gnuplot

Erik Schnetter

schnetter@cct.lsu.edu

CCT, Baton Rouge, February 2006

Scientific visualisation

Scientists runs simulations or take measurements. These data need to be presented in an appealing and professional manner.
Sometimes there are a lot of data. Sometimes it is difficult to make the interesting features come out right. Sometimes many plots must be produced.

- Interactive Tools: Easy to use, allow playing with the data and their display. Often restrictive for the professional.
- Scripting Tools: Require some knowledge to get started. Can handle any amount of data.

Outline

- Overview over existing tools
- Simple data plotting with gnuplot
- Advanced features of gnuplot
- Incorporating figures into latex

Plotting tools

Tools for creating 1D or 2D plots from numerical data:

- High level: Mathematica, Maple, IDL, Mathlab, Excel (Can also manipulate data in arbitrary ways)
- Interactive: xgraph, ygraph, xmgrace, SuperMongo
- Script based: gnuplot

First steps in gnuplot

Get gnuplot from `www.gnuplot.info`, or from your Unix distribution of choice.
Make sure you have an X11 server running. Then start it by executing `gnuplot`.
Type `help` for help, and `quit` to quit.
Say `plot sin(x)` to draw a first figure. Note how the graphical window appears.
Say `replot sin(x)**2` to add a second figure.
Say `plot sin(x), - sin(x)` to plot two graphs at once.

Note that you can use your cursor keys to edit what you type,

and to recall what you typed earlier.

Mouse interaction

Press h in the graphical window to see a help text.
Use the middle mouse button to leave a marker.
Use the right mouse botton to zoom into the picture. Press u to unzoom.
Press g to toggle a grid.
Press r to toggle a cross hair.

Press b to change the appearance of the border.

Plotting data

Most of the time, the data that should be displayed will be available in files. Gnuplot files are text files with one data point per line. Empty lines separate series of data points. Look at `u_l2_m2_mode_r3_gf.sum.asc`, which is a real data file.
Column 2 contains the simulation time, column 3 the amplitude of a wave. Say (on a single line) `plot "u_l2_m2_mode_r3_gf.sum.asc" using 2:3 with lines` to display this.

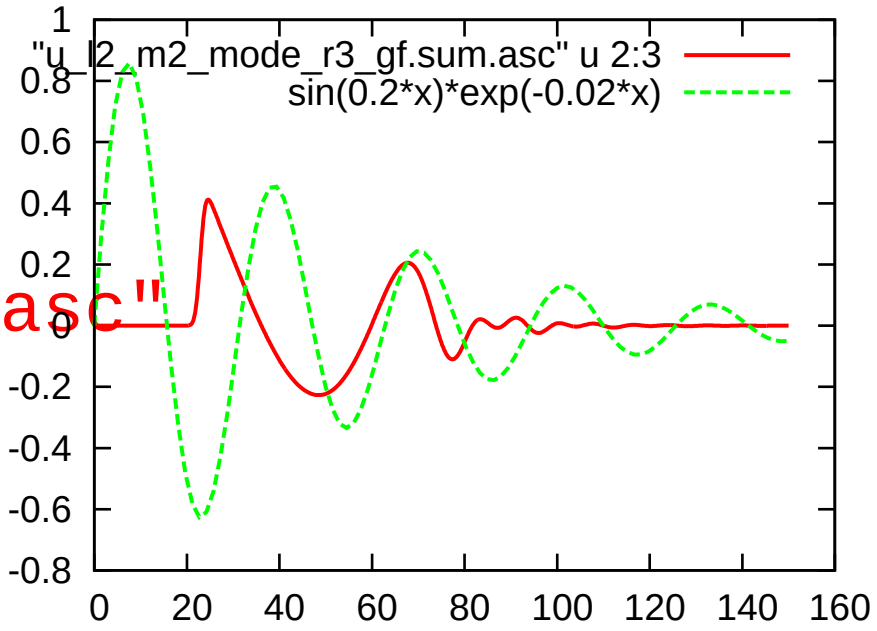
(Data files are on `is.cct.lsu.edu` in the directory `~eschnett/gnuplot`.

More plotting

- Abbreviate commands: `plot` $\longrightarrow$ `p`, `using` $\longrightarrow$ `u`, `with` $\longrightarrow$ `w`, `lines` $\longrightarrow$ `l`
- Use different plotting styles: `p` (points), `lp` (linespoints), `d` (dots)
- Say `test` to see all plotting styles (and colours)
- Select regions: `set xrange [0:100]`, `set yrange [-0.5:0.5]`, `set autoscale x`

Fancy plotting

Plot a function on top of the data: `p "u_l2_m2_mode_r3_gf.sum.asc" u 2:3 w l, cos(0.2*x)*exp(-0.02*x)`



Plot two data files, scaling the second: `p "u_l2_m2_mode_r3_gf.sum.asc" u 2:3 w l, "u_l4_m2_mode_r3_gf.sum.asc" u 2:(1000*$3) w l`

Using a script

With time, it becomes tedious to repeat these commands, especially if the commands are longish. Write a script instead. Put the commands into a text file, then say `load "textfile"` in gnuplot. Or use cut-and-paste from the text file. This makes it easy to try many variations of the commands.

When you create a figure for a paper, it is important to document how the figure was created. Keep the gnuplot script together with the data and the paper. The script documents what data are displayed in what way. Both the data and the script should go into CVS, together with the paper.

Writing figures into files

In general, there are two kinds of file formats for figures: pixel-based (e.g., gif, tiff, jpeg, png) and scalable (e.g., eps, png). Scalable file formats are much smaller and look much better for graphs. Really.

In gnuplot —or in a script— say `set terminal postscript eps enhanced color` to produce output in "eps" format instead of for the screen. Say `set output "filename.eps"` to redirect output to a file.

Say `set terminal x11` and `set output` to switch back.

Incroporating figures into latex

You can incorporate eps figures into your latex documents. Here is an example:

```
\documentclass{revtex4}
\usepackage{graphicx}
\begin{document}
\begin{figure}
  \includegraphics[width=0.45\textwidth]
    {u_l2_m2_mode_r3_gf.sum}
\end{figure}
\end{document}
```

Brushing up figures

```
set terminal postscript eps enhanced color
set output "nicefigure.eps"
set size 0.5 # larger text
set title "Waveform"
set xlabel "time"
set ylabel "amplitude"
p "u_l2_m2_mode_r3_gf.sum.asc" u 2:3 \
  t "large" w l lw 3 lt 3, \
  "u_l4_m2_mode_r3_gf.sum.asc" u 2:3 \
  t "small" w l lw 3 lt 4
```

`t "large"` assigns a name to the graph.

`w l lw 3 lt 3` means “with lines, line width 3, line type 3”.

2D plots

It is also possible to plot surfaces in gnuplot.
Say for example `sp "h.t188743680.ah1.gp" u 4:5:6 w l`.
Or say `sp "h.t188743680.ah1.gp" u 4:5:6 w l, "h.t188743680.ah2.gp" u 4:5:6 w l, "h.t188743680.ah3.gp" u 4:5:6 w l`

Use the mouse to rotate and zoom the graph.

Fancy 2D plots

Say `set pm3d`, then `sp "waveeqn-lagrange-kesc-me.asc" u 1:2:($2*$5) w l`.

Try also `set contour surface`. Or `set hidden3d`.

Various other commands

- For logarithmic scaling, say `set log x` etc.
- To control the aspect ratio, say e.g. `set size ratio -1`.

Closing remarks

The advantages of scripting:

- The results are repeatable. This is very useful if one needs many similar graphs, or if one potentially needs to redo graphs with updated data.
- Plots can be produced automatically. Sometimes plots take a long time, especially if there are much data to be processed.
- Plots can be produced on a remote supercomputer, written into a file, and then displayed locally. The data do not need to be transferred.
- For a quick look at small datasets there are many tools. For serious work gnuplot is one of the best.