

Physics 305
Introduction to Computational Physics
Fall 2009

The simpleplot program

Unfortunately, graphics programs are not as universal as we would like. One graphics program available to us is “`simpleplot`”. It is just an interface into a program called “`PhilsPlot`”, written by Phil Pinto, that we will probably use for animations later in this course. Unfortunately, this is totally homemade software, and if you want it anywhere else you will have to install it yourself. At the moment (9/27/09) it is just compiled in your instructor’s directory on `faraday`, so you will have to give some complete filenames. We may move it to a standard place once the kinks are worked out.

Before you try out “`simpleplot`”, make a couple of data files to practice on. These should be just two columns of number, where each line is an (x, y) pair. For example, file “`data1`” might look like:

```
0.0 0.123
0.5 0.2245
0.7 0.3145
1.1 0.224
```

Note that the x values don’t have to be evenly spaced.

`simpleplot` can make two kinds of output. The first is X-windows output, which makes plots appear on the screen in front of you, and the second is “`png`”, one of many standard graphics file formats. `png` can be converted to PostScript, which you can email as part of your homework assignment. (You can, and should, also view the PostScript on your screen by using the “`gv`” (short for “`ghostview`”) command.

Now try it out. Suppose “`data1`” is the name of one our your practice data files. First, you must set a couple of variables in your system (but you only need to do this once in your session, and once you get familiar with it we can show you how to make it happen automatically.) If you are using `bash` (you probably are), type (exactly):

```
PGPLOT_DIR=/home/doug/PhilsPlot/pgplot; export PGPLOT_DIR
```

```
PGPLOT_DEV=/xwin; export PGPLOT_DEV
```

If you are using `csh`, type

```
setenv PGPLOT_DIR /home/doug/PhilsPlot/pgplot
```

```
setenv PGPLOT_DEV /xwin
```

Now for the plot. Type

```
/home/doug/PhilsPlot/simpleplot data1
```

WATCH OUT. Upper and lower case matters. Spacing and punctuation matter. This should produce a graph of your data file on your screen. When you are tired of admiring the graph, type “return” in your SSH window to make it go away.

To store your result in a file, use

```
simpleplot data1 -png somefile.png
```

This will create a file called “somefile.png”. You can use “xv somefile.png” to view this file. You can convert it to PostScript using “convert somefile.png somefile.ps”. (“ps” is the conventional name extension for PostScript files.) You can check this file by

```
gv somefile.ps
```

and you can attach it as part of a homework submission.

Now, here are a few examples of more complicated things you can do. Some of these are almost always needed, like putting labels on the x and y axes. These are all “options” to the `simpleplot` command, and you can string as many of them together as you like. Here is a table of a few simple ones:

Option	purpose
-png filename	make png output
-x "time (minutes)"	label the x axis “time (minutes)”
-y "temperature (C)"	label the y axis “temperature (C)”
-log	use log scale on both x and y axes

Some examples may make this clearer.

```
graph -T ps -X "time(sec)" -Y "height (cm)" -x 0 10 -y -5 5 datafile1
```

puts labels on the x and y axes, and chooses the range. (You don’t have to choose the range; if you don’t the program will try to do something sensible.)

You can plot several files:

```
simpleplot datafile1 datafile2 datafile3
```

Plots three data lines, one from each file, in different colors.

Make a log-log plot of error versus integration bin size, for example.

```
simpleplot -x "bin size" -y "error" -log err_vs_bin_file
```

You may want to do this while logged in to faraday from your home computer. To do this, you will need to have an X-server installed. `Xming`, which we use in PAS 272, is free. I’ve never installed it, so I don’t know how hard it might be.