Data Fits

Introduction

Most experiments involve several variables that are interdependent, such as force (F) and displacement (X) for a spring. A range of F will produce a range of X, with proportionality constant "k", as described by Hooke's law F(X) = -kX. Each measurement of the variables (F,X) has its own mean value and errors as described earlier in the "DataAna" handout. Now we want to combine these functional data to determine a mean value and error for the parameter "k". Other situations may be more complex. Thus we have, in general, some function f(v1,v2,v3,...c1,c2,c3,...) that relates multiple variables (v_i) and constants (c_i).

The experimental goal is to assess whether the assumed function is a good fit to the data, and, if so, determine the best value and uncertainty of the constants.

We begin this discussion using a simple linear function. Later, we briefly elaborate to linearized plots and and non-linear fits.

Linear Regression

Consider an experiment in which students measure force vs. stretch for a vertical spring. Force is applied by hanging weights, calibrated to ±1gm. Stretch is measured by sighting a ruler to an accuracy of ± 5mm. They obtain data as shown in Fig. 1. By convention we will refer to the horizontal axis as "X" and the vertical axis as "Y". Also by convention, we plot the independent variable on X and the dependent variable on Y. The linear regression routine assumes that the X-values are error free. In practice this means that

the variable with the larger (percentage) errors goes on the Y-axis.

In the present experiment, we see that the data for stretch show the larger % error, so we put stretch on the Y-axis.

