Simple GNUPlot Commands

Luke Abraham
(Dated: 24/09/04)

As well as this file I can also recommend the webpage `http://t16web.lanl.gov/Kawano/gnuplot/index-e.html`.

I. GNUPLOT

GNUPlot is a simple but useful program producing graphs, primarily from data files, although it can draw functions such as $x \rightarrow x^2$.

It is started by the command

```
gnuplot
```

The plot command is

`p` (short for `plot`)

In the case of a function

`p x**2` (note the use of `**` instead of `^`)

If you have a file called '`data.file`' with 2 columns, and you wish to plot the left hand column as the x and the right had column as the y then type

```
p 'data.file'
```

this will plot the data as points

`p 'data' w l` (short for `with lines`)

plots with lines/

You can also change the colour of the lines

`p 'data' w l lt 3` (changes to `linetype` 3)

If you type

```
p 'data' with
```

you will get a list of possible plot styles.

If you wish to plot in a particular `x` range, say `0` to `10`;

`p [0:10] 'data.file'` (or `x**2` instead of `'data.file'`)

If you want to plot in a particular y range, say 5 to 20;

`p [][5:20] 'data.file'` (the `[]` keeps the original `x` range)

If you wanted to plot `x` between `0` and `10` and `y` between `5` and `20`;

```
p [0:10][5:20] 'data.file'
```

If you wanted to plot the right hand column as x and the left hand column as y

```
p 'data.file' u 2:1
```

If you had a datafile with 5 columns and you wanted to plot the 3rd as `x` and the 5th as `y`

```
p 'data.file' u 3:5
```

If you want to plot 2 (or more) graphs on the same axis

```
p 'data1.file' u 2:3 w l, 'data2.file' u 7:1 w lp
```

Here `lp` = `linespoints` which plots points connected by lines

You can do three dimensional plots;

```
sp sin(x)*cos(y)
```
(`sp` = `splot`. Also note that functions of `y` will not work with `p`)

Also

```
sp 'data.file' u 1:5:2
```

Note also that the `with` and the `range` commands work with `sp`, but with the `z` range placed after the `y` range

```
sp [][][0:1] y**2
```

The variable `z` is not allowed in 3D plots just as the variable `y` is not allowed in 2D plots.

If you want to give the datapoints the title of '`Points 1`'

```
p 'data.file' t 'Points 1'
```
(`t` = `title`)

It sould be noticed that the order of these is specific. The following is the correct order

```
p 'data.file' u 1:2 t 'Points 1' w l lt 4
```

If you wanted the title of the graph to be "Graph 1"

```
set title 'Graph 1'
```

The x label as "`Q`"

```
set xlabel 'Q'
```

the same command is used for setting the ylabel and the zlabel.

If you don't want the key (which lists the filenames/plot titles)

```
set nokey
```
(`set key` brings it back)

to remove the graph title/labels

```
set title
set xlabel
```

When you have the graph how you like it you can type

```
replot
```

instead of scrolling up or re-writing out the plot again

II. MAKE FILES FOR PRINT

The graphs have to be saved into a postscript file first. After you have the graph how you like it, and you want to save it to file `filename.ps`, type

```
set term postscript
set output 'filename.ps'
replot
```

To return to the original terminal so that graphs may be viewed on the screen

```
set term x11
```

also to see what other terminal types are available

```
set term
```

The file `filename.ps` is now ready to print.

You can also type help at the prompt.