

# Appendix C: 3D Graphing with Gnuplot

---

## Appendix C: 3D Graphing with Gnuplot

---

### 3D Graphing with Gnuplot

Gnuplot is a free, open-source software package for producing a variety of graphs. Versions are available for many operating systems. Below is a very brief tutorial on how to use Gnuplot to graph functions of several variables.

#### INSTALLATION

1. Go to <http://www.gnuplot.info/download.html> (<http://www.gnuplot.info/download.html>) and follow the links to download the latest version for your operating system. For Windows, you should get the Zip file with a name such as `gp420win32.zip`, which is version 4.2.0. All the examples we will discuss require at least version 4.2.0.
2. Install the downloaded file. For example, in Windows you would unzip the Zip file you downloaded in Step 1 into some folder (use the “Use folder names” option if extracting with WinZip).

#### RUNNING GNUPLOT

1. In Windows, run `wgnuplot.exe` from the folder (or `bin` folder) where you installed Gnuplot. In Linux, just type `gnuplot` in a terminal window.
2. You should now get a Gnuplot terminal with a `gnuplot>` command prompt. In Windows this will appear in a new window, while in Linux it will appear in the terminal window where the `gnuplot` command was run. For Windows, if the font is unreadable you can change it by right-clicking on the text part of the Gnuplot window and selecting the “Choose Font..” option. For example, the font “Courier”, style “Regular”, size “12” is usually a good choice (that choice can be saved for future sessions by right-clicking in the Gnuplot window again and selecting the option to update `wgnuplot.ini`).
3. At the `gnuplot>` command prompt you can now run graphing commands, which we will now describe.

#### GRAPHING FUNCTIONS

The usual way to create 3D graphs in Gnuplot is with the **splot** command:

```
splot <range> <comma-separated list of functions>
```

For a function  $z = f(x, y)$ , `<range>` is the range of  $x$  and  $y$  values (and optionally the range of  $z$  values) over which to plot. To specify an  $x$  range and a  $y$  range, use an expression of the form `[a : b][c : d]`, for some numbers  $a < b$  and  $c < d$ . This will cause the graph to be plotted for  $a \leq x \leq b$  and  $c \leq y \leq d$ .

Function definitions use the  $x$  and  $y$  variables in combination with mathematical operators, listed below:

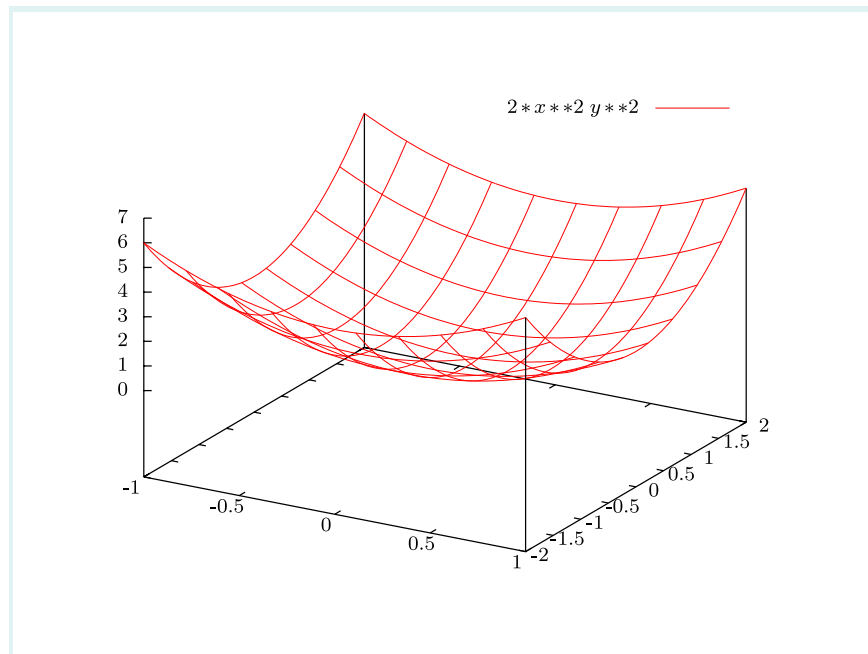
| Symbol    | Operation      | Example       | Result    |
|-----------|----------------|---------------|-----------|
| +         | Addition       | $2 + 3$       | 5         |
| −         | Subtraction    | $3 - 2$       | 1         |
| *         | Multiplication | $2 * 3$       | 6         |
| /         | Division       | $4 / 2$       | 2         |
| **        | Power          | $2 ** 3$      | $2^3 = 8$ |
| $\exp(x)$ | $e^x$          | $\exp(2)$     | $e^2$     |
| $\log(x)$ | $\ln x$        | $\log(2)$     | $\ln 2$   |
| $\sin(x)$ | $\sin x$       | $\sin(\pi/2)$ | 1         |
| $\cos(x)$ | $\cos x$       | $\cos(\pi)$   | −1        |
| $\tan(x)$ | $\tan x$       | $\tan(\pi/4)$ | 1         |

