graph

A utility for creating plot files

3 Command Line Options

The following table describes each of the command line arguments to graph. Each option which takes an argument is followed by the type and default values of the argument in parentheses.

‘`-C`’ Print out the GNU copyright notice.

‘`-D`’ Read double precision binary data rather than ascii data. Since `graph` is generally I/O bound, this can decrease execution time.

‘`-E`’ Use an extended plot file format. This provides automatic adjustment of labels with respect to tick marks. The ‘`-H`’ and ‘`-W`’ options are ignored if ‘`-E`’ is specified. You will need the GNU plot utility `plot2ps` in oder to print the output.

‘`-H`’ (double, default .03) The fractional height of printed characters. This value is used to adjust location of labels with respect to the tick marks.

‘`-I`’ Read integer binary data rather than ascii data.

‘`-K`’ Change symbol when each new line is stared. A new line is started whenever the abscissal (x) value decreases and the ‘`-b`’ option is specified. The symbol number is incremented through the values show in the table below (see ‘`-S`’).

‘`-L`’ Change line style when each new line is stared. As in ‘`-K`’, the line style number is incremented through the values shown in the table below (see ‘`-m`’).

‘`-M ‘[x|y]’` *`margin`*’

(double, default 0 for x and .05 for y) *margin* is the fractional amount of margin between the data and the horizontal or vertical edges of the box. A value of .05 for y produces a space which is 5% of the height of the box, placed above and below the data.

‘`-M`’ amount of margin between data and x or y axes.

‘`-N` *`ticks`*’ (integer, default 5) *ticks* is the minimum number of tick marks for each axis. `graph` tries to pick as few digits as possible for tick mark labels and uses at least this many tick marks.

‘`-P` *`size`*’ (integer, default 4096) *size* is the upper limit on coordinate values in the plot file `graph` creates. For devices with higher resolution, this value can be increased. A value of 4096 is sufficient for 300 dot per inch devices such as laser writers. This limit on the values of coordinates (0 to *size*) output by `graph` is rigidly enforced to prevent unpredictable behaviors by device drivers such as `plot`.

‘`-S` *`symbol_number symbol_size`*’

(integer and float, defaults -1 and 0.01) Draw a symbol at each point in the data. *symbol_number* specifies the shape of the symbol according to the following table and *symbol_size* specifies the fractional size of the symbol with respect to the height and width of the plot. Note that you can specify sybols to be drawn without any line connecting them by specifying the option `-m -1`.

-1 no symbol at all
0 plus sign
1 cross

2 diamond

3 square

4 triangle

5 circle

6 circle with a line through it

‘`-T` ***tick_size***’

(float, default .01) *tick_size* is the fractional size of the tick marks on each axis. A value of 1.0 produces tick marks on the x (y) axis whose length is equal to the width (height) of the plot.

‘`-W`’ (double, default .02) The fractional width of printed characters. Like the ‘`-H`’ option, this value is used to adjust location of labels with respect to the tick marks.

‘`-X` ***x_label***’

(string, default blank) *x_label* is a label printed below the x axis.

‘`-Y` ***y_label***’

(string, default blank) *y_label* is a label printed to the right of the y axis.

‘`-a` ***step_size lower_limit***’

(floats, defaults 1 and 0) Automaticly generate abscissa (x) values. This option specifies that the data contains only ordinate (y) values. *step_size* specifies the interval between neighboring points and *lower_limit* specifies the first abscissa (x) value.

‘`-b`’ Break the lines in the graph whenever the abscissa (x) values decrease. Several curves with monotonically increasing abscissa values can be concatenated and fed to graph using this option.

‘`-c` ***point_label***’

(string, default blank) *point_label* is the default label placed at each point in the plot when none is read from the input for that point. That is, labels read from the input are printed instead of the default whenever they are present.

‘`-d`’ Debugging information, including the data read in, is sent to the standard error output. This is useful for double checking binary data files.

‘`-g` ***grid_style***’

(integer, default 1) *grid_style* specifies the type of box framing the plot and whether grid lines are drawn inside the box.

0 no box around plot, no axes, no labels.
1 box around plot, axes with tick marks and labels.
2 box containing a grid and axes with tick marks and labels.

‘`-h` ***height***’

(float, default 0.8) *height* specifies the fractional height of the plot with respect to the height of the plotting area. A value of 1.0 will produce a box which fills the available area. Note that the tick marks and labels are outside this area so that values less than 1.0 are generally used.

‘-l *top_label*’

(string, default blank) *top_label* is a label placed above the plot.

‘-m *line_mode*’

(integer, default 0) *line_mode* specifies the mode (or style) of lines drawn between data points.

-1 no line at all
0 solid

1 longdashed
2 dotted

3 disconnected
4 dotdashed
5 shortdashed

‘-r *right*’ (float, default 0.1) Move the plot to the right by a fractional amount *right* with respect to the width of the plotting area. This produces a margin on the left hand side of the plot. A value of 0.5 will produce a margin half the width of the available area. Note that the tick marks and labels are drawn in the margin.

‘-s’ Save the screen. This option prevent graph from erasing the previous contents of the graphics window or device.

‘-t’ Transpose the axes and axis labels. This option flips the plot over by interchanging x and y values and labels.

‘-u *up*’ (float, default 0.1) Move the plot up by a fractional amount *up* with respect to the height of the plotting area. This produces a margin below the plot. A value of 0.5 will produce a margin half the height of the available area. Note that the tick marks and labels are drawn in the margin.

‘-w *width*’ (float, default 0.8) *width* specifies the fractional width of the plot with respect to the width of the plotting area. A value of 1.0 will produce a box which fills the available area. Note that the tick marks and labels are outside this area, so values less than 1.0 are generally used.

‘-x ‘[BTl]’ *lower_limit upper_limit*’

(char, floats) Labels for the tick marks are placed on the top or bottom of the plot when ‘B’ or ‘T’ are specified respectively. The default position is the bottom. ‘l’ is optional and specifies that a logarithmic scale be used for the x axis, by default a linear scale is used. Each of these option letters must be listed seperately, and seperated from the other letters or options by white space characters. The arguments *lower_limit* and *upper_limit* specify the limits of the x axis. By default the upper and lower limits are taken from the data.

‘-y ‘[LRl]’ *lower_limit upper_limit*’

Labels for the tick marks are placed on the left or right of the plot when ‘L’ or ‘R’ are specified respectively. The default position is the right. The other arguments specify the scale and limits of the y axis as those do for the x axis above.

'`-z`' Do not read data from the standard input. You can specify input files on the command line. `graph` prints the file names and point numbers when it encounters error in ascii input files. If input files are named and contain one coordinate per line in the file axis whose length is equal to the width (height) of the plot you can use the emacs `next-error` function to locate the source of the error.

4 graph Examples

4.1 The format of input to graph

graph reads in ascii data. Most often, the data is in the form of pairs of x and y values:

```
0.0  0.0
1.0  0.2
2.0  0.0
3.0  0.4
4.0  0.2
5.0  0.6
```

You can add labels to specific data points by appending a string after the y coordinate. The label ends at the end of the line:

```
3.0  0.4 this is a label for point (3.0, 0.4).
```

Using the '-b' option, you can put more than one data set in the input as long as each is monotonic in x values and decreases from the end of one set to the start of the next:

```
0.0  0.0 first data set
2.0  0.0
4.0  0.2
0.0  0.1 second data set
2.0  0.2
4.0  0.3
```

If your data contains only y values which are equally sampled along the x axis, you can use the '-a' option to specify the x axis values implicitly.

```
0.0
0.1
0.2 label for point (2.0, 0.2)
0.3
0.2
0.3
```

4.2 How to put multiple plots on one page

The command

```
graph -h .4 -w .4 -r .1 -u .1 < ascii_data_file_1 >  plot_file
```

will put a single box containing the plot in the lower left hand quarter of the page. You can add another plot to the upper left hand corner of the page using the command

```
graph -h .4 -w .4 -r .1 -u .6 < ascii_data_file_2 >> plot_file
```

Likewise you can add plots to the right hand side of the page using

```
graph -h .4 -w .4 -r .6 -u .1 < ascii_data_file_3 >> plot_file
graph -h .4 -w .4 -r .6 -u .6 < ascii_data_file_4 >> plot_file
```

The tick marks can be moved inside the box and labels moved to the opposite sides using

```
graph -T -.005 -x T -y L < ascii_data_file >> plot_file
```

4.3 The plot2ps Utility

plot2ps is a utility for converting Unix plot files into postscript. The plot2ps utility reads plotting commands from named files or the standard input and writes postscript to the standard output. You can then print the postscript output on a printer, or edit it using the idraw graphics editor. You can also easily include the output in LaTeX documents using the dvi2ps utility and the LaTeX command psfig.

The source code and documentation for plot2ps and a more complete description of this are available via anonymous ftp from qed.rice.edu (128.42.4.38) in the directory /pub.

Table of Contents