An Introduction to GNUPLOT: a Linux plotting and fitting program

Gnuplot is a open source, command line driven, plotting program that is available on Linux. It has excellent capabilities for quickly plotting two- and three-dimensional data in many different formats (lines, surfaces, contour plots, etc.) Gnuplot can also be used to very quickly and easily plot mathematical functions within any specified range. It also has the ability to do fitting of data points to functions (least squares fitting).

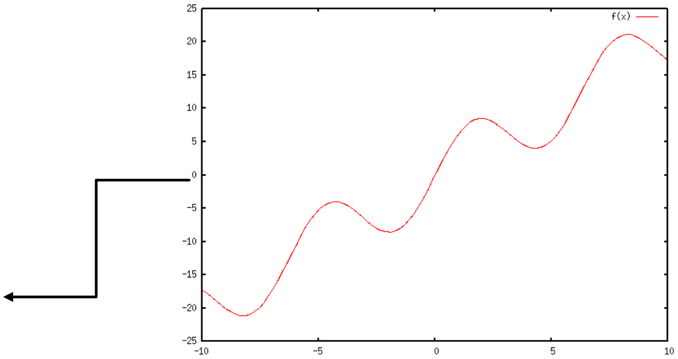
To use gnuplot on Wooki, make sure that an X-windows session is active.

Some common tasks with gnuplot are summarized below:

Opening gnuplot environment:

```
Linux> gnuplot
gnuplot>
```

Defining and plotting a function:



```
gnuplot> f(x) = 2*x + 5*sin(x)
gnuplot> plot f(x)
gnuplot> set xrange [0:20]
gnuplot> set yrange [0:30]
gnuplot> set xlabel 'x-axis name'
gnuplot> set ylabel 'y-axis name'
gnuplot> set grid
gnuplot> replot
```

Try this for other functions.

To remove changes in settings:

```
gnuplot> reset
```

Plotting data from files:

To plot a data file named *filename* use the following command:

```
gnuplot >plot 'filename' u 1:3 w linespoints
```

The name of the external file has to appear in single quotes. In this case the data is plotted **us**ing data for the x-axis from column 1 of the data file and the y-axis data from column 3. The data is plotted **w**ith the 'linespoints' option. By default, gnuplot plots column 1 as the x-axis, column 2 as the y-axis, and does not connect the points with lines.

Fitting data to a function and determining fitting parameters:

Assume we want to fit data in a file *filename.dat* to a straight line. We proceed as below:

1) Define the fitting function with unknown fit parameters. The name of the function can be anything *f*(x), *g*(x), *fff*(x), ... and the name of the fit parameters can also be any defined characters, *a*, *b*, *c*, *d*1, *g*888, …

```
gnuplot > g(x) = a + b*x
```

2) Fit the data in *filename.dat* using the function *g*(x) via the unknown parameters *a* and *b*.

```
gnuplot > fit g(x) ‘filename.dat’ via a,b
```

The fit parameters and the error margin in these fit parameters will be printed out on the screen and inserted to a file fit.log produced in the directory you are working in.

4) Plot the data points and the fit function to be sure the fit is adequate:

```
gnuplot > plot g(x), ‘filename.dat’
```

You can use any fit function as long as it is defined properly. The results of all fitting done during a gnuplot session will be written in a file called *fit.log* in your directory.

NOTE: For fits to non-linear functions, sometimes, one needs to ‘seed’ the fit parameters such that the fitting algorithm has a good initial guess. For example, if the two fit parameters are ‘a’ and ‘b’, one can ‘seed’ these values using:

```
gnuplot > a = 0.1

gnuplot > b = -94.3
```

Producing a PDF from gnuplot

Getting gnuplot to produce graphs you can print or add to MSWord documents is a bit try and involves a number of steps.

First you have to prepare the graph, do the fit, set the x-ranges and labels, etc. Then proceed as below:

```
gnuplot > set terminal postscript
gnuplot > set output ‘filename.ps’
gnuplot > replot
gnuplot > set terminal x11
```

- By default, gnuplot sends its output to a terminal window called “x11”, which is X-windows. The first command sets the output of gnuplot to a “postscript” format file instead, which is a file format similar in concept to ‘PDF’.
- The second line defines the name of the output postscript file.
- The third line actually replots the data into the above named postscript file.
- The forth line resets the output of gnuplot to the computer terminal.

Finally, once you are out of the gnuplot environment, convert the postscript file to a pdf file:

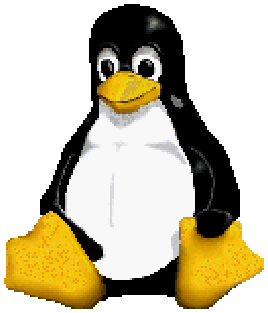
```
Linux> ps2pdf filename.ps filename.pdf
```

You can export the *filename.pdf* file to your windows system and do further manipulations on it or include it in other files.

Gnuplot Tips and Tricks

- One can scroll through your history of commands just as in the Linux shell by using the Cntl-p and Cntl-n key combinations. You can also search your history with Cntl-r.
- When typing a filename, you can use the 'TAB' key to auto-complete just like in the Linux shell.
- For more information you can go to the gnuplot website at http://www.gnuplot.info or search the internet.
- When doing fits, you need to launch gnuplot in a directory that you have write permission because it writes a 'fit.log' file.

Gnuplot Exercise



Here we will plot some data of measured distance that particles travel as a function of the ejection pressure. The advanced theory on the topic predicts that the distance, 'd', should follow the equation:

$$\mathrm{d} = a \cdot \mathrm{p}^{b}$$

where 'p' is the pressure and 'a' and 'b' are constants.

In this exercise, we will perform a non-linear least squares fit to the experimental data, in order determine the constants 'a' and 'b'.

1. Go to the directory /local_scratch/CHM4390/Gnuplot and copy the file 'penguin.dat' to your home directory. The first column is the pressure, and the second is the distance.

2. Open gnuplot and view the data in the file. Set the grid, xlabel and ylabel variables to facilitate the understanding of the data

3. Fit the data points to the above function, which in gnuplot format is given as:

   f(x) = a*(x**b)

   where a and b are fit parameters.

4. Determine the values of 'a' and 'b' that best fit the experimental data. Please note, that you may have to 'seed' the fit with values of 'a' and 'b'.

5. Construct a plot of the original data points along with the fit function. Make a pdf file of the plot.