NAME

plot – Use gnuplot to plot one or more data files, and gv to display the resulting .eps file.

SYNOPSIS

${HOME}/bin/plot datafile1 datafile2 ...

DESCRIPTION

If invoked with no arguments, this shell script writes usage information and stops with return code 1. Otherwise it begins by sanity-checking the input file(s): each must have at least 2 lines and contain the same number, either 2 or 3, of columns. A file containing 2 columns is assumed to be x−y data, while a file containing 3 columns is assumed to be x−y−z data. Next the data files are copied to ${HOME}/tmp, with any D exponents changed to E (gnuplot does not interpret D exponents correctly). The script constructs a file of plotting commands in ${HOME}/tmp/plot.gnu, specifying the size of the plot and that the output should be left in ${HOME}/tmp/plot.eps. If the data file(s) have 2 columns, a plot command is added specifying that the input files should be plotted on the same axes with lines; if the data file(s) have 3 columns an splot command is added specifying that the data files should be used together to plot one or more surfaces on the same axes. Next the script pipes a command to gnuplot to load ${HOME}/tmp/plot.gnu. Finally, the script invokes gv with appropriate parameters, to display ${HOME}/tmp/plot.eps on the workstation screen.

DIAGNOSTICS

If everything works this script stops with a return code of 0. If a data file has fewer than 2 lines, or has other than 2 or 3 columns, or if the data files have different numbers of columns, the script writes an appropriate message and stops with return code 1. If gnuplot issues a nonzero return code this script stops with a return code of 2. If gv issues a nonzero return code this script stops with a return code of 3.

BUGS

No provision is made for labeling graphs or graph axes, or for varying the size or position of the graph; to do these things one must edit ${HOME}/tmp/plot.gnu and carry out the final steps of the shell script by hand.

AUTHOR

Michael Kupferschmid

EXAMPLE

```
unix[1] plot datafile1 datafile2
```

If all goes well this command results in the display of a graph having two curves, one representing the data in datafile1 and the other representing the data in datafile2.