width height 0.5pt

**Example C.1.** To graph the function  $z = 2x^2 + y^2$  from  $x = -1$  to  $x = 1$  and from  $y = -2$  to  $y = 2$ , type this at the `gnuplot>` prompt:

```
splot [-1:1][-2:2] 2*x**2 + y**2
```

The result is shown below:



width height 0.5pt

Note that we had to type `2*x**2` to multiply 2 times  $x^2$ . For clarity, parentheses can be used to make sure the operations are being performed in the correct order:

```
splot [-1:1][-2:2] 2*(x**2) + y**2
```

In the above example, to also plot the function  $z = e^{x+y}$  on the same graph, put a comma after the first function then append the new function:

```
splot [-1:1][-2:2] 2*(x**2) + y**2, exp(x+y)
```

By default, the  $x$ -axis and  $y$ -axis are not shown in the graph. To display the axes, use this command *before* the **splot** command:

```
set zeroaxis
```

Also, by default the  $x$ - and  $y$ -axes are switched from their usual position. To show the axes with the orientation which we have used throughout the text, use this command:

```
set view 60,120,1,1
```

Also, to label the axes, use these commands:

```
set xlabel "x"
set ylabel "y"
set zlabel "z"
```

To show the level curves of the surface  $z = f(x, y)$  on both the surface and projected onto the  $xy$ -plane, use this command:

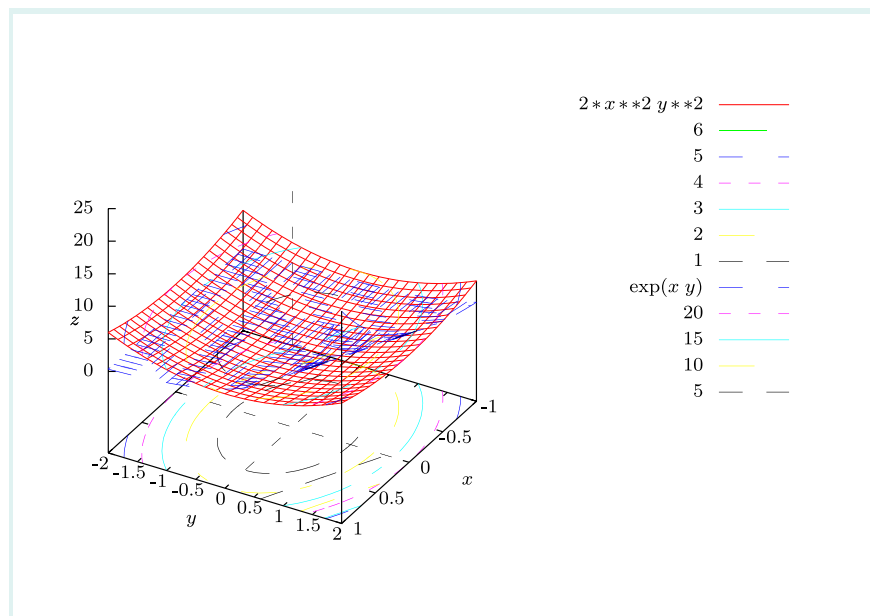
```
set contour both
```

The default mesh size for the grid on the surface is 10 units. To get more of a colored/shaded surface, increase the mesh size (to, say, 25) like this:

```
set isosamples 25
```

