

Slide 1

Administrivia

- Homework 6 not due until next Wednesday, but work on it if you can, before you forget all about `make`?
- "Sample solution" to Homework 4 posted on the "lecture topics etc." page. (It's basically a list of things students this year and in previous years mentioned and might be worth a skim.)

Slide 2

Minute Essay From Last Lecture

- Many people, though not all, had used `make` in some class, but it sounds like few people had written `makefiles`. So Homework 6 should be useful!

Homework 4 Essays

Slide 3

- These weren't particularly noteworthy since in a way the whole assignment was an essay.
- Many people seemed to have learned things that they thought could be useful.
- Several people expressed interest in `emacs`. My sense is that most of what makes it more attractive than "real" `vi` (syntax highlighting, automatic indentation, semi-visual cut/copy/paste, etc., etc.) can be done in `vim`. But it's a fine editor. Maybe the biggest drawback in our environment is the lack of local experts!
- A few people mentioned preferring IDEs for large projects. Agreed that they have some useful features!

(Mostly-)Text-Mode Plotting — `gnuplot`

Slide 4

- Often run in graphical mode, but interface is text-only. Help available from within program by typing `help`. (Help is modeled after online help on VAX VMS operating system, and is — different.)
- (Interestingly enough, the name has nothing to do with the GNU project, but apparently was an attempt at humor by the authors, who wanted to call it `newplot` but didn't want to conflict with some other program by that name.)
- Admittedly the text-only interface seems clunky by comparison to GUI tools, and this does seem like the kind of application for which a GUI tool would be good, but ...

gnuplot, Continued

Slide 5

- A benefit to the text-mode interface is that commands/settings can be saved to a (text!) file for later reuse. (And since it's a text file it can be edited directly with a text editor.)
- Program can also be run in "batch" mode. Might be useful for generating many similar plots from different data, or for recreating plots as data changes.
- All this makes for something that fits nicely into the world of traditional-UNIX text-mode tools.
- Nice for \LaTeX users because it can produce output in various \LaTeX -friendly formats (including ones that allow final typesetting to use same fonts as document). (More about this in discussion of \LaTeX — next topic.)

gnuplot Basics

Slide 6

- `plot` to plot a mathematical expression or data.
- `set xrange`, `set yrange` to set ranges for two axes.
`set logscale` to scale one or more axes logarithmically.
- `set style` to use nondefault "style" (particularly useful for plotting data — default is points, but you can do lines or bars too).
- `replot` to repeat previous `plot` command, as you might want to do after changing style, etc.
- `save` to save settings to a file. `load` to retrieve them.
- `set terminal` to choose a different output format. Combine with `set output` to save plot to a file.

gnuplot Examples

- A few examples on “Sample programs” page.
- *Lots* of examples available at `gnuplot.sourceforge.net`

Slide 7

Minute Essay

- Can you think of a situation in which you might use `gnuplot`?
- What tool(s) do you usually use to make plots?

Slide 8