Finding the Slope of XY Data from a File with Gnuplot

This is a brief introduction by example to using the fitting capability of the GNUPLOT plotting program to find the slope of a range of XY data from a file. We'll use command-line arguments only (you may also have access to a graphical interface). You can find more information from `gnuplot> help fit`, which will tell you all about fitting with gnuplot.

Here is the file `quadratic_eq2.dat`, which was generated by a test program exploring subtractive cancellations in formulas for the roots of a quadratic equation $ax^2 + bx + c = 0$ (based on problem 1 in section 3.4 of Landau and Paez, *Computational Physics*).

```
# Calculation of quadratic equation roots in single precision
#   log(1/ac)    log|rel error 1| log|rel error 2|
  1.000000e-00    -6.360947e+00    -6.310341e+00
  2.000000e+00    -5.753236e+00    -5.732268e+00
  3.000000e+00    -4.455532e+00    -4.455234e+00
  4.000000e+00    -3.850767e+00    -3.850878e+00
  5.000000e+00    -2.867869e+00    -2.868444e+00
  6.000000e+00    -1.334177e+00    -1.313577e+00
  7.000000e+00    -7.164884e-01    -7.927985e-01
#  8.000000e+00     0.000000e+00    inf
```

Shown are the values of $1/ac$ (where $a = 1$ and $b = 2$) and the relative error in the "bad" way of calculating the first and second roots (with respect to the "good" way). However, we have modified the program to now print the logarithm base 10 (log10 in C++) of these quantities. The last line is commented out to avoid problems with the "inf" result. *(Always check your data file before plotting!)*

If we make a quick plot of the second and third columns versus the first with:
`gnuplot> plot "quadratic_eq2.dat" using 1:2, "quadratic_eq2.dat" using 1:3`
we get the same plot as in an earlier handout, but without switching to a log-log plot. Now we can fit a linear function to any region of the plot. (We'll see later how to fit from regions of the log-log plot directly.)

Let's suppose we want to fit a straight line to the region $2 \leq x \leq 6$ of the first data set. It's just two steps: 1) define a function, and 2) give the fit command:

```
gnuplot> f(x) = m*x + b
gnuplot> fit [2:6] f(x) "quadratic_eq2.dat" using 1:2 via m,b
```

which yields some output detailing the iterated fit, then ends up printing the final values (with uncertainties):

```
m                      =  1.04258           +/- 0.08521          (8.173%)
b                      = -7.82263           +/- 0.3615           (4.622%)
```

So the slope m is one within errors, as expected. The values of m and b are saved, so we can make a plot with the original points *and* the fit line using:
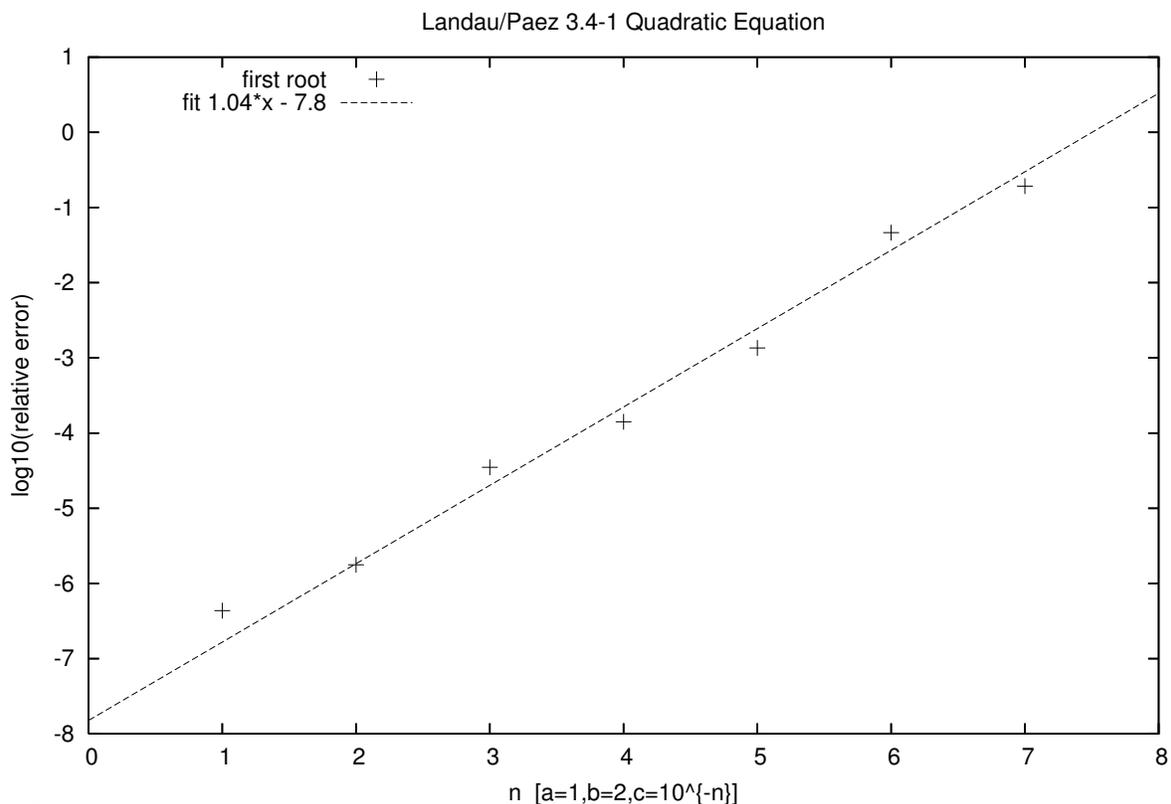
```
gnuplot> plot "quadratic_eq2.dat" using 1:2, m*x+b
```

Here is a complete example and the resulting plot:

```
gnuplot> set title 'Landau/Paez 3.4-1 Quadratic Equation'
gnuplot> set xlabel 'n   [a=1,b=2,c=10^{-n}]'
gnuplot> set ylabel 'log10(relative error)'
gnuplot> set xrange [0:8]
gnuplot> set pointsize 1.5    # set the size of the plotted points
gnuplot> set key top left     # move the key away from the points
gnuplot> set timestamp        # turn on a date/time indicator
gnuplot> f(x) = m*x + b       # define a linear function
gnuplot> fit [2:6] f(x) 'quadratic_eq2.dat' using 1:2 via m,b  # do the fit
gnuplot> plot 'quadratic_eq2.dat' using 1:2 title 'first root',\
> m*x+b title 'fit 1.04*x - 7.8'       # plot points and line
gnuplot> set out 'quadratic_eq2.ps'   # an output postscript file
gnuplot> set terminal postscript       # switch to postscript mode
gnuplot> replot                        # plot to the file
```


Landau/Paez 3.4-1 Quadratic Equation

Thu Mar 27 08:58:33 2008