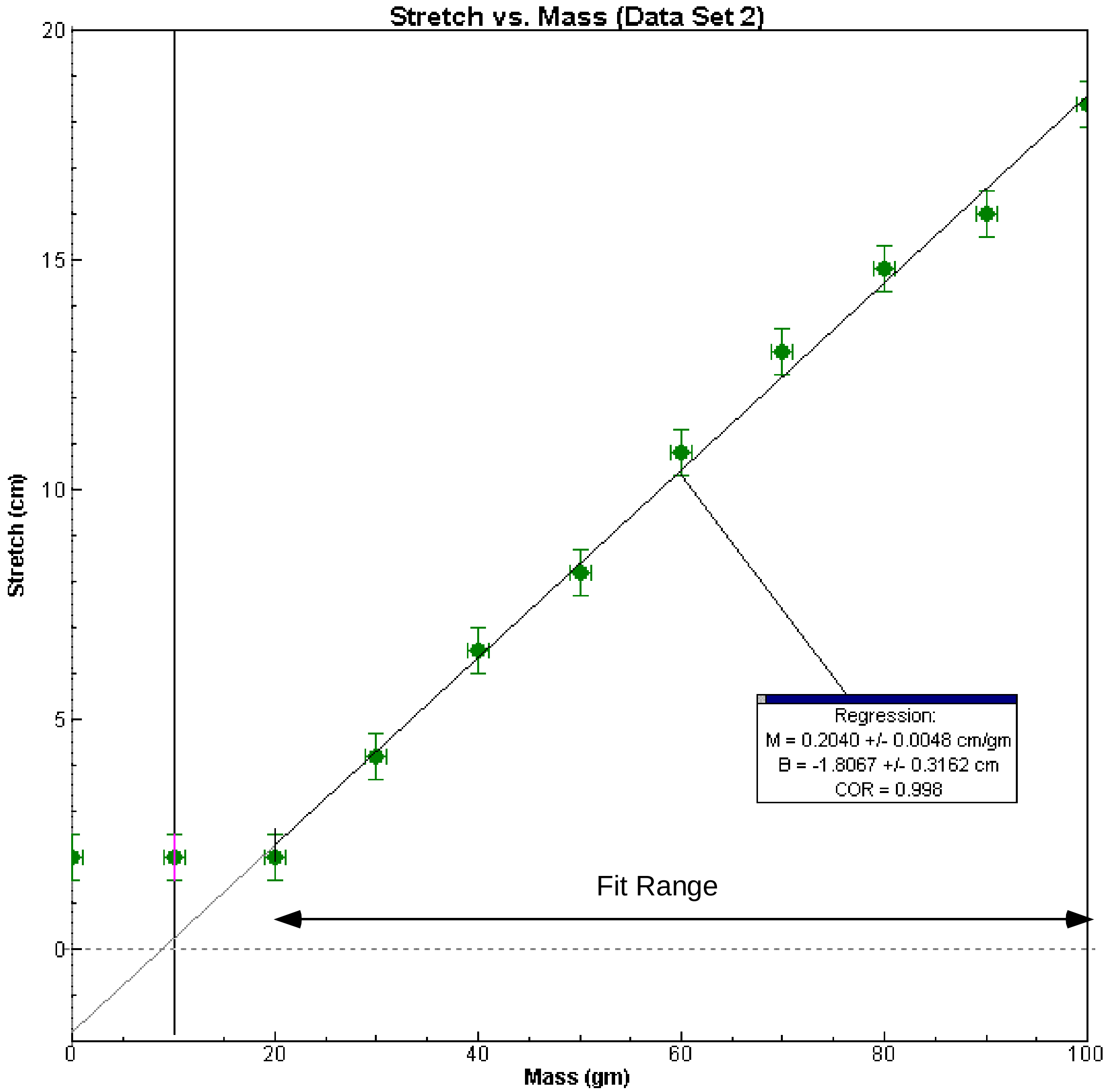

Fig. 1 Linear fit to spring data.

The students recall that a simple spring should follow a linear behavior given by Hooke's Law

$$F(X)=kX \qquad \text{Eq. 1}$$

where F is the applied force (opposite the force by the spring on the object), X is the displacement from the equilibrium position, and "k" is the spring constant. They further realize that F = Mg, where M is the applied weight. So their model for the experiment is

$$M(S) = (k/g)\,(S-S_0) \qquad \text{Eq. 2}$$

Notice the structure of Eq. 2: We have two variables (M, S) which take a range of values, and two parameters (k, S_0) which are constants for this experiment. The aim is to find the values of parameters such that the function is a "best fit" to the data. That is: find the function (straight line in this case) that passes closest to all the data points. This is done by minimizing the root-mean-square of deviations (Y_{calc}-Y_{data}).[1,2] This "linear regression" operation is available in all analysis programs as a canned macro. The fitting function is written in generic form as

$$y(x) = mx + b \qquad \text{Eq. 3}$$

This is a linear function with variables (x,y) and constants (m,b). The program spits out the best fit values (and errors!) as

$$m = m_0 \pm \Delta m = 0.204 \pm 0.005 \text{ cm/gm}$$
$$b = b_0 \pm \Delta b = -1.8 \pm 0.3 \text{ cm} \qquad \text{Eq. 4ab}$$

Here, "m" is the slope and "b" the y-intercept of a straight line. Notice that the students properly exclude the data below 20 gm from the fit, since they recognize that "something else is happening" there.

Before interpreting these results, we first ask

Is this model a good fit to the data?

This is a subtle but very important question. The model function is a good fit to data if no *systematic* errors are apparent. Systematic errors show some pattern of deviations between the data (points) and the function (line). For example, if you find that points near the center of the range tend to be too high (above the line), this is a systematic error. It suggests that the theory or experiment or both are missing something or contain extraneous effects. Conversely, if the deviations are random, (ie: there is no discernible systematic deviation), then the model function is a good fit to the data. This evaluation depends entirely on the comparison of systematic vs. random errors. It is always possible, in principle, to reduce random errors far enough

[1] D. Preston and E. Dietz, The Art of Experimental Physics J. Wiley (1991).
[2] P. R. Bevington and D. K. Robinson, Data Reduction and Error Analysis for the Physical Sciences, McGraw Hill (1992).

to expose systematic errors. The hard part is to identify and correct the source of systematic error (whether theory or experiment). New laws of physics may be lurking. Such was the case for Einstein's theory of special relativity. Ironically, it is also possible to make random errors large enough so that any model function will be a "good fit" to the data. This has been proven many times in the Introductory Labs!

Back now to interpretation of the fit results, Eq. 4ab. Note first that we must invert the function M(S) to write S(M), reflecting the fact that "S" is the dependent variable (goes on the Y-axis). Thus we have

$$S(M) = (g/k)M + S_0 \qquad \text{Eq. 5}$$

Comparing with the model function, y(x) = mx+b, we see that the slope and intercept are given by

$$\text{(slope)} = (g/k) = 0.204 \pm 0.005 \text{ cm/gm} \qquad \text{Eq. 6}$$

$$\text{(y-intercept)} = S_0 = -1.8 \pm 0.3 \text{ cm} \qquad \text{Eq. 7}$$

Note that each of these values carries units and uncertainty. You must carry these through your calculations! S_0 is already in final form, so no further work is needed. Determination of "k" requires a little more work, however. We solve for "k" in equation 6 as

$$k = g/\text{(slope)} = 48.0 \text{ gm/sec}^2 = 48.0 \times 10^{-3} \text{ N/meter} \qquad \text{Eq. 8}$$

Note that we have converted gm/sec^2 to N/m, which are the more conventional units for spring constant. The error in k is given as

$$\left(\Delta k/k\right)^2 = (\Delta g/g)^2 + \left[(-1)\left(\Delta slope/slope\right)\right]^2 \qquad \text{Eq. 9}$$

The two terms are

$$\Delta g/g = 0 \qquad \text{Eq. 10}$$

$$|(-1)(\Delta\text{slope})/\text{slope})| = 0.0048/0.204 = 0.0235 \qquad \text{Eq.11}$$

Thus we have

$$\Delta k = (0.023)*(48) = 1.1 \text{ gm/sec}^2 \qquad \text{Eq.12}$$

Here we have used special case I and II from the DataAna handout (simple product and power law with m = -1). The final results are stated as

$$S_0 = -1.8 \pm 0.3 \text{ cm} \quad (16\%)$$
$$k = 48\pm1 \times 10^{-3} \text{ N/meter} \quad (2\%) \qquad \text{Eq.13ab}$$

Finally we ask what these results mean, in particular the errors. Comparing with the plot, we note that the value for S_0 agrees with the visual y-intercept of the line. "k" is the inverse slope of the line, and corresponds to the stiffness of the spring. A stronger spring would have a smaller slope. The kink at 20 gm corresponds to a preset force that must be exceeded before the spring begins to stretch. This example could have been simplified somewhat by plotting force (=mg) instead of mass on the X-axis, since "g" is a well-known constant. We have shown it this way to demonstrate the propagation of errors, which must be done in general. Regarding errors in the fit values, it is important to realize that

the errors in the fit values have nothing to do with the data error bars!

Instead, they are determined by the mean values of the data points. They correspond nominally to the range of values for slope and intercept that would bracket 2/3 of the data points. Some analysis programs indicate this by drawing a wide line/wedge for the fit function result.

Linearized fits

Model functions are often more complex than the simple linear relation y(x) = mx +b. Data in such cases can still be fit by linear regression if the model function can be linearized. For example the model function $f(q) = Cq^n$ can be linearized by taking the n^{th} root of both sides, as $[f^{1/n}] = C^{1/n}$ [q], where we have used [] to indicate that the structure is linear: that is we tabulate, plot and fit using $y=[f^{1/n}]$ and x = q. The usual procedure for interpreting slope and intercept and their uncertainties apply, as before. Linearizing involves algebraic manipulation and/or function inversion. The resulting units on the X and Y-axes may look strange, but you must carry them though your calculations without fear. Units are your friend (!) since they help identify mistakes in math. Another example is $f(q) = C/(q+q_0)$. This is linearized as $[1/f] = [q]/C + q_0$, so we have slope =1/C and intercept q_0. Another common example is the pure exponential function $f(q)=\exp(-C/q)$. This is linearized as $[\ln(f)] = -C\ [1/q]$. Note that the original variables f, q appear on neither axis in this case.

In Fig. 2ab we show an example of a linearized plot of $y(x) = 1/(x+x_0)^2$. Note the “funny”, but correct units of the linearized variable. Fitting the original function y(x) would require a non-linear fit (see below). The nature of the model function and quality of the fit is not readily apparent by inspection of this plot. On the other hand, it is immediately apparent from the linearized plot that the data are a good fit to the model function, since the human eye/brain can readily visualize a straight line. From the standard linear regression routine, we get values for slope and intercept, plus well-defined error bars for each.

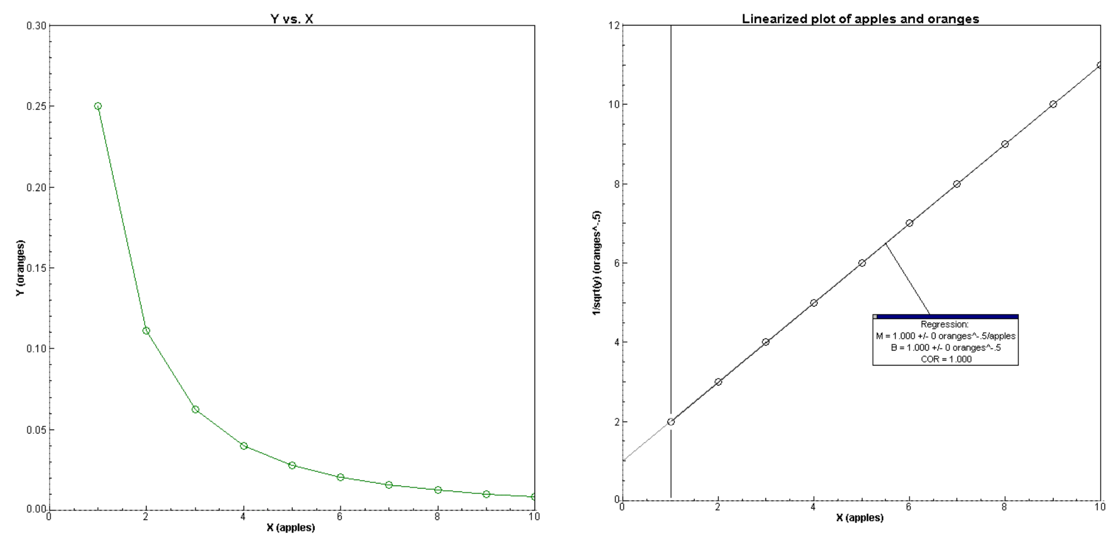

Fig. 2 (a) Plot of $y(x) = 1/(x+x_0)^2$. The connecting line is only a guide to the eye. (b). Linearized plot and fit.

Non-linear fits

Not all functions can be linearized. An example is $f(q) = Aq^2 + Bq$. This requires a non-linear fit. In Graphical Analysis this is done by either selecting a canned function from a list, or writing your own function. The procedure and interpretation follow in detail. Menu steps refer to the Windows GA program.

1. Manual fit in GA: Select range of data (all or part); ANALYZE; choose a pre-defined function or write your own with OTHER. Type the function in the box for y(x), using "X" for the independent variable (regardless of what label you gave it). GA assumes that any other letters/words are adjustable constants. GA is stupid about parsing expressions so use plenty of parentheses. If you start with bad guesses, the theory line will be way off – probably not even visible in the range of the graph. Adjust the constants to get a reasonable eyeball fit, and write these down on paper. You can come somewhat closer by minimizing the Mean Square Errors (MSE) which are shown with the fit results.
2. Auto fit in GA: Select data range \ analysis \ automatic. GA starts autofit with all values = 1.0. This has two potential problems: the function curve is probably not visible on the chart, and worse, the fitting routine may go on a "wild goose chase", diverging from the true solution. You need to "pause" the fit, type in your guesses (now it looks like a manual fit), then resume. Once you are close, GA will converge quickly to an optimal fit.
3. Errors for non-linear fits: The computer shows "± errors" for each variable while it is searching, but these are meaningless and disappear after you stop the fit. GA does show "Mean Square Errors" (MSE) after the fit. This can be used to extract ±errors in the variables by manual manipulation. First find the optimal fit using automatic fit, and write down the corresponding MSE. Now manually tweak each variable slightly off its optimal value until the MSE is twice the minimal value you wrote down. You can tweak either plus or minus, it won't matter much. Repeat this procedure for each variable in the function, always starting from the original optimal values and MSE. This process does not take long if you are systematic, and its kind of fun. You need not obsess about precise doubling of MSE, since this is a statistical value, anyway.