The Gnuplot Graphing program

“”

Contents

.

Gnuplot is an amazingly capable program with the usual plethora of switches, options, codes, and hidden and secret corners only explored by university thesis writers, who occasionally publish a desultory web-page describing some serendipitous discovery.

Gnuplot can make mathematical graphs and curves as well as bar-charts (histograms) along with lots of other things. In this book, the convention of writing '$>>' may be used to indicate commands which are entered from within the gnuplot interpreter.

`http://www.gnuplot.info/`
: The official home page of gnuplot

`http://www.ibm.com/developerworks/library/l-gnuplot/`
: a good tutorial

`http://sparky.rice.edu/gnuplot.html`
: an intro to gnuplot

`http://t16web.lanl.gov/Kawano/gnuplot/plot1-e.html`
: lots of info

Start the interactive gnuplot shell

```
gnuplot
```

Section 1

History

Originally developed by Colin Kelley and Thomas Williams in 1986

Section 2

Getting Help

View a general help menu with submenu selections

```
help
```

Get help for the gnuplot 'set yrange' command

```
help set yrange
```

Display test graphs with example linewidths and pointtypes

```
test
```

View a list of all keywords setting with the 'set' command

```
set
```

View a list of all possible options after the 'with' keyword

```
plot x with
```

Section 3

Command Abbreviations

Any command can be abbreviated to any length as long as there is no ambiguity, that is, no other command which could be referenced by the same abbreviation (eg 'u' or 'usi' for using) . Other commands have and explicit abbreviation (eg 'lt' for 'linetype')

example

`using`	'u', 'usi', 'usin' are all valid abbreviations

Section 4

Simple Usage

Show the graph of 'parabola' curve (x to the power of 2)

```
gnuplot      ~(a welcome message is shown and a new 'gnuplot' prompt
  ⇒ starts)
```

```
plot x**2    ~(the graph is shown in a separate window)
```

```
quit         ~(exit gnuplot, the graph window is also closed)
```

Plot a sin curve with x-axis (horizontal axis) -20 to 20

```
set xrange [-20:20]
```

```
plot sin(x)
```

Section 5

Axis Value Range

Plot a sin curve with x-axis -20 to 20, all commands on one line

```
set xrange [-20:20]; plot sin(x)
```

```
plot [-20:20] sin(x)       ~(the same)
```

Plot a sin curve with x-axis -pi to pi ('pi' is a mathematical transcendental)

```
set xrange [-pi:pi]; plot sin(x)
```

```
plot [-pi:pi] sin(x)    ~(almost the same)
```

Plot the function 'y=x' with a y-axis range of [-2:5]

```
plot [][-2:5] x
```

```
plot [  ] [-2:5] x          ~(the same, spaces dont matter)
```

```
set yrange [-2:5]; plot x   ~(the same, but persists for next graph)
```

Set the x value range to 0 to a multiple of pi

```
set xrange [0:2*pi]
```

Section 6

Using The Gnuplot Shell

Plot a curve, add a title and then replot it (which saves some typing)

```
plot sin(x); set title 'hello'; replot
```

Span a single command over several lines with a backslash '\'

```
set \
grid
```

Section 7

Graph Titles

Plot a graph of the sin curve with a graph title 'a simple curve'

```
set title "a simple curve"; plot sin(x)
```

```
set title 'a simple curve'; plot sin(x)   ~(the same, no escapes)
```

Set the title for just one plot

```
plot x*x title 'a parabola'
```

```
plot x*x t 'a parabola'    ~(exactly the same)
```

Create a title for a parabola graph which is on 2 separate lines

```
set title "a simple\ncurve"; plot x**2   ~(note the double quotes)
```

Plot an exponential with the title 'curvy' from the (bash) command line

```
echo 'set title "curvy"; plot x**x' | gnuplot -persist
```

Section 8

Borders And Frames

The border is the box which surrounds the graph and on which the tick marks are drawn.

8.1 Erasing Borders

To erase some or all or the borders of a graph one must use a sort of binary arithmetic.

Remove all borders from the graph (but the ticks remain)

```
set border 0
```

Display only the bottom and left borders, tic marks and labels

```
set border 3; set xtics nomirror; set ytics nomirror
```

```
set border 3; set tics nomirror    ~(the same, with modern gnuplot)
```

Show the bottom, left and right borders, but not the top, nor the top tics

```
set border 11; set xtics nomirror; plot x**2
```

Show no graph border, no tic marks and no value labels for a parabola

```
unset tics; set border 0; plot x**2
```

```
set notics; set border 0; plot x**2   ~(an older syntax)
```

Display only the bottom border for a parabola graph

```
set border 1; plot x**2
```

the graph border 'keys'

1	Bottom
2	Left
4	Top
8	Right

border examples

`set border 3`	(1+2) display bottom and left borders
`set border 6`	(2+4) display left and top borders
`set border 11`	(1+2+8) display the bottom, left, and right borders

Axis Labels

The axis label is the text which is printed alongside of the vertical (y) or horizontal (x) axis. This text generally explains the meaning of the value of that axis, such as 'temperature' or 'date' etc.

The formal syntax statement

```
set xlabel {"<label>"} {offset <offset>} {font "<font>{,<size>}"}
       {{textcolor | tc} {lt <line_type> | default}} {{no}enhanced}
       {rotate by <degrees>}
```

Show the settings which currently apply for the x-axis label

```
show xlabel
```

Suppress the label on the vertical axis (no label is shown)

```
set ylabel
```

Set the vertical axis label to be the text "degrees"

```
set ylabel "degrees"
```

```
set ylabel 'degrees'      ~(the same, but maybe not for all versions)
```

Plot a graph of a parabola (x squared) setting the y-axis label

```
set ylabel "sin"; plot x**2
```

Put a newline character in the y-axis label

```
set ylabel "Degrees\ncelsius"
```

```
set ylabel 'Degrees\ncelsius'   ~(No! it doesnt work, no escapes)
```

Plot a graph of sin(x) setting the text of the x-axis and y-axis labels

```
set xlabel "Angle, \n in degrees"; set ylabel "sin"; plot sin(x)
```

9.1 Axis Label Font And Size

Set the font for the vertical axis label to Courier, size 14 points

```
set ylabel font "courier,14"
```

Set the font for the vertical axis label to Courier

```
set ylabel font "courier"
```

Set the horizontal axis label to the colour specified by 'linetype 4'

```
set xlabel textcolor lt 4   ~(but what colour is lt 4? you ask)
```

```
set xlabel tc lt 4          ~(the same)
```

9.2 Rotating The Axis Label

This isnt working for me

Rotate the label

```
set xlabel "Date" rotate by -45;
```

9.3 Moving The Axis Label

Move the x axis label 1 character width to the left

```
set xlabel offset -1,0
```

9.4 Customizing The Axis Label

Set the horizontal label text, the font and the text color in one go

```
set xlabel "hello" font "helvetica" textcolor lt 6
```

```
set xlabel "hello" font "helvetica" tc lt 6
```

Section 10

Graph Size And Scale

Make the graph an exact square (regardless of value ranges

```
set size square    ~(only available in modern versions of gnuplot)
```

Scale the graph so that the y (vertical) axis is twice as long as the x-axis

```
set size ratio 2
```

Scale the graph so that the x (horizontal) axis is twice as long as the y-axis

```
set size ratio 0.5
```

Scale y-axis by 2, retain x-axis size

```
set size ratio square 1,2
```

Section 11

Grids

Plot a sin curve over a grid of dotted lines (grid lines at each value 'tic') ::gnuplot>> set grid; plot sin(x)
Remove a value grid from a plotted graph

```
unset grid; replot;
```

```
set nogrid; replot;     ~(older versions of gnuplot)
```

Section 12

The Value Ranges

Plot only data from 0.5 to 10 on the x-axis and 30 to 48 on the y-axis

```
set xrange [0.5:10]; set yrange [30:48]; plot 'data.txt'
```

```
plot 'data.txt' [0.5:10][30:48]    ~(this is the same)
```

Plot data using using an x-range of 0 to the highest x value

```
set xrange [0:]; plot 'data.txt'  ~(the horizontal axis starts at 0)
```

Reset the value ranges previously set with 'set xrange' or 'set yrange'

```
reset     ~(the value ranges revert to the defaults)
```

Section 13

Tics

'Tics' are those small little lines which 'mark' or are perpendicular to the axis (either horizontal or vertical). These tics are supposed to indicated value points or ranges. I think it should be spelt 'tick' but in gnuplot its spelt 'tic'.

View help for all the options for changing the x-axis value tics

```
help set xtics
```

Show information about all options currently set for the ytics

```
show ytics
```

Show the vertical (y-axis) tics outside of the axis line

```
set ytics out    ~(this affects the 'mirrored' opposite tics as well)
```

Plot a curve with no 'tics' or value labels on the horizontal axis

```
unset xtics; plot sin(x)
```

Display vertical tics at intervals of 2 (by value)

```
set ytics 2
```

Display horizontal tics only starting at 50 with interval 100

```
set xtics 50,100     ~(values outside this range are still plotted)
```

Set x (horizontal) tics at the values 1, 2, 4, ... 1024

```
set xtics (1,2,4,8,16,32,64,128,256,512,1024)
```

Set y (vertical) tics at the values 1, 11, and 21

```
set ytics (1,11,21)
```

13.1 Minor Tics

Minor tics are the even smaller little markes which occur on the horizontal and vertical axis

Display one minor tic halfway between each major tic

```
set mxtics 2
```

Make the minor and major tics the same length

```
set ticscale 1 1
```

Section 14

Tic Labels

Plot a paraboloa with the x-axis labels rotated clockwise by -45 degrees

```
set xtics rotate by -45; plot x**2
```

Rotate the tic labels by 90 degrees

```
set xtics rotate
```

Plot a sin curve with only 3 x-axis value tics, on a grid

```
set xtics ("0" 0, "90" pi/2, "-90" -pi/2); set grid; plot sin(x)
```

(the grid only has 3 vertical lines, since there are only 3 x-axis tics)

Set the value tics for the x-axis (format: "label" value [level])

```
set xtics ("0" 0, "90" pi/2, "-90" -pi/2, "" pi/4 1, "" -pi/4 1, "" 3*
  ⇒ pi/4 1, "" -3*pi/4 1)
```

Explicit 'tic' examples

```
  set xtics ("low" 0, "medium" 50, "high" 100)
  set ytics ("bottom" 0, "" 10 1, "top" 20)
```

14.1 Text Tick Labels

The tic label is the text which sits just next to the little tick mark on the vertical or horizontal axis. These labels can be customized in gnuplot in many ways.

Explicitly set three tic text labels at the values 0, 50 and 100)

```
set xtics ("low" 0, "medium" 50, "high" 100)
```

Dont put tics on the oposite side to the main y-axis (vertical axis) $>> set ytics nomirror there are no tics but the oposite box-line stays

Display 3 tics on the vertical axis with no tick labels (no text at the tick)

```
set ytics ("" 0, "" 10, "" 20)
```

Add a tic and label 'Pi' on the x axis without affecting the default tics

```
set xtics add ("Pi" 3.14159)
```

Make tics on the y-axis 0,.5,1,1.5...10 and added one label 'Pi' at 3.141

```
set ytics 0,.5,10; set ytics add ("Pi" 3.141)
```

14.2 Tic Intervals

Display the vertical tics at value intervals of 5

```
set ytics 5    ~(this places tics at ... -10, -5, 0, 5, 10 etc)
```

Display the x-tics starting at value 0, ending at 10 with an interval 0.5

```
set xtics 0,0.5,10   ~(this show values 0, 0.5, 1, 1.5 ... 9.5, 10)
```

data plotting styles

`points`	Each value a point or a cross or a box etc
`lines`	A straight line from each data value to the next
`linespoints`	A line with points to indicate values
`steps`	Looks like a city skyline
`boxes`	These look like a histogram and are normal contiguous (no gaps)
`errorbars`	
`impulses`	Lines which go from the axis to the value point
`and others ...`	

Set the style of data plotting to 'histogram'

```
set style data histogram
```

15.1 Getting Help

View a list of the available plotting styles

```
plot x with
```

View help for different curve plotting styles

```
help with                  ~(lots of examples)
help plot with             ~(the same)
help plotting styles       ~(gnuplot version >= 4.0)
```

Show the current setting for what style of data plotting

```
show style data
```

See a list of valid data plotting styles

```
set style data
```

15.2 Plotting With Points

Plot a parabola with points

```
plot x**2 w p
plot x**2 with points                  ~(the same)
set style data points; plot x**2       ~(the same, but persistent)
set style data p; plot x**2            ~(the same, again)
```

Plot the line 'y=x' with points (crosses) which are 4 times the normal size

```
plot x w points pointsize 4    ~(the default point seems to be a cross)
plot x w points ps 4           ~(the same)
plot x w p ps 4                ~(the same, again)
```

Plot a parabola with a line with 'x'es at value points

```
plot x**2 with linespoints
plot x**2 w linespoints                    ~(the same)
plot x**2 w lp                             ~(the same again, nice and terse
  ⇒ )
set style data linespoints; plot x**2      ~(the same, persistent)
```

Plot a parabola with a line with little boxes at value points

```
plot x**2 with linespoints pointtype 5
```

```
plot x**2 w lp pt 5          ~(the same, not so verbose)
```

Plot a parabola with 'boxes' or contiguous vertical bars (a histogram)

```
plot x**2 w boxes      ~(this looks like a mathematical bar-chart)
```

```
plot x**2 with boxes   ~(the same)
```

Plot a parabola with purple bars

```
plot x**2 with boxes lt 4
```

```
plot x**2 with boxes linetype 4    ~(the same)
```

Plot a sin curve and a cos curve on one field using different curve types

```
plot sin(x) with linespoints pointtype 5, cos(x) with boxes
```

the linespoints pointtype values

4	Empty boxes
5	Filled boxes

Section 16

The Graph Legend

The 'legend' of the plot describes what each curve (or dataset) actually means and by default is in the top right corner.

Put a box around the graph legend and place it in the top left corner $>> set key top left; set key box; plot x*x;

Section 17

Side By Side Graphs

Side by side

```
# multiplot mode
# This sets up bounding boxes and may be required on some terminals
set size 1,1
set origin 0,0

# Done interactively, this takes gnuplot into multiplot mode
# and brings up a new prompt ("multiplot >" instead of "gnuplot >")
set multiplot

# plot the first graph so that it takes a quarter of the screen
set size 0.5,0.5
set origin 0,0.5
plot sin(x)

# plot the second graph so that it takes a quarter of the screen
set size 0.5,0.5
set origin 0,0
plot 1/sin(x)
unset multiplot
reset
```

Section 18

Drawing Arrows

Draw some arrows

```
set arrow from 1,2 to 4,8.4 nohead lt -1 lw 1.2
```

The process of plotting data with gnuplot involves taking a text file which has 'fields' separated by a 'separator' character or characters (usually a space or tab characters) and turning that data into a graph or chart. Often this chart would be a 'histogram' (that is a 'bar-chart), but other forms are possible.

`http://www.linuxquestions.org/questions/linux-software-2/gnuplot-how-to-label-x-axis-with-`
How to use a field to supply the x-tic labels (horizontal axis value labels)

The word 'field' and 'column' are used interchangably.

19.1 The Gotchas

File names must always be enclosed in quote characters (unless its the shell)

```
plot data.txt      ~(No! doesnt work)
```

```
plot 'data.txt'    ~(correct)
```

Normally the datafile should only contain numbers (or use 'xticlabels' etc)

```
datafile:
  italy 2
  spain 5
  ...
>> plot 'data.txt'   ##(this produces an error because of the text '
 ⇒ italy' etc)
```

If the data file contains text (not numbers), use 'using' to avoid that field

```
plot 'data.txt' using 2:3   ~(the first field may contain text)
```

19.2 Basic Data Plotting

List the contents of the text data file 'data.txt'

```
!less data.txt
```

Plot data using column 1 for the x-axis and column 2 for the y-axis

```
plot 'data.txt' using 1:2   ~(data is plotted with little crosses '+')
```

Plot data using field 3 for the x-axis and field 2 for the y-axis

```
plot 'data.txt' using 3:2
```

Plot data which is entered at the terminal, (useful for experimenting)

```
plot '-'
 1 10
 2 20
 3 5
```

19.3 Plotting Data With Value Ranges

The default data range is from the minimum value to the maximum + 1 on both the x and y axes.

Plot the data using a horizontal (x) range of 0 to 20 and the default y-range

```
plot [0:20] 'data.txt'
```

```
set xrange [0:20]; plot 'data.txt' ~(the same, but persists)
```

Plot field 2 vs field 3 using an x range of -10 to 10 and a y-range of -5 to 6

```
plot [-10:10][-5:6] 'data.txt' using 2:3
```

Plot column 2 vs 3 using the y value range of -5 to 5 and the default x range

```
plot [][-5:5] 'data.txt' using 2:3
```

```
set yrange [-5:5]; plot 'data.txt' using 2:3   ~(the same, but persists)
```

Start the x and y value range at -10 to their respective defaults

```
plot [-10:][-10:] 'data.txt'
```

Plot the data file 'data.txt' with the x (horizontal) range starting at 0

```
use xrange [0:]; plot 'data.txt'   ~(file can only contain numbers)
plot [0:] 'data.txt'               ~(the same)
```

19.4 Plotting Text Value Data

Plot 'list.txt' taking x-axis labels from the first column of the datafile

```
plot "list.txt" using 2:xticlabels(1)    ~(gnuplot version >= 4.1)
plot "list.txt" u 2:xticlabels(1)        ~(the same)
```

Plot some text versus number data from the bash command line

```
echo 'plot "list.txt" using 2:xticlabels(1)' | gnuplot -persist
```

Plot label vs value data using boxes instead of little crosses

```
plot 'data.txt' using 2:xticlabels(1) with boxes ~(almost like a
  ⇒ barchart)
plot 'data.txt' using 2:xticlabels(1) w boxes    ~(the same)
plot 'data.txt' u 2:xticlabels(1) w b            ~(the same, but better
  ⇒ )
```

Plot text vs value data using lines (a crooked line)

```
plot 'data.txt' u 2:xticlabels(1) w l
plot 'data.txt' using 2:xticlabels(1) with lines ~(the same)
```

By default the labels are printed horizontally

Plot the data with labels from the 1st column, rotated 45 degrees clockwise

```
set xtics rotate by -45; plot "data.txt" using 2:xticlabels(1)
```

Use lines starting at the axis going to the value to graph the data

```
plot "data.txt" u 2:xticlabels(1) w imp
plot "data.txt" using 2:xticlabels(1) with impulses  ~(the same)
```

Define a new thick linestyle and use it to graph impulses, barchart style

```
set linestyle 1 lt 1 lw 50;
set linestyle 1 linetype 1 linewidth 50      ~(the same)
set style line 1 linetype 1 linewidth 50     ~(more modern versions of
  ⇒ gp)
plot 'd.txt' u 2:xticlabels(1) with impulses linestyle 1
plot 'd.txt' u 2:xticlabels(1) w imp ls 1     ~(the same)_
```

19.5 Error Bars

Plot data which contains an 'error margin' in the 3rd column

```
plot "test.dat" using 1:2:3 with yerrorbars
example data:
1.0   1.2   0.2
2.0   1.8   0.3
3.0   1.6   0.2
```

http://t16web.lanl.gov/Kawano/gnuplot/plot5-e.html
examples of drawing barcharts with 'impulses'

The simplest way to plot a bar-chart or histogram is to use the 'histograms' plotting style (after 'with') The 'histograms' is by default separated and the 'boxes' are by default contiguous.

20.1 Bar Charts With Histograms

histogram styles

```
set style histogram clustered {gap <gaps
set style histogram errorbars {gap <gapsize>} {<linewi
set style histogram rowst
set style histogram columnst
```

If the bars on the bar-chart are filled with colour then the tics on the axis are no longer visible.

View help about using the 'histograms' plotting style

```
help histograms
```

See help about fillstyles for use with boxes and histograms

```
help set style fill
```

Set the gap between bars to be equal to the width of the bar

```
set style histogram clustered gap 1   ~(the bar is centered over the
  ⇒ tick)
```

Narrow the gap between bars by increasing the boxwidth

```
set style histogram clustered gap 1; set boxwidth 1.5
```

```
se sty histog clustered gap 1; set boxwidth 1.5       ~(abbreviated)
```

Set the bars to be filled with solid colour

```
set style fill solid 1; plot 'data.txt' u 1:2 w histograms
```

Fill each bar with colour which is half the intensity of the border colour

```
set style fill solid 0.5; plot 'data.txt' u 1:2 w histograms
```

Fill each bar with 'pastely' (faint) colour without any borders on the bars

```
set style fill solid 0.5 noborder; plot 'data.txt' u 1:2 w histograms
```

Plot column 1 vs column 2 of 'data.txt' with separated 'bars'

```
plot 'data.txt' using 1:2 with histograms
```

```
plot 'data.txt' u 1:2 w histo                          ~(the same)
```

```
set style data histograms; plot 'data' using 1:2   ~(the same)
```

Use labels with multiple columns

```
plot 'file.dat' using 2, '' using 4, '' using 6:xticlabels()
```

Plot text vs number data as a barchart with a y-axis range starting at 0

```
plot [][0:] 'data.txt' u 2:xticlabels(1) w histograms
```

Plot col 2 and col 3 as histograms clustered around the labels of field 1

```
plot 'data.txt' u 2:xticlabels(1) w histog, '' u 3 w histog
```

20.2 Bar Charts Using Boxes

Plot a barchart with filled boxes (fill intensity 0.7)

```
plot "test.dat" usi 1:2 w boxes fs solid 0.7
```

```
plot "test.dat" using 1:2 w boxes fillstyle solid 0.7    ~(the same)
```

Create a barchart with boxes taking up half the possible space

```
set boxwidth 0.5; plot 'test.dat' using 1:2 with boxes   ~(boxes
  ⇒ separated)
```

```
set boxwidth 0.5; plot 'test.dat' u 1:2 w boxes          ~(the same)
```

```
set boxw 0.5; plot 'data.txt' u 1:2 w boxes              ~(the same)
```

Make thin filled rectangles for the bar chart

```
set boxw 0.2; plot 'data.txt' u 1:2 w boxes fs solid 0.7
```

20.3 Bar Charts With Impulses

Use lines from the axis to the value ('impulses') for graphing the data

```
plot "data.txt" u 2:xticlabels(1) w imp
```

```
plot "data.txt" using 2:xticlabels(1) with impulses   ~(the same)
```

Section 21

The Datafile Format

The text datafile normally consists of rows of data where each row consists of a set of numbers separated by space or tab characters.
If a line begins with '#' is normally ignored
Set the character which indicates that data is missing in the data file. $>> set datafile missing "-"
Show what character currently indicates missing data in the text file $>> show datafile missing there is no default missing data character
Set the character which separates fields in the datafile to a comma ','

```
set datafile separator ','
```

21.1 Plotting Only Some Of The Data

Plot only every second line from the text data file 'test.dat'

```
plot "test.dat" every 2
```

Plot only every second data block from the data file

```
plot "test.dat" every :2   ~(the datablocks are separated by blank lines
  ⇒ )
```

More examples

```
every I:J:K:L:M:N
I     Line increment
J     Data block increment
K     The first line
L     The first data block
M     The last line
N     The last data block
every 2       plot every 2 line
every ::3     plot from the 3-rd lines
every ::3::5  plot from the 3-rd to 5-th lines
every ::0::0  plot the first line only
every 2::::6  plot the 1,3,5,7-th lines
every :2      plot every 2 data block
every :::5::8         plot from 5-th to 8-th data blocks
```

Use a shell command to select or modify data to plot

```
>> plot "< head -10 test.dat" using 1:2 with lines
>> plot "< tail -3 test.dat" using 1:2 with lines
>> plot "< head -5 test.dat" using 1:2 with lines,\
>        plot "< tail -5 test.dat" using 1:2 with points
```

21.2 Calculating The Data

Plot data from 'data.txt' doing arithmetic on each column

```
plot 'table.dat' using ($3/$1):($2*134.44)
```

Plot field 1 versus the square root of field 2

```
plot "test.dat" using 1:(sqrt($2)) with points
```

21.3 Plotting Time Data

```
help set timefmt
```

```
help time/data
```

The data file 'data.txt'

```
 10-Jun-04      90.23
 9-Jun-04       89.90
 8-Jun-04       88.64
 7-Jun-04       88.75
 4-Jun-04       87.95
 3-Jun-04       87.85
```

```
set xdata time            # The x axis data is time
```

```
set timefmt "%d-%b-%y"    # The dates in the file look like 10-Jun-04
```

```
set format x "%b %d"      # On the x-axis, we want tics like Jun 10
```

```
plot ["31-May-04":"11-Jun-04"] 'data.txt' using 1:2 with linespoints
```

A clustered histogram with time data

```
 10-Jun-04        90.23   90.75   89.89   90.46
 9-Jun-04         89.90   90.55   89.81   90.09
 8-Jun-04         88.64   90.50   88.40   90.04
 7-Jun-04         88.75   88.99   88.01   88.64
 4-Jun-04         87.95   88.49   87.50   87.56
 3-Jun-04         87.85   88.10   87.35   87.35
```

Section 22

Multiple Graphs On One Field

Plot the curves x-squard and x-cubed on the same grid (field)

```
plot x**2, x**3
```

Section 23

Output Formats

View all possible output formats which gnuplot can produce

```
set terminal
```

Create a 'png' image file 'out.png' with the graph of a sin curve

```
set terminal png; set output "out.png"; plot sin(x)
```

Plot graphs to a linux window (using wxWidgets tool kit)

```
set terminal wxt
```

some important output formats for gnuplot

`svg`	Scalable vector graphics (if you want to resize alot)
`png`	Image format good for web-pages
`jpg`	Image also good for web-pages
`x11`	Unix/linux window
`wxt`	Linux window
`gif`	Good for animations

Create a png image 'o.png' graph of a sin curve in the current folder

```
echo "set terminal png; set output 'o.png'; plot sin(x)" > temp;
 ⇒ gnuplot temp
```

```
echo "set terminal png; set output 'o.png'; plot sin(x)" | gnuplot ~(
 ⇒ same)
```

Create a jpeg image file 'o.jpg' graph of a sin curve with a value grid

```
echo "set terminal jpg; set output 'o.jpg'; set grid; plot sin(x)" |
 ⇒ gnuplot
```

Create postscript file 'o.eps' graph of a sin curve

```
echo "set terminal eps; set output 'o.eps'; plot sin(x)" | gnuplot
```

(you could include this in a quick report with 'enscript')

Section 24

Vim And Gnuplot

It is possible to create some new commands and command mappings to automatically generate graphs from the lines in a text file (be it data or gnuplot commands)

A vim command to plot the current line (assuming it is a gnuplot command)

```
.w !sed 's/^ *#//;s/ \#.*$//' | gnuplot -persist
```

A 'mapping' to plot the current line (as above)

```
:map! ,gp .w !sed 's/^ *#//;s/ \#.*$//' | gnuplot -persist
```

A new vim command to plot the current line

```
:command! Gp .w !sed 's/^ *#//;s/ \#.*$//' | gnuplot -persist
```

A vim command to plot the current line to the file 'o.jpg'

```
:.w !sed 's/^ */set terminal jpeg; set output "o.jpg";/' | gnuplot
```

A command to plot to an image to a given file name in the image folder

```
command! -nargs=1 Glti .w !sed 's/^ */set terminal jpeg; set output "
 ⇒ image\/<args>.jpg";/' | gnuplot; gthumb image/<args>.jpg
```

The new vim command above can be executed with

```
:Glti test
```

Section 25

Plotting From The Command Line

Plot a sin curve, from x=-20 to 20, from a bash shell, leaving plot window open

```
echo 'set xrange [-20:20]; plot sin(x)' | gnuplot -persist
```

(gnuplot displays the graph in a separate window, which it leaves open) (-persist keeps the plot window open after gnuplot exits)

Run gnuplot in batch mode with a command file

```
echo 'set grid; plot sin(x)' > gp.txt; gnuplot -persist gp.txt
```

```
echo 'set grid; plot sin(x)' > gp.txt; echo 'load gp.txt' | gnuplot -
  ⇒ persist
```

Quickly graph a list of numbers

```
gnuplot -persist <(echo "plot '<(sort -n listOfNumbers.txt)' with lines
  ⇒ ")
```

An example of creating a graph with gnuplot

```
http://www.pixelbeat.org/docs/web/access_log/analyzing.html
```

3D GRAPH PLOTTING

The gnuplot command 'splot' is capable of plotting 'surfaces' and other three-dimensional graphs

Plot a surface with color lines

```
splot x*x-y*y with line palette
```

Section 26

Some People

`Colin`
: Kelley and Thomas Williams original developers of gnuplot in 1986

`David`
: Kotz developed a version of gnuplot for tex output