Graphics and Plotting

Jeremy Sanders

October 2011

1 Introduction

You will need to make figures and graphs. When you use a graph in LaTeXyou will need a PostScript (.ps) or encapsulated (.eps) version version of the plot. Alternatively, `pdflatex` can use .pdf, .png and .jpg figures. Postscript and PDF figures scale well when included in a document. Bitmapped formats may give a blocky appearance if they have too low a resolution.

Common ways to make figures and plots are:

1. Use a graphical plotting program such as `veusz`, `xmgrace`, `openoffice.org` to make a plot. `fv` can also make simple plots, but requires the XANADU package (see the Users' Guide to set this up).
2. Use a command-line based program such as `gnuplot`, `sm` (Supermongo), `idl`. `qdp` is also available in the XANADU package.
3. Create images from astronomical FITS files using `ds9` or `fv`.
4. Convert between formats with `convert`, `gimp`, `jpeg2ps` or `xv`.
5. Draw line diagrams with `inkscape`, `xfig` or `openoffice.org`/`staroffice` (draw package).
6. Produce line and bitmap graphics from your program with the `pgplot` library.
7. Generate plots from Python programs using `matplotlib`, `veusz` or `chaco` modules.
8. Draw bitmapped images with the `gimp` (good for photo manipulation).

Quite often you'll want to write a data file from your program and plot it using a plotting applications.

2 Veusz

Veusz is my own plotting package. It has a easy to use interface and can also be scripted. You can run it with the `veusz` command on Linux:

```
> veusz
```

Here is a quick guide to plotting some data:

1. Suppose you have a data file consisting of your data, e.g. in `test.dat`:

   ```
   0 0
   1 1
   2 4
   3 9
   5 25
   ```

   Read you data by going to the menu option ‘Data, Import’. Enter your filename in the filename box. You can enter ‘`x y`’ as the names for the columns in the dataset names box (you can use any names, but this is easier). Click import to get the data. If you wanted to plot error bars, add ‘`+-`’ (symmetric) or ‘`+ -`’ (asymmetric, note space) to either or both of the names, and add further columns to the data file.

   You can also plot data using CSV files saved from spreadsheets or FITS files.

2. Click the ‘Plot points’ button, which looks like a scatter-plot (or go to ‘Insert, Add xy’ on the menu).

3. Choose the names of the datasets in the properties tab (if you didn’t use ‘x’ and ‘y’ when importing).

4. Modify the appearance of the plot by using the formatting tab.

5. Export eps file using the ‘File, Export’ menu option.

6. Functions can be plotted by clicking on the function button (looks like a green sine plot), or going to ‘Insert, Add function’ on the menu.

To see example plots, open some documents from the directory `/data/ioasoft/veusz/examples`. You can add extra datasets by adding more scatter plot components, add functions, or arrange graphs in a grid. There is much more documentation on the project website at `http://home.gna.org/veusz/`, where you can find an introductory video.

If you save a plot, it is saved in the form of a script which can be modified with a text editor.

3 Gnuplot

Gnuplot is a common and easy way to make a graph (see its `help` command or the gnuplot central webpage `http://www.ucc.ie/gnuplot/`). The program is also free to redistribute, allowing it to be run on your computer at home or a laptop. To use You start gnuplot with `gnuplot`. To plot the data file you type a command like:

```
# plot points with lines between them
plot 'test.dat' with lines
# plot points with lines, and plot function x^2 (x**2)
plot 'test.dat' with lines, x**2
```

We can plot errorbars if the data file has more than two columns (here with symmetric y-error bars):

```
0 0 1
1 1 1.5
2 4 2.5
```

in gnuplot

```
set ylabel 'this is the y axis'
set xlabel 'this is the x axis!'
set title 'this is the graph title'
plot 'test.dat' with errorbars
```

To send the output graph to a file instead of a window, we do the following commands before plot (producing a monochrome eps file called `output_file.eps`).

```
set terminal postscript eps monochrome
set output 'output_file.eps'
```

(`set terminal x11` switches back to normal output mode). gnuplot has many options (log scales, tick marks, fonts, titles) accessed with the `set` command. The most useful command is the `help` command which gives you an interactive help interface.

Gnuplot can be scripted. If you want to save the commands produced in your current session use the `save` command (edit and copy with emacs in the usual way). You can load them back with the `load` command, or type `gnuplot scriptname` to execute it automatically. Gnuplot can also plot 3d-surface plots with the `splot` command, and do fitting of functions to data (non-linear squares fitting) with the `fit` command.

4 Grace (xmgrace)

Grace is a powerful plotting package, held back (IMHO) by its user interface. Start it with the `xmgrace` command. You can import data with the Data, Import, ASCII menu. Modify your plot appearance with the Plot menu.

5 SuperMongo (sm)

`sm`, with its rather dubious icon and quote 'you can't beat sm', is probably the most popular plotting package at the IoA. It is rather old, however, and is not free (the IoA pays for it), so you can't run it on your own machine without paying $$$. Lots of sm information is on its web page (see `http://www.astro.princeton.edu/~rhl/sm/`), but people often borrow other peoples' `sm` scripts and adapt them (as with anything else). `sm` is quite powerful, however, and is virtually a full programming language. `sm` scripts are like `gnuplot` commands in that they act on a data file. The best way to explain how it works is for you to either follow the tutorial (linked from my links page, `http://www.astro.princeton.edu/~rhl/sm/tutorial.html`), or look at a sample set of commands.

In my opinion the output looks pretty old fashioned, and scripts which work on one system don't work on another (due to device support). It's also relatively hard to get nice looking plots unless you already have a script which looks good.

A very simple sm script which reads two columns of data from `test.dat` would be

```
device postencap out.eps

data test.dat
read { x 1 y 2 }
limits x y
box
points x y
xlabel This is the X axis
ylabel This is the Y axis
quit
```

Enter these commands in a file e.g. `test.sm`, enter some data in `test.dat`, and do

```
sm < test.sm
```

and it should write an encapsulated postscript file called `out.eps`. You can view this with `gv`, and embed it in a LaTeXdocument.

A more complex example follows, writing to the screen instead of a file. If `test2.dat` contains

```
0 0 0
1 1 0.5
2 4 2
3 9 4.5
4 16 8
5 25 12.5
```

we can type into `sm` (or redirect as above)

```
device x11 -bg black <-- chooses window output
data test2.dat       <-- chooses test.dat as data
read { x 1 y 2 z 3 } <-- col1->x, col2->y, col3->z
limits x y           <-- limits of plot chosen by x,y
box                  <-- draw frame
connect x y          <-- draw line between x, y points
points x z           <-- draw points on x, z points
set r = y-z/2        <-- make r vector = y-z/2
points x r           <-- plot x, r
xlabel This is the x axis
ylabel This is the y axis
quit                 <-- finish
```

6 FITS viewers

FITS is a data file format for the transport of astronomical data (usually observational, but sometimes theoretical). There are several programs to view an image in a FITS file. I find `ds9` the best, but I'm not an IRAF user. `fv` is also very useful to probe the structure of FITS files, make images, plot graphs and fiddle with the keywords. You will need to set up XANADU to get fv in your path.

To use `ds9` to generate a PostScript (.ps or .eps) file to include in your LaTeXdocument, go to the File, Print menu option. Choose File as the output option and enter a filename. Choose to create a Level 1 PostScript file. Click Ok.

7 Image conversion

ImageMagick (`convert` command) is a great tool for converting between different image types. It can even create movies from single frames. The `gimp` is another good tool for image conversion, when you want more precise control.

8 pgplot

If you want to produce graphics direct from your code, the pgplot library is the most common way to do this. I won't provide you with information here on how to use it, but ask around if you want to learn. I would advise you not to produce plots from your program, unless you wish to make it interactive, but dump data files to plot instead. For the pgplot reference guide see my links page or `http://www.astro.caltech.edu/~tjp/pgplot/`.

9 OpenOffice

Openffice is an office-suite (including word processor, spreadsheet, plotting package). It is mostly-compatible with MS products (can read many doc, ppt and xls files). Use the `openoffice.org` command to start the program.

10 inkscape

Inkscape is quite a good vector drawing program, suitable for creating line diagrams for papers.

11 xfig

`xfig` is a common way of producing line drawings for papers (you can also write its input files from your program). Its user interface is rather difficult to learn (quite unlike Windows programs) though, but okay once you've got the hang of it (but also try StarOffice draw). It can write `eps` files. A useful associated program is `pstoedit`, which can convert eps/ps files to xfig files for editing (you have to download that, I'm afraid).

12 The gimp

The gimp is an excellent bitmap editing tool and is very good at converting Postscript to bitmap formats, and also for editing your photo collection (see `http://www.gimp.org/` for information). A great way of getting your figure smaller in disk space is to load the eps in gimp, save it as jpeg (you can adjust the compression here), and then use jpeg2ps to convert it back to eps! `xv` and `convert` are also useful for image conversions.

13 IDL

IDL is a very powerful way to make plots and figures. I won't include any details here, but talk to other IDL users, and look at the IDL documentation.

14 Figures in LaTeX

When you write a paper in LaTeX you can insert graphics if they are in `eps` format. You need the following text at the top of your LaTeX script:

```
\usepackage{graphicx}
```

and to insert a figure do (to make it 99% the width of a column)

```
\begin{figure}
  \centering % optional - centre figure
  \includegraphics[width=0.99\columnwidth]{figure.eps}
  \caption{Blah} % text under figure
  \label{fig:blah} % how to refer to figure
\end{figure}
```

(you can also specify an angle with `angle=90` or a height instead of a width with `height=10cm`, using commas to separate parameters). LaTeX will decide where to put the figure (usually the top of a page), but with some arm-twisting you get some control (see some LaTeX book).