Putting all this together, we get the following graph with these commands:

|                                                           |
|-----------------------------------------------------------|
| <code>set zeroaxis</code>                                 |
| <code>set view 60,120,1,1</code>                          |
| <code>set xlabel "x"</code>                               |
| <code>set ylabel "y"</code>                               |
| <code>set zlabel "z"</code>                               |
| <code>set contour both</code>                             |
| <code>set isosamples 25</code>                            |
| <code>splot [-1:1][-2:2] 2*(x**2) + y**2, exp(x+y)</code> |



The numbers listed below the functions in the key in the upper right corner of the graph are the “levels” of the level curves of the corresponding surface. That is, they are the numbers  $c$  such that  $f(x, y) = c$ . Because of the large number of level curves, the key was put outside the graph with the **set key outside** command. If you do not want the function key displayed, it can be turned off with this command: **unset key**

### PARAMETRIC FUNCTIONS

Gnuplot has the ability to graph surfaces given in various parametric forms. For example, for a surface parametrized in cylindrical coordinates

$$x = r \cos \theta, \quad y = r \sin \theta, \quad z = z$$

you would do the following:

```
set mapping cylindrical
```

```
set parametric
```

```
splot [a:b][c:d] v*cos(u),v*sin(u),f(u,v)
```

where the variable  $u$  represents  $\theta$ , with  $a \leq u \leq b$ , the variable  $v$  represents  $r$ , with  $c \leq v \leq d$ , and  $z = f(u, v)$  is some function of  $u$  and  $v$ .

width height 0.5pt

**Example C.2.** The graph of the helicoid  $z = \theta$  in Example 1.34 from Section 1.7 (p. § 1.34) was created using the following commands:

|                                                      |
|------------------------------------------------------|
| <code>set mapping cylindrical</code>                 |
| <code>set parametric</code>                          |
| <code>set view 60,120,1,1</code>                     |
| <code>set xyplane 0</code>                           |
| <code>set xlabel "x"</code>                          |
| <code>set ylabel "y"</code>                          |
| <code>set zlabel "z"</code>                          |
| <code>unset key</code>                               |
| <code>set isosamples 15</code>                       |
| <code>splot [0:4*pi][0:2] v*cos(u),v*sin(u),u</code> |

The command **set xyplane 0** moves the  $z$ -axis so that  $z = 0$  aligns with the  $xy$ -plane (which is *not* the default in Gnuplot). Looking at the graph, you will see that  $r$  varies from 0 to 2, and  $\theta$  varies from 0 to  $4\pi$ .

width height 0.5pt

## PRINTING AND SAVING

In Windows, to print a graph from Gnuplot right-click on the titlebar of the graph's window, select "Options" and then the "Print.." option. If that does not work on your version of Gnuplot, then go to the File menu on the main Gnuplot menubar, select "Output Device ...", and enter `pdf` in the *Terminal type?* textfield, hit OK. That will allow you to print the graph as a PDF file.

To save a graph, say, as a PNG file, go to the File menu on the main Gnuplot menubar, select "Output Device ...", and enter `png` in the *Terminal type?* textfield, hit OK. Then, in the File menu again, select the "Output ..." option and enter a filename (say, `graph.png`) in the *Output filename?* textfield, hit OK. Now run your `splot` command again and you should see a file called `graph.png` in the current directory (usually the directory where `wgnuplot.exe` is located, though you can change that setting using the "Change Directory ..." option in the File menu).

In Linux, to save the graph as a file called `graph.png`, you would issue the following commands:

|                                     |
|-------------------------------------|
| <code>set terminal png</code>       |
| <code>set output 'graph.png'</code> |

and then run your splot command. There are many terminal types (which determine the output format). Run the command **set terminal** to see all the possible types. In Linux, the **postscript** terminal type is popular, since the print quality is high and there are many PostScript viewers available.

To quit Gnuplot, type **quit** at the `gnuplot>` command prompt.