A gnuplot Crash Course

Revision 2

BY KEVIN CROKER

Email: kcroker@hawaii.edu
Web: http://www.phys.hawaii.edu/~kcroker

Abstract

gnuplot is a Freely available professional tool for representing data graphically used widely in academia and industry. The following pages summarize how to use gnuplot immediately to graph experimental data with errors and perform a linear fit on the data. As this is a crash course, references on the Internet and within gnuplot itself are given for those wishing to flush out their knowledge.

1 Very Basics

Getting gnuplot. This is the trickiest part of the whole thing! Depending on what operating system you use, you will need to get gnuplot differently.

Windows. Use your favorite browser to go to http://sourceforge.net/projects/gnuplot. Click on the big green download button. This will take you to a different page. Scroll down until you see the heading File Releases. Under there, click on the file **gp422win32.zip**. Unzip this file onto your desktop, it will make a folder called gnuplot. To run the program, open this folder, then open the folder bin. Double click on **wgnuplot.exe**.

GNU/Linux. You should be able to use whatever package manager you normally use to add software to find and install gnuplot. Just search for gnuplot. And they say GNU/Linux is harder to use... :)

MacOS X. You have to install two seperate things for gnuplot to work on your Mac.

Warning 1. You must install Aqua before installing gnuplot

1. Go here http://sourceforge.net/projects/aquaterm/ and get Aquaterm. Aquaterm lets gnuplot draw on the MacOS screen. Click on the big green download button and then click on the small green download button. Install this as you normally would.
2. Go here http://naranja.umh.es/~atg/incubator.html and get the correct installer under Gnuplot 4.2. You should be able to find out whether you need the Intel binary or the PowerPC binary by going to the Apple at the top menu and clicking on About this Mac. Make sure you click on the link that says "binary installer package." Install this as you normally would.
3. To run gnuplot:
   a) Goto Applications/Utilities and start Terminal
   b) Goto File and select "Open New Shell" You will now have two boxes to type things in
   c) Choose one box, and type

      ```
      /usr/local/bin/gnuplot
      ```

   d) In the other box, type

```
nano
```

e) Enter your data sets into the nano box, save is called "Write Out" Note that ^O means Ctrl+O

Preparing Data. If you wish to plot actual data points you have taken, create a new text file with your favorite editor (notepad in Windows). gnuplot reads data from your file one line at a time. Each line in our example specifies one data point: independant variable value (here, time), a dependent variable value (here, distance in the y direction), and an uncertainty in our dependent variable. Separate the three values with a [Space] or [Tab].

Example 2. mydata.dat

```
# mydata.dat
# gnuplot ignores lines that start with #
#
# t  y      uncertainty in y
# ------------------------------------------------
0    0      0.001
1    0.25   0.1
2    0.5    0.05
3    0.75   0.4
4    1.25   0.2
5    1.30   0.3
6    1.55   0.33
7    1.80   0.1
8    2.05   0.5
```

Plotting Data. Start up gnuplot.

Warning 3. If you are using Windows, first use the ChDir button at the top the move to the folder where you saved your data.

After some version and author information, you will see a prompt (a place to type things). It will look like this:

```
gnuplot> _
```

Note 4. There are some very important things the prompt does for you to save you lots of time. **You never have to type the same thing twice**, you can just press the Up Arrow. Up and Down arrows will scroll through everything you've ever typed and when you find what you want, you can move around using the Left and Right Arrows, Home and End keys, and make whatever changes you want to. You can press Enter anywhere in the line, and it will read the entire line. Never type a really long command more than once!

Let's get down to brass tacks and plot our data.

Warning 5. These are double quotes, not single quotes (apostrophes). If you just copy this text, it will copy as two single quotes, which will confuse gnuplot!

Warning 6. Windows is stupid and likes to hide the extensions of files it "knows." It may have saved your file as filename.dat.txt or filename.txt or filename.txt.txt.

```
gnuplot> plot ‘‘mydata.dat’’
```

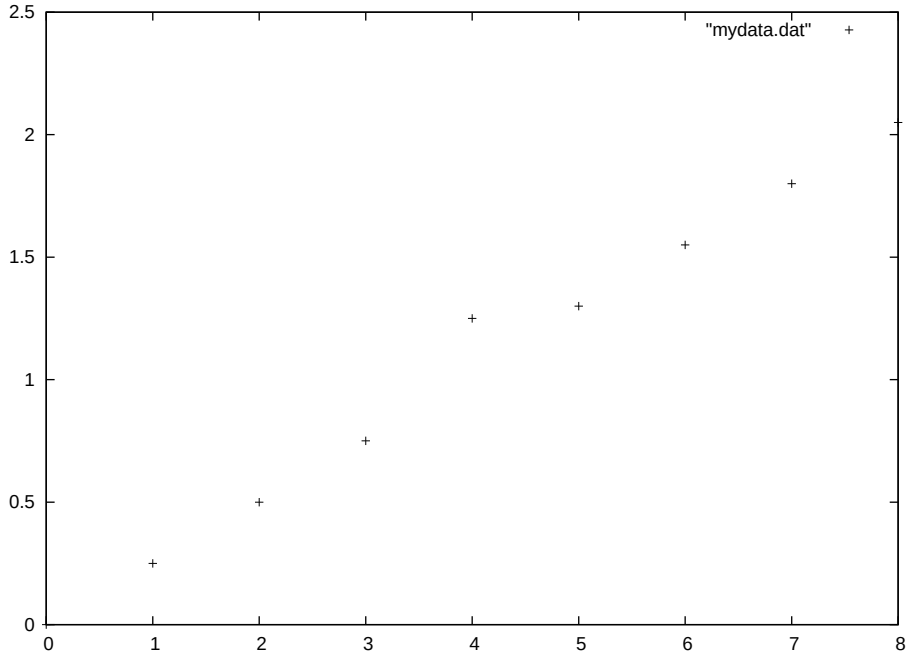

Figure 1.

Pretty bland! But it has our points, it automatically scaled our axes for us, and it automatically made a legend for us. (Your legend will have a different name than mine, but that's okay.) But wait, the dots are really hard to see, and where are our errors? We have to explicitly tell gnuplot to draw our errors, so while we're at it, let's use bigger dots too:

```
gnuplot> plot ‘‘mydata.dat’’ with errorbars pointtype 7 pointsize 2
```

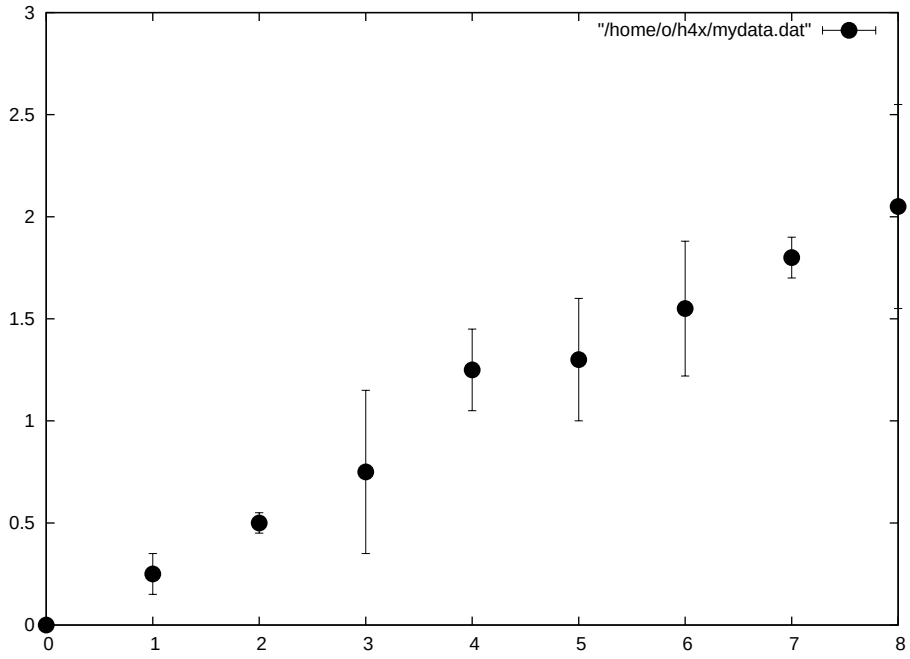

Figure 2.

Excellent! You can play with the number after "pointtype" to choose different shapes and you can make the points bigger or smaller by changing the number after "pointsize".

Fitting our Data. We usually want to fit against a line, that is to fit $y = m\,x + b$ to our data. To do this, type:

```
gnuplot> f(x) = m*x + b
gnuplot> fit f(x) ‘‘mydata.dat’’ using 1:2:3 via m,b
```

Note 7. gnuplot will print out a bunch of stuff as it figures out how to fit your graph. If it works you will see something like this near the end

```
Final set of parameters
================================
m = 0.258333 +/- 0.01045 (4.044%)
b = 0.0166667 +/- 0.04973 (298.4%)
```

If you have a zero in one of your uncertainties, you will run into problems. gnuplot will spit out things saying “nan” (not a number) because it has tried to divide by your zero. *You shouldn’t have zero uncertainty in any of your measurements*! But if you do for some reason, you must help gnuplot out with an initial guess.

```
gnuplot> m = 1
gnuplot> b = 1
```

Now repeat the fit command and things should work.

If our fit has worked, it will have found m and b for us! So, lets add the fit to our existing graph:

```
gnuplot> replot f(x)
```

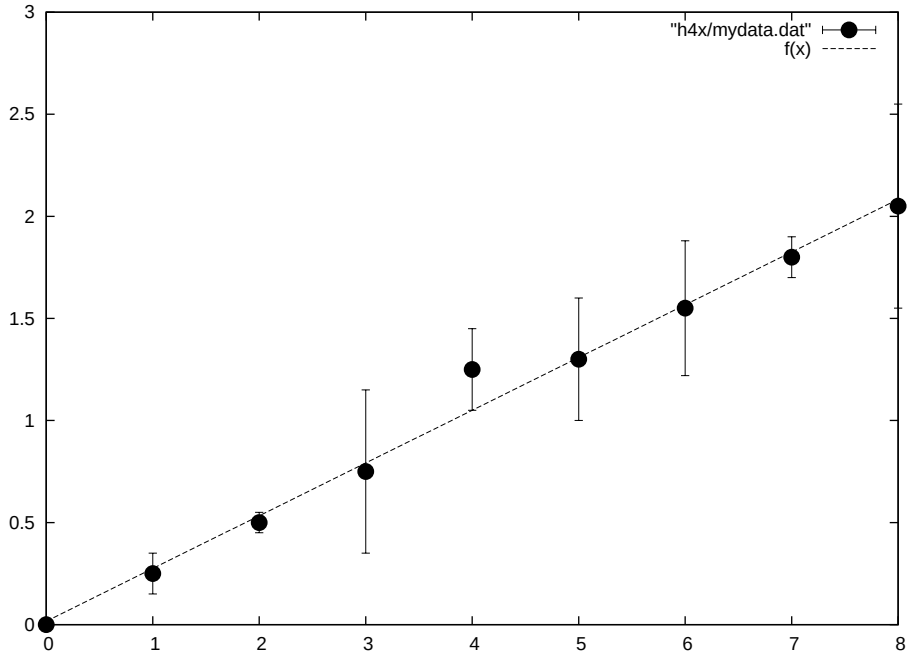

Figure 3.

Notice that to update or add something to the current graph, we use “replot” instead of “plot”.

Adding Text. We are almost done! All we have to add are labels to our axes and a title. We can label our axes and add a title all at once. Remember to specify your units!

```
gnuplot> set ylabel ‘‘displacement (m)’’
```

```
gnuplot> set xlabel ‘‘time (s)’’
gnuplot> set title ‘‘Distance Gone by Cake per Time’’
gnuplot> replot
```

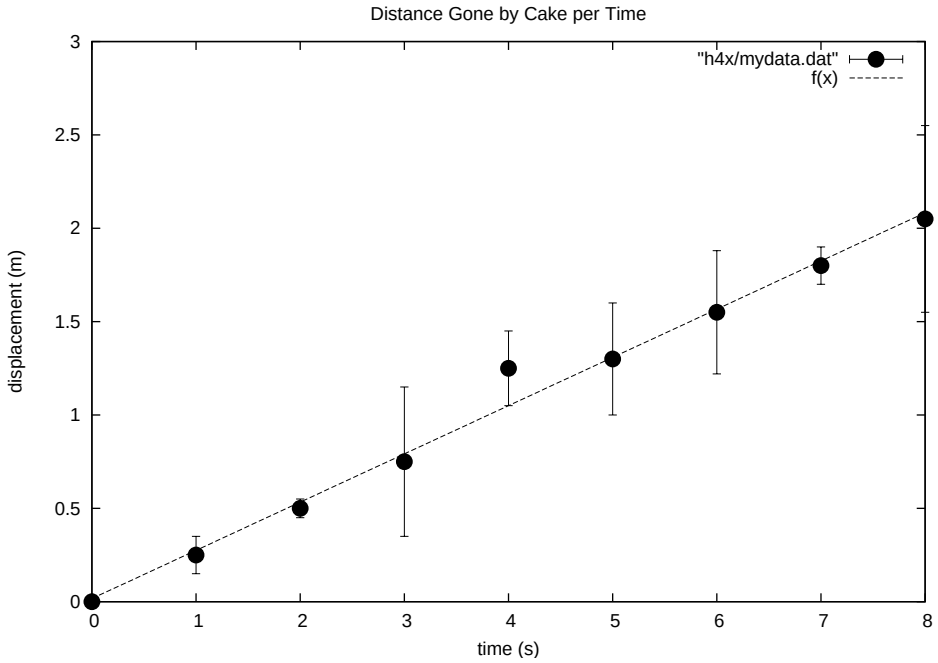

Figure 4.

Saving your Graph. So far, gnuplot has been drawing your graph on the screen. You have to explicitly tell gnuplot to draw your graph somewhere else, like an image file. To do this

```
gnuplot> set term gif
gnuplot> set output ‘‘mygraph.gif’’
gnuplot> replot
```

This will save your graph in mygraph.gif. You can now load this file into your browser of choice and print it from there, or you could insert this image into a word-processor document, etc.

If you want to continue graphing, you must tell gnuplot to start drawing back to the screen again!

```
[if you have Windows] gnuplot> set term windows
[if you have Mac] gnuplot> set term aqua
[if you have Linux] gnuplot> set term x11
```

Remark 8. If you are on Mac, you will not be able to save GIF. However, saving is really easy as in the Aqua window of your graph, you can goto File/Save and save it as a PDF without having to change the term in gnuplot.

2 References

Because gnuplot is Free software, there are numerous other tutorials, references, guides, and mailing lists that can help you learn to accomplish more complex things. Just to show off, take a look at this:

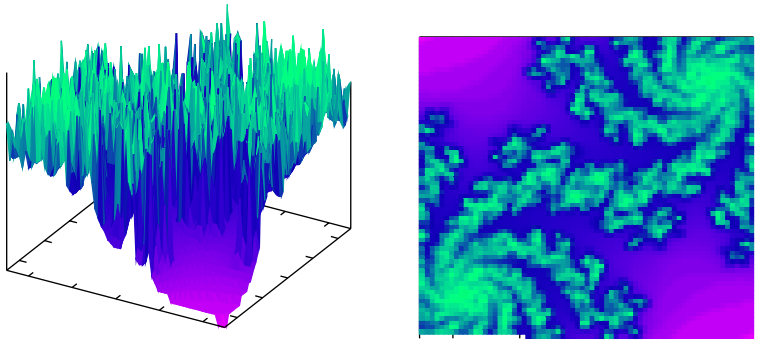

Figure 5. http://ayapin.film.s.dendai.ac.jp/~matuda/Gnuplot/Pm3d/julia.plt

gnuplot is a fantastically powerful program, although making it do crazy things definitely takes some practice. You might want to start by learning how to decorate: adding a more descriptive legend (called a "key" in gnuplot), changing colors and styles, filling areas under curves, etc. To get you headed in the right direction, go here:

- http://gnuplot.sourceforge.net/help.html

From the gnuplot webpage, the first section contains many links to help and small (and large!) tutorials other people have written. Extensive help is available through gnuplot itself by typing:

```
gnuplot> help
```

It's a lot to read though, but after each screen it will give you a list of more specific things that you can get help on. These specific areas usually have lots of really useful examples that you can follow or modify for your own needs. So, if you go digging through those webpages and try out some of the samples, you will quickly find yourself making attractive graphs! Thank you for reading this tutorial; I hope it has helped you finish your lab (on time)!