LEAST SQUARES: FITTING A CURVE TO DATA POINTS

1. An example to illustrate the motivation

We illustrate the method of the *least squares* fitting of a curve (here a straight line) to a set of data points by considering a classic experiment from introductory physics, in which a spring is hung from a rigid support, and a mass M is hung on the spring. If the mass is pulled down and released, the system will oscillate with a period T given by the expression

$$T = 2\pi(M/k)^{1/2}$$

where k is the spring constant. The formula is correct if the spring has negligible mass. If the spring mass is not negligible, we can revise the simple formula to take the mass of the spring into account:

$$T = 2\pi[(M+m)/k]^{1/2} \tag{1}$$

where m is the “effective” mass of the spring. A somewhat questionable theoretical calculation predicts that m should be one third of M_s, the actual mass of the spring.

We now think of doing an experiment to test this hypothesis, and so load the spring with a number (say N) of different masses, $M_1, M_2, \ldots, M_N$, and measure the associated periods $T_1, T_2, \ldots, T_N$.[1] We may use these N data points to test the theoretical relationship given by Eq. 1.

We could plot our experimental points on a graph of T vs M, but if Eq. 1 holds, our experimental points would fall on a curved line, and it would be difficult to tell whether or not the functional form suggested by Eq. 1 actually describes the data. However if we plot our points on a graph of T^2 vs M, they should fall on a straight line if Eq. 1 is a good description of the system:

$$T^2 = \frac{4\pi^2}{k}M + \frac{4\pi^2 m}{k} = \alpha M + \beta \tag{2}$$

where $\alpha = 4\pi^2/k$ is the slope of the line and $\beta = 4\pi^2 m/k$ is its intercept with the T^2 axis. If the data do in fact fall on a straight line for this plot, we can estimate the slope and intercept of the line to provide estimates for the two parameters m and k. Note that for

[1] We number our points from 1 to N, because it is traditional, natural, and notationally elegant. Such a vector is called a *unit-offset* vector. However in the C language, which we use to do the calculations we shall soon describe, a vector such as $M[i]$ is traditionally *zero-offset*, that is, i would run from 0 to $N-1$. If you allocate memory for a nine-dimensional vector it will have components $M[0]\ldots M[8]$, and you may well get garbage, or perhaps a “segmentation fault”, if you ask for $M[9]$. While it is possible to twiddle the C code to handle unit-offset vectors and matrices, we do not do so. In referring to our programs you will just have to make the mental adjustment to shift the offset. We trust this will cause no confusion.

our example, the ratio of the intercept to the slope of the line should equal m if the theory is correct.

The table below and the associated plots of Fig. 1 show the data from an actual experiment, in which a spring is suspended from a support and the oscillation period is measured for each of 9 different masses, ranging from 55 to 455 grams, which are hung on the spring. The mass of the spring itself was measured to be approximately 48.2 grams.

M (grams)	55	105	155	205	255	305	355	405	455
T (sec)	.496	.645	.761	.867	.957	1.037	1.113	1.194	1.254
T^2 (sec^2)	.246	.416	.579	.752	.916	1.075	1.239	1.426	1.573

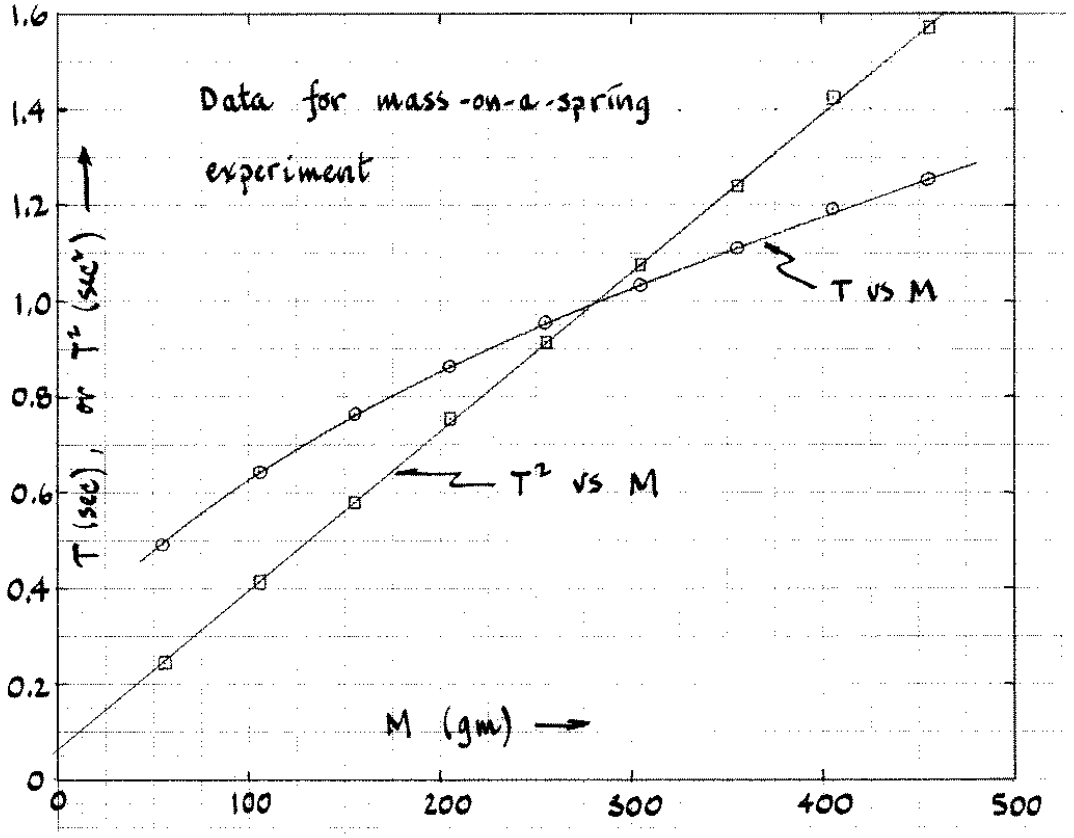

Figure 1 — The straight line is to be preferred where possible.

Note that when we plot T_i versus M_i, the experimental points fall on a curve, but that when we plot T_i^2 versus M_i, the points appear to fall on a straight line, as we might expect from the theoretical hypothesis represented by Eq. 2. As we have mentioned, this straight line is much easier to analyze than the curve.

Now the question arises: What are the "best" values of m and k that can be determined from the data? Furthermore, once we have determined the "best" values for m and k, can we say that the data are consistent with the hypothesis that $m = M_s/3$?

We can make some initial estimates by hand, just by drawing, by eye, a straight line through the data points (as shown on the graph of T^2 versus M), and reading off the slope and the intercept of this straight line. Thus, we can read an intercept of about 0.063 sec^2, and a slope of about $(1.390 - 0.063)/400 = 3.318 \times 10^{-3}$ sec^2/gram, and hence a ratio of the intercept to the slope (this will be m) of about 19 grams. The theoretically predicted value of m is about 16.1 grams—one third of the actual spring mass. To say whether these two values agree with each other, we need to estimate the uncertainty associated with our 19-gram result. We can make a rough estimate of this uncertainty by shifting the straight edge on the graph while estimating (subjectively) whether the straight edge still goes through the points, and thus deduce that m is accurate to within a few grams.

We can, however, be still more quantitative, by making a *least squares* straight line fit to the data. Such a fit is also called a *linear regression* by the statisticians.

In the following section we discuss the general methods for fitting a straight line to a set of data points. Later on, we'll discuss the more general problem of fitting *any* hypothesized function to a set of data points. We include details of the derivations, since they are not readily accessible elsewhere. For further reading, consult the references listed at the end of the chapter.

2. General discussion of the least squares method: Fitting a straight line

We consider the general problem of fitting a straight line, of the form $f(x) = \alpha x + \beta$, to a set of N data points (x_i, y_i), where i goes from 1 to N. Our hope is that y_i will be well approximated by $f(x_i)$. (In our example, $x_i = M_i$, and $y_i = T_i^2$, so we hope that T_i^2 will be well approximated by $\alpha M_i + \beta$).

We assume that the x_i are known exactly, and that for each x_i there is a normally distributed population of observations y_i, whose mean is $\alpha x_i + \beta$, and whose variance is σ_i^2.[2]

If the x_i are not known exactly, but the y_i *are* known exactly, we can just reverse the roles of x_i and y_i. If neither x_i nor y_i are known exactly, a unique straight line fit is considerably more difficult.[3] In practice, we choose the independent variable x_i to be the most precisely measured quantity. Thus in our example of the previous section, we shall take the masses M_i to be known exactly.

[2] In the ideal situation, independent estimates for σ_i^2 would be obtained from the data. For an experiment such as the one we describe here, this would require that several values of y_i be measured for each x_i, so that a sample variance in y may be calculated for each x_i. For some kinds of experiments, especially in particle physics, the y_i are numbers of counts, as registered, say, by a Geiger counter. If we make the reasonable assumption that such y_i obey Poisson statistics, an estimate of σ_i is available: $\sigma_i \approx \sqrt{y_i}$, since for a Poisson distribution the variance is equal to the mean. More commonly, an estimate of σ_i is not available, in which case we must make some assumption about its value. Often it is assumed that all the σ_i are equal, so that each data point is considered to be weighted equally.

[3] See Reference 3 at the end of this chapter.

The overall situation may be shown graphically:

