslatex and cairolatex terminals produce both a graphics file (\*.eps or \*.pdf) and a \*.tex document file that refers to it. The tikz terminal produces full text and graphics to a pdf file when the output is processed with pdflatex.

Most word processors can import bitmap images (png, pbm, etc). The disadvantage of this approach is that the resolution of your plot is limited by the size of the plot at the time it is generated by gnuplot, which is generally a much lower resolution than the document will eventually be printed in.

The mif terminal type produces output for FrameMaker.

### 3.4 How do I edit or post-process a gnuplot graph?

This depends on the terminal type you use.

- **svg** terminal (scalable vector graphics) output can be further edited by a svg editor, e.g. **Inkscape** (<http://www.inkscape.org>), **Skencil** (<http://www.skencil.org>) or **Dia** (<http://projects.gnome.org/dia/>), or loaded into **OpenOffice.org** with an on-fly conversion into OO.o Draw primitives.
- PostScript or PDF output can be edited directly by tools such as Adobe Illustrator or Acrobat, or can be converted to a variety of other editable vector formats by the **pstoedit** package. Pstoedit is available at <http://www.pstoedit.net>.
- The mif terminal type produces an editable FrameMaker document.
- The DXF format is the AutoCAD's format, editable by several other applications.
- Bitmapped graphics (e.g. png, jpeg, pbm) can be edited using tools such as ImageMagick or Gimp. In general, you should use a vector graphics program to post-process vector graphic formats, and a pixel-based editing program to post-process pixel graphics.

### 3.5 How do I change symbol size, line thickness and the like?

Gnuplot offers a variety of commands to set line and point properties, including color, thickness, point shape, etc. The command `test` will display a test page for the currently selected terminal type showing the available pre-defined combinations of color, size, shape, etc. You can use the command `set linetype` to change this or define additional combinations.

### 3.6 Can I animate my graphs?

Only one **gnuplot** terminal type (`gif`) directly outputs an animated file:

```
set terminal gif animate {delay <time>} {loop <N>} {optimize}
```

Have a look at <http://gnuplot.sourceforge.net/demo/animate.html> in the demo collection.

### 3.7 How do I plot implicit defined graphs?

Implicit graphs or curves cannot be plotted directly in **gnuplot**. However there is a workaround.

```
gnuplot> # An example. Place your definition in the following line:
gnuplot> f(x,y) = y - x**2 / tan(y)
gnuplot> set contour base
gnuplot> set cntrparam levels discrete 0.0
gnuplot> unset surface
gnuplot> set table 'curve.dat'
gnuplot> splot f(x,y)
gnuplot> unset table
gnuplot> plot 'curve.dat' w l
```

The trick is to draw the single contour line  $z=0$  of the surface  $z=f(x,y)$ , and store the resulting contour curve to a **gnuplot** datafile.

### 3.8 How to fill an area between two curves

A plot with filled area between two given curves can be easily obtained using the pseudo file '+' with *filled-curves closed*. The example below demonstrates this for two curves  $f(x)$  and  $g(x)$ :

```
f(x)=cos(x)
g(x)=sin(x)
xmax=pi/4
set xrange [0:xmax]
plot '+' using 1:(f($1):(g($1)) with filledcurves closed
```

Note that the above code fills area between the two curves, not area satisfying inequality  $g(x)<f(x)$ . If you want the latter, you should use the ternary operator in *using* statement to return an undefined value (0/0) if the inequality is not satisfied.

See the documentation for *help filledcurves*, *help special-filenames*, and *help ternary* and see *fillbetween.dem* in the *demos* directory.

### 3.9 Pm3d plot from a datafile does not draw anything

You do `set pm3d; splot 'a.dat'` and no plot but colorbox appears. Perhaps there is no blank line in between two subsequent scans (isolines) in the data file? Add blank lines! If you are curious what this means, then don't hesitate to look to files like `demo/glass.dat` or `demo/triangle.dat` in the `gnuplot demo` directory.

You can find useful the following `awk` script (call it e.g. `addblanks.awk`) which adds blank lines to a data file whenever number in the first column changes:

```
/^[[:blank:]]*#/ {next} # ignore comments (lines starting with #)
NF < 3 {next} # ignore lines which don't have at least 3 columns
$1 != prev {printf "\n"; prev=$1} # print blank line
{print} # print the line
```



Then, either preprocess your data file by command `awk -f addblanks.awk <a.dat` or plot the datafile under a unixish platform by `gnuplot> splot "<awk -f addblanks.awk a.dat"`.

### 3.10 Drawing 2D projection of 3D data

Use *set view map* There are also plotting styles with `image` and with `rgbimage` for plotting 2D color images.

### 3.11 How to overlay dots/points scatter plot onto a pm3d map/surface

Use the explicit (see also implicit) switch of the `pm3d` style:

```
gnuplot> set pm3d explicit
gnuplot> splot x with pm3d, x*y with points
```

### 3.12 How to produce labeled contours

Labeling individual contours in a contour plot required special tricks and extra processing steps in **gnuplot** version 4. See <http://gnuplot.sourceforge.net/scripts/index.html#tricks-here>.

In version 5 the procedure is much simpler. Plot the contours twice, once "with lines" and once "with labels". To make the labels stand out it may help to use

```
set style textbox opaque noborder
set contours
splot 'DATA' with line, 'DATA' with labels boxed
```

### 3.13 How to overlay contour plot over pm3d map/surface

[Not needed any more?] This requires you to write contours into a temporary file using the table terminal, and then use this file in the final drawing without `set contours`. The following example demonstrates this for a map; for surface, remove `set pm3d map` and put `set ticslevel 0`.

```
# Write contours of function x*x-y*y to a (temporary) file
set contour base; set cntrparam level 20
unset surface
set table 'contour.dat'
splot x*x-y*y
unset table

# Change single blank lines to double blank lines
!awk "NF<2{printf\"\n\n"}{print}" <contour.dat >contour1.dat

# Draw the plot
reset
set palette gray
set palette gamma 2.5
set pm3d map
set pm3d explicit
splot x*x+y*y with pm3d, 'contour1.dat' with line lt -1
!rm contour.dat contour1.dat
```

The last command deletes the two temporary files.

### 3.14 Color facets with pm3d

It is possible to draw colors facets of a 3D objects, organized in such a file:

```
# triangle 1
x0 y0 z0 <c0>
x1 y1 z1 <c1>
```

```
x2 y2 z2 <c2>
x2 y2 z2 <c2>
```

```
# triangle 2
x y z
...
```

Notice the positioning single and double blank line. `<c>` is an optional color. Then plot it by (either of `splot`'s):

```
set pm3d
set style data pm3d
splot 'facets.dat'
splot 'facets_with_color.dat' using 1:2:3:4
```

Note that you avoid surface lines by *set style data pm3d* or *splot ... with pm3d*.

In the above example, `pm3d` displays triangles as independent surfaces. They are plotted one surface after another, as found in the data file. Parts overlapping in 2D projection are overdrawn.

Gnuplot is not 3D modeling program. Its hidden routines apply for points and lines, but not for faces. Without handling the data as a collection of faces, there would be no surface anything could be hidden behind. The 'hidden3d' algorithm works by using the input data in two ways: first, to set up a collection of triangles (made from a mesh of quadrangles) that form the surface, second as a collection of edges. It then goes through all those edges, checking what parts of them are not hidden behind any faces, and draws those.

Consequently, gnuplot won't draw your surface or 3D object as a virtual reality. It works OK for *set pm3d map* but for true 3D you would be probably more happy writing a converter of your facets into a VRML file.

### 3.15 Palette for printing my color map on color as well as black&white printer?

Try *set palette cubehelix*.

## 4 Wanted features

### 4.1 What's new in gnuplot 4.6, 5.0 etc?

Too many things to list here. Please refer to the *NEWS* file in the source distribution, or the "New features" section in the gnuplot documentation.

### 4.2 Does gnuplot support a driver for <graphics format>?

To see a list of the available graphic drivers for your installation of **gnuplot**, type `set term`.

Some graphics drivers are included in the normal distribution but are not built by default. If you want to use them, you'll have to change file `gnuplot/src/term.h`, and recompile.

### 4.3 Does gnuplot have hidden line removal?

Yes.

### 4.4 Does gnuplot support bar-charts/histograms/boxes?

**Gnuplot** supports various clustered and stacked histogram styles to display pretabulated data. It also offers a few options for accumulating raw data into bins, which can in turn be displayed as a bar chart. See the documentation for *smooth frequency*.

#### 4.5 Does gnuplot support pie charts? quarterly time charts?

Pie charts are sort of difficult in **gnuplot**, but see <http://gnuplot.sourceforge.net/demo/circles.html>, or have a look at <http://gnuplot-tricks.blogspot.com/2009/08/pie-charts-entirely-in-gnuplot.html>.

The demo collection contains an example of a simple Gantt chart. This capability is new in **gnuplot** version 5.

#### 4.6 Can I put multiple pages on one page?

Yes. `set multiplot`.

#### 4.7 Does gnuplot support multiple y-axes on a single plot?

Yes. 2D plots can have separate x axes at the bottom (x1) and top (x2), and separate y axes at the left (y1) and right (y2). Version 5 offers a plot mode *with parallelaxes* that allows additional y axes to be defined.

#### 4.8 Can I put both commands and data into a single file?

Version 5 supports named blocks of data in "here document" format:

```
gnuplot> $DATABLOCK << EOD
cats 4 2
dogs 1 4
EOD
gnuplot> plot $DATABLOCK using 2:3:1 with labels
```

Once the named block has been defined, it can be used as many times as you like.

Data can also be provided in-line as part of a plot command using the pseudo-file `plot "-"`. In this case the data can only be used once.

```
gnuplot> plot "-"
1 1
2 4
3 9
e
```

#### 4.9 Can I put Greek letters and super/subscripts into my labels?

Most terminal types (output device drivers) support an "enhanced text" mode. This lets you use sub- and superscripts. It also allows to use Greek letters and math symbols to the extent that these are supported by the fonts installed on your system. The simplest way to enter special characters of any sort, if your system supports it, is to select UTF-8 encoding. This obviates the need to change fonts.

The various  $\LaTeX$  terminal types (*latex*, *epslatex*, *tikz*, *context*, *cairolatex*) hand off text generation to  $\LaTeX$ . In this case you can use normal  $\LaTeX$  markup like `"\alpha_{3}"` or `'\alpha_{3}'`.

#### 4.10 How do I include accented characters

To obtain accented characters like `ü` or `ň` in your labels you should use 8bit character codes together with the appropriate encoding option. See the following example:

```
gnuplot> set encoding iso_8859_1
gnuplot> set title "M\374nchner Bierverbrauch \374ber die Jahre"
gnuplot> plot "bier.dat" u 1:2
```

Consequently, you can type labels in Czech, French, Hungarian, Russian... by means of an appropriate *set encoding*. However, you cannot mix two encodings in one file (e.g. accents for west and east latin encodings).

A more general solution is to use UTF-8 encoded fonts, and type the UTF-8 characters directly into gnuplot. This works for many terminal types but is very cumbersome for PostScript.

#### 4.11 Can I do 1:1 scaling of axes?

Try `set size square` or `set view equal xy`.

#### 4.12 Can I put different text sizes into my plots?

Most terminal types allow you to specify a starting font face and size. The "enhanced text" mode allows you to change fonts, text sizes, bold and italic styles within a plot.

#### 4.13 How do I skip data points?

By specifying `?` as a data value, as in

```
1 2
2 3
3 ?
4 5
```

See also `set missing`. See also `set datafile commentschars` for specifying comment characters in data files.

#### 4.14 How do I plot every nth point?

This can be specified with various options for the command `plot`, for example `plot 'a.dat' every 2`. If you want to draw a line through every point but only draw a point symbol at every nth point, then try `plot 'a.dat' with linespoints pointinterval n`.

#### 4.15 How do I plot a vertical line?

Depending on context, the main methods are:

- `set arrow .... nohead` where you have to compute explicitly the start and the end of the arrow.
- generate (inlined) datapoints and plot them

#### 4.16 How do I plot data files

Easily: by a command `plot 'a.dat'`. In 3D, use `splot 'a.dat'` – but don't forget to put a blank line in between two subsequent scans (isolines), otherwise you will get an error that the data is not gridded; see also question 3.9. If your data are not gridded, then use `set dgrid3d {many options}`.

#### 4.17 How do I replot multiplot drawing

You cannot directly: `gnuplot` supports `replot` command, not `remultiplot`. You have to write the complete sequence of commands since `set multiplot` till `unset multiplot` into a script file. Then you can load the script into `gnuplot` as many times as you need for replotting the drawing to different terminals or output files.

## 5 Miscellaneous

### 5.1 I've found a bug, what do I do?

First, try to see whether it actually is a bug, or whether it is a feature which may be turned off by some obscure `set`-command.

Next, see whether you have an old version of **gnuplot**; if you do, chances are the bug has been fixed in a newer release.

The CVS development version may already contain fixes for bugs reported since the release of the current version. Before submitting a bug report, please check whether the bug in question has already been fixed.

If, after checking these things, you still are convinced that there is a bug, proceed as follows. If you have a fairly general sort of bug report, posting to [news://comp.graphics.apps.gnuplot](mailto:news://comp.graphics.apps.gnuplot) is probably the way to go. If you have investigated a problem in detail, especially if you can provide a simple script that reproduces the error, please upload it to the bug-tracker at <http://sourceforge.net/p/gnuplot/bugs>.

The tracker on sourceforge is for reporting bugs and collecting bug fixes that will appear in a subsequent release. The [news://comp.graphics.apps.gnuplot](mailto:news://comp.graphics.apps.gnuplot) newsgroup will be more help for finding work arounds or actually solving **gnuplot** related problems. If you do send in a bug report, be sure and include the version of **gnuplot** (including patchlevel) as shown by the command `show version long`, terminal driver, operating system, an exact description of the bug and input which can reproduce the bug. Failure to indicate these details can render a solution to your problem almost impossible. Also, any context diffs should be referenced against the latest official version of **gnuplot** if at all possible.

## 5.2 Can I use gnuplot routines for my own programs?

On systems supporting pipes, you can pipe commands to **gnuplot** from other programs. Many applications with **gnuplot** as the graphics engine, like Octave (<http://www.octave.org>), uses this method. This also works from a cgi script to drive **gnuplot** from a forms-based web page.

John Campbell ([mailto://jdc@nauvax.ucc.nau.edu](mailto:jdc@nauvax.ucc.nau.edu)) modified a much earlier version of **gnuplot** (3.5) to be a library of C subroutines callable from a C program. Gnuplot itself has changed radically since then, and we are not aware of any plans to create a similar library based on the current version.

## 5.3 What extensions have people made to gnuplot? Where can I get them?

Extensions or patches are available on the "Patches" page of the **gnuplot** development site <http://sourceforge.net/p/gnuplot/patches/>. The current development version will generally include features that are not yet part of the most recent official release of **gnuplot**.

## 5.4 I need an integration, fft, iir-filter,...!

**Gnuplot** has been and is a plotting program, not a data processing or mathematical program suite. Therefore **gnuplot** can't do that. Look into the demo file "bivariat.dem" for a basic implementation of an integration. However **gnuplot** version 5 does support calling functions from a dynamically loaded external shared object, i.e. a plugin. So if you want to code up some complicated function in C or another language with compatible calling conventions, you can compile it into a plugin for **gnuplot** to import.

## 5.5 Can I do heavy-duty data processing with gnuplot? or What is beyond gnuplot?

**gnuplot** by itself is not suited very well for heavy numerical computation. On the other hand it can handle very large data sets.

One thing you might try is *fudgit*, an interactive multi-purpose fitting program written by Martin-D. Lacasse. It can use **gnuplot** as its graphics back end.

You might also want to look at the applications developed by the Software Tools Group (STG) at the National Center for Supercomputing Applications <http://ncsa.uiuc.edu>.

You can also try *pgperl*, an integration of the PGPLOT plotting package with Perl 5. Information can be found at <http://www.ast.cam.ac.uk/AAO/local/www/kgb/pgperl>, the source is available from <ftp://ast.cam.ac.uk/pub/kgb/pgperl/> or [linux.nrao.edu/pub/packages/pgperl/](http://linux.nrao.edu/pub/packages/pgperl/).

Another possibility is **Octave**. To quote from its README: Octave is a high-level language, primarily intended for numerical computations. Octave is licensed under GPL, and in principle, it is a free Matlab clone. It provides a convenient command line interface for solving linear and nonlinear problems numerically. The latest released version of Octave is always available from <http://www.octave.org>. By the way, octave uses **gnuplot** as its plotting engine, so you get a data-processing program on top of **gnuplot**.

Finally there is *scilab* (<http://www.scilab.org>), an open source alternative to *matlab*.

## 5.6 How to use hotkeys in my interactive terminals

Most of the interactive terminals support both pre-defined and user-defined hotkeys to replot, toggle plot elements, change axis scaling, and so on. Hit *h* in an active gnuplot plot window to get list of hotkeys. Read *help mouse* and *help bind* for more information.

## 5.7 I have ported gnuplot to another system, or patched it. What do I do?

The preferred way of submitting, commenting and upgrading patches is via 'Patches' section on <http://sourceforge.net/p/gnuplot/patches/>. You may want to send a note to <mailto://gnuplot-beta@lists.sourceforge.net> for more lively discussion.

## 5.8 I want to help in developing the next version of gnuplot. What can I do?

Join the **gnuplot** beta test mailing list by sending a mail containing the line `subscribe gnuplot-beta` in the body (not the subject) of the mail to <mailto://Majordomo@lists.sourceforge.net>.

## 5.9 Open questions for inclusion into the FAQ?

Please submit your questions (along with the answer) to <mailto://gnuplot-beta@lists.sourceforge.net>.

# 6 Making life easier

## 6.1 How do I plot two functions in non-overlapping regions?

This used to be complicated. In version 5 it is easy - place the desired range immediately before each function being plotted. For example, to plot experimental data and two different functional models *f1* and *f2* covering two different portions of the domain:

```
gnuplot> set autoscale x # get x range from the data
gnuplot> plot "data", [-100:0] f1(x), [0:100] f2(x)
```

## 6.2 How do I run my data through a filter before plotting?

If your system supports the `popen()` function, as Unix does, you should be able to run the output through another process, for example a short `awk` program, such as

```
gnuplot> plot "< awk ' { print $1, $3/$2 } ' file.in"
```

The `plot` command is very powerful and is able to do some arithmetic on datafiles. See `help plot`.

The above filtering works seamlessly under Unixes and OS/2. On Windows, this is only supported by default in *gnuplot* version 5, but required a separate program *wgnuplot\_pipes* in some earlier versions.

## 6.3 How do I save and restore my current settings?

Use the `save` and `load` commands for this; see `help save` and `help load` for details.

You can save the current terminal and restore it later without touching the filesystem by `set term push` and `set term pop`, respectively.

## 6.4 How do I plot lines (not grids) using `splot`?

If the data input to `splot` is arranged such that each line contains the same number of data points (using blank lines as delimiters, as usual), `splot` will by default treat the data as describing a surface. If you want to draw individual lines instead, try some combination of `unset surface`, `set surface explicit`, `plot ... nosurface`.

## 6.5 How do I plot a function $f(x,y)$ that is bounded by other functions in the $x$ - $y$ plane?

Here is one way:

```
gnuplot> f(x,y) = x**2 + y **2
gnuplot> x(u) = 3*u
gnuplot> yu(x) = x**2
gnuplot> yl(x) = -x**2
gnuplot> set parametric
gnuplot> set cont
gnuplot> splot [0:1] [0:1] u,yl(x(u))+(yu(x(u)) - yl(x(u)))*v,\
> f(x(u), (yu(x(u)) - yl(x(u)))*v)
```

## 6.6 How do I call gnuplot from my own programs?

On unix-like systems, commands to gnuplot can be piped via stdin. Output from **gnuplot**'s `print` command can be read via a named pipe. On Windows, due to the lacking standard input (stdin) in GUI programs, you either need to use the console version *gnuplot* (recommended), or use *wgnuplot\_pipes*, which has an additional console window attached. The old helper program *pgnuplot* is still included in the distribution package.

## 6.7 What if I need h-bar (Planck's constant)?

The most straightforward way is to use a UTF-8 font, and type in the  $\hbar$  character (Unicode code point #x210F) directly.

This does not work in PostScript, however, so you must use approximations like  $\textcircled{\scriptstyle h}$  or  $\textcircled{\scriptstyle h}$ . In the latter, the "h" (a long one in /Symbol) is non-spacing and 24-pt. The 14-pt "h" is offset by an 8-pt space (which is the space preceding the "\_") but smaller, since it's written as a subscript. But these don't look too much like the hbar we're used to, since the bar is horizontal instead of sloped. I don't see a way to get that. I tried using an accent (character 264 in iso-latin-1 encoding), but I haven't found a way to scale and position the pieces correctly. One more possibility would be  $\textcircled{\scriptstyle h}$ .

The reduced Planck's constant can be set very easily by using the AMS-LaTeX PostScript fonts which are available from <http://www.ams.org/tex/amsfonts.html> (also included in many LaTeX distributions). **Gnuplot** (see `help fontpath`) and the PostScript interpreter (usually Ghostscript) have to know where the file `msbm10.pfb` (or `msbm10.pfa`) resides. Use `{/MSBM10 \175}` to produce  $\hbar$  which is a "h" superimposed by a sloped bar. The standard  $\bar{h}$  (horizontal bar) has the octal code 176. Please note that h-bar exists only as an italic type.

## 6.8 What if I need the Solar mass symbol?

As with Planck's constant, the most straightforward way is to use a UTF-8 font, and type in the  $\odot$  character (Unicode code point #x2299 ; "circled dot operator") directly. The very similar glyph at code point #x2609 ; "sun" may be even better, but not many fonts have it.

## 6.9 How do I produce blank output page?

Well, you probably don't want a blank page, but page with a just a title (overprinting title in another graph in multiplot page):

```
reset; unset xtics; unset ytics
unset border; unset key
set title 'Title on an empty page'
plot [][0:1] 2
```

## 6.10 How do I give exact positions for the graph borders on the page?

Specify the position of the top, bottom, left, and right borders in terms of their fractional position within the page:

```
set lmargin at screen 0.05
set bmargin at screen 0.05
set rmargin at screen 0.95
set tmargin at screen 0.95
```

## 7 Common problems

### 7.1 Help! None of my fonts work.

Gnuplot does not do font handling by itself; it must necessarily leave that to the individual device support libraries. Unfortunately, this means that different terminal types need different help in finding fonts. Here are some quick hints. For more detailed information please see the gnuplot documentation for the specific terminal type you are having problems with.

**png/jpeg/gif** These terminal types use the libgd support library, which searches for fonts in the directories given in the environmental variable GDFONTPATH. Once you get libgd fontpaths sorted out, you will probably want to set a default font for gnuplot. For example: `setenv GNUPLOT_DEFAULT_GDFONT verdana`

**pdf** The libpdf support library should have come with an associated font configuration file, usually installed as `/usr/local/share/pdflib.upr`. The environmental variable PDFLIBRESOURCE should point to this file.

**post** PostScript font names are not resolved until the document is printed. Gnuplot does not know what fonts are available to your printer, so it will accept any font name you give it. However, it is possible to bundle a font with the gnuplot output; please see the instructions given by gnuplot's internal command "help set term post fontfile".

**svg** Font handling is viewer-dependent.

**x11** The x11 terminal uses the normal x11 font server mechanism. The only tricky bit is that in order to use multi-byte fonts you must explicitly say so:

```
set term x11 font "mbfont:sazanami mincho,vera,20"
```

**win** Select "Choose font..." from the "Options" pull-down menu in the toolbar.

**wxt, qt** On linux systems these terminals rely on fonts provided by the system's *fontconfig* utility.

### 7.2 Gnuplot does not open a plot window on VMS. Why?

On VMS, you need to make several symbols:

```
$ gnuplot_x11 ::= $disk:[directory]gnuplot_x11
$ gnuplot ::= $disk:[directory]gnuplot.exe
$ def/job GNUPLOT$HELP disk:[directory]gnuplot.hlb
```

Then run **gnuplot** from your command line, and use `set term x11`.

On Unix systems the x11 and qt terminals require installation of separate helper programs *gnuplot\_x11* and *gnuplot\_qt*. These are usually installed in a directory `/usr/libexec/gnuplot/5.0/` and **gnuplot** knows to look for them there. If they are installed somewhere else or gnuplot is having trouble finding them, try setting the environmental variable `GNUPLOT_DRIVER_DIR`.



### 7.3 Why does gnuplot ignore my very small numbers?

For some purposes **Gnuplot** treats numbers less than  $1e-08$  as being zero. Thus, if you are trying to plot a collection of very small numbers, they may be plotted as zero. Worse, if you're plotting on a log scale, they will be off scale. Or, if the whole set of numbers is "zero", your range may be considered empty:

```
gnuplot> plot 'test1'
Warning: empty y range [4.047e-19:3e-11], adjusting to [-1:1]
gnuplot> set yrange [4e-19:3e-11]
gnuplot> plot 'test1'
      ^
y range is less than `zero`
```

The solution is to change **gnuplot's** idea of "zero":

```
gnuplot> set zero 1e-20
```

For more information, type `help set zero`.

### 7.4 When I run gnuplot from the shell or a script, the resulting plot flashes by on the screen and then disappears

1. Put a pause `-1` after the plot command in the file, or at the file end.
2. Use command `gnuplot filename.gp -` (yes, dash is the last parameter) to stay in the interactive regime when the script completes.
3. Run **gnuplot** as `gnuplot -persist`
4. On Windows you can also use either `-persist` or `/noend`.
5. Give the `persist` option as part of the `set terminal` command.

### 7.5 My formulas (like 1/3) are giving me nonsense results! What's going on?

**Gnuplot** does integer, and not floating point, arithmetic on integer expressions. For example, the expression  $1/3$  evaluates to zero. If you want floating point expressions, supply trailing dots for your floating point numbers. Example:

```
gnuplot> print 1/3
      0
gnuplot> print 1./3.
      0.333333
```

This way of evaluating integer expressions is shared by both C and Fortran.

### 7.6 My output files are incomplete!

You may need to flush the output with a closing `set output`. Some output formats (postscript, pdf, latex, svg, ...) can include several pages of plots in a single output file. For these output modes, gnuplot leaves the file open after each plot so that you can add additional plots to it. The file is not completed and made available to external applications until you explicitly close it (`set output` or `unset output`), or select a different terminal type (`set term`) or exit gnuplot. Output formats that contain only a single 'page' (png, emf, ...) should not suffer from this problem.

### 7.7 When using the L<sup>A</sup>T<sub>E</sub>X-terminal, there is an error during the L<sup>A</sup>T<sub>E</sub>X-run!

The L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>-core no longer includes the commands "`\Diamond`" and "`\Box`"; they are included in the `latexsym` package. Other symbols are taken from the `amssymb` package. Both of these are part of the base distribution and thus part of any L<sup>A</sup>T<sub>E</sub>X implementation. Please remember to include these packages in your L<sup>A</sup>T<sub>E</sub>X document.

## 7.8 Calling gnuplot in a pipe or with a gnuplot-script doesn't produce a plot!

You can call **gnuplot** by using a short Perl-script like the following:

```
#!/usr/local/bin/perl -w
open (GP, "|/usr/local/bin/gnuplot -persist") or die "no gnuplot";
# force buffer to flush after each write
use FileHandle;
GP->autoflush(1);
print GP,"set term x11;plot '/tmp/data.dat' with lines\n";
close GP
```

**Gnuplot** closes its plot window on exit. The `close GP` command is executed, and the plot window is closed even before you have a chance to look at it.

There are three solutions to this: first, use the `pause -1` command in **gnuplot** before closing the pipe. Second, close the pipe only if you are sure that you don't need **gnuplot** and its plot window anymore. Last, you can use the command line option `-persist`: this option leaves the X-Window System plot window open.

## 8 Credits

**Gnuplot** 3.7's main contributors are (in alphabetical order) Hans-Bernhard Broecker, John Campbell, Robert Cunningham, David Denholm, Gershon Elber, Roger Fearick, Carsten Grammes, Lucas Hart, Lars Hecking, Thomas Koenig, David Kotz, Ed Kubaitis, Russell Lang, Alexander Lehmann, Alexander Mai, Carsten Steger, Tom Tkacik, Jos Van der Woude, James R. Van Zandt, and Alex Woo. Additional substantial contributors to version 4.0 include Ethan Merritt, Petr Mikulík and Johannes Zellner. Version 4.2, 4.4, 4.6 and 5.0 releases were coordinated by Ethan Merritt.

This list was initially compiled by John Fletcher with contributions from Russell Lang, John Campbell, David Kotz, Rob Cunningham, Daniel Lewart and Alex Woo. Reworked by Thomas Koenig from a draft by Alex Woo, with corrections and additions from Alex Woo, John Campbell, Russell Lang, David Kotz and many corrections from Daniel Lewart. Again reworked for **gnuplot** 3.7 by Alexander Mai and Juergen v.Hagen with corrections by Lars Hecking, Hans-Bernhard Broecker and others. Revised for **gnuplot** version 4 by Petr Mikulík and Ethan Merritt. Revised for **gnuplot** version 5 by Ethan Merritt.