Using Gnuplot to sketch graphs

Introduction

Gnuplot is a powerful, open-source graphing tool that can be used to plot functions and data as well as fit curves.

Useful web sites

Duke Gnuplot manual and tutorial A nice introduction to the most common uses of Gnuplot in engineering. <`http://www.duke.edu/~hpgavin/gnuplot.htm`>

Gnuplot home page <`http://www.gnuplot.info/`>

Curve Fitting A nice example of curve fitting with Gnuplot.Click the *Contents* link to see other pages related to Gnuplot. <`http://www.ictp.trieste.it/~manuals/userguide/manual_178.html#SEC177`>

Example Code

The following snippets of code will help you get started.

Plot a function

```
f(x)=exp(x)
plot [-1:1] f(x)
```

Plot a piece-wise function

You can use `?` for writing `if-else` statements to define a function. The `set` command is used to set things like labels, title, etc. You can use the `Symbol` font to display greek letters in text.

```
y1(x)=(x<-100) ? 0 : 0.15*x
y2(x)=(x<100) ? y1(x) : 0
set title "charge density"
set xlabel "x (nm)"
set ylabel "{/Symbol r} (mC/cm^3)"
set nokey
plot [-300:300] y2(x)
```

Plot data

If your data is stored in tab-delimted format in a file called `data.txt` , then you can plot it using

```
plot data.txt
```

If you want to plot a function and your data on the same graph, use

```
y(x)=exp(x)
plot y(x), data.txt
```

Running a Gnuplot script

It's best, I think, to write your Gnuplot code and save it as a `.plt` file. Then run the code from the command line using `>gnuplot myfile.plt` . The one drawback is that the graph window will appear and then disappear. To keep the graph window open, use `pause -1` as shown in this example.

```
y(x)=x**2
plot [-1:1] y(x)
pause -1
```

Printing the graph to a file

To print the graph to a file, perhaps so you can import the file into a document or something, you have to `set output` to the name of the file and `set term` to postscript eps enhanced.

```
y(x)=x**22
set term postscript eps enhanced
set output "plot.eps"
plot [-1:1] y(x)
```

Curve Fitting

GnuPlot can find the parameters for a function that best fits a set of data. You have to give GnuPlot a "guess" for the function that fits the data. It then determines the values of the coefficients for that function. It's easiest, I think, to put the data and the initial guess for parameters into separate files from the main gnuplot file.

For example, let's fit data to a 7th order polynomial. I used Excel to generate the data, and saved the data in tab-delimited format in the file `data.txt` . Here's the data.

```
0 1
0.1 1.101010101
0.2 1.204081633
0.3 1.309278351
0.4 1.416666667
0.5 1.526315789
0.6 1.638297871
0.7 1.752688166
0.8 1.869565199
0.9 1.989010942
1 2.111111
1.1 2.235954815
1.2 2.363635875
1.3 2.494251936
1.4 2.627905261
1.5 2.764702867
1.6 2.904756792
1.7 3.048184367
1.8 3.195108512
1.9 3.345658045
2 3.499968
```

I then created a file `guess.txt` that contained name=value pairs with initial values of the parameters of the function. For a 7th order polynomial, there are 8 coefficients. Here are the initial values.

```
a=1
b=1
c=1
d=1
e=1
f=1
g=1
h=1
```

The gnuplot file defines the function in terms of the parameters given in `guess.txt` , fits the data (i.e. finds the best-fit values of the coefficients in the function), and plots it. Here's the gnuplot code contained in a file I called `polynomial.gpt` .

```
y(x)=a+b*x+c*x**2+d*x**3+e*x**4+f*x**5+g*x**6+h*x**7

fit y(x) "data.txt" via "guess.txt"

set xlabel "x"
set ylabel "y"
set nokey

plot y(x), "data.txt"

pause -1
```

Run the program using the command

```
gnuplot polynomial.gpt
```

The last line uses `pause` so that the graph window will stay open. Of course, you can always output the graph to a file instead of the screen. If you run this code, the output will be

```
Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a               = 1                +/- 2.438e-10    (2.438e-08%)
b               = 1                +/- 5.251e-09    (5.251e-07%)
c               = 0.1              +/- 3.455e-08    (3.455e-05%)
d               = 0.01             +/- 9.459e-08    (0.0009459%)
e               = 0.000999968      +/- 1.288e-07    (0.01289%)
f               = 0.000100028      +/- 9.222e-08    (0.09219%)
g               = 9.9878e-06       +/- 3.318e-08    (0.3322%)
h               = 1.002e-06        +/- 4.728e-09    (0.4718%)


correlation matrix of the fit parameters:

          a      b      c      d      e      f      g      h
a               1.000
b              -0.641  1.000
c               0.451 -0.953  1.000
d              -0.359  0.889 -0.984  1.000
e               0.304 -0.831  0.953 -0.992  1.000
f              -0.267  0.782 -0.920  0.974 -0.995  1.000
g               0.241 -0.741  0.888 -0.952  0.983 -0.997  1.000
h              -0.221  0.706 -0.858  0.930 -0.968  0.988 -0.997  1.000
```

The graph produced by GnuPlot is shown in Figure 1.

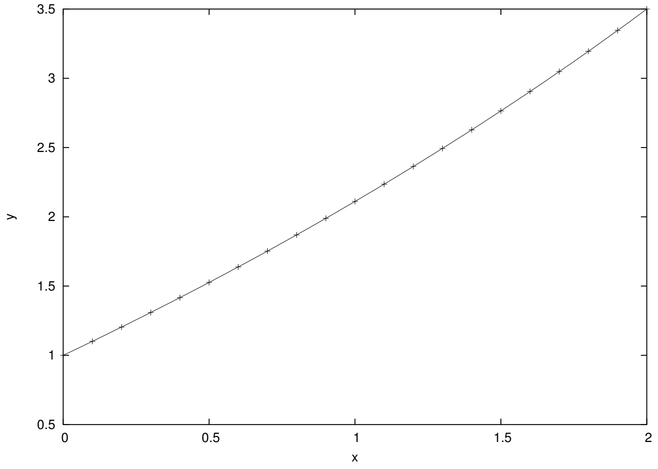

Figure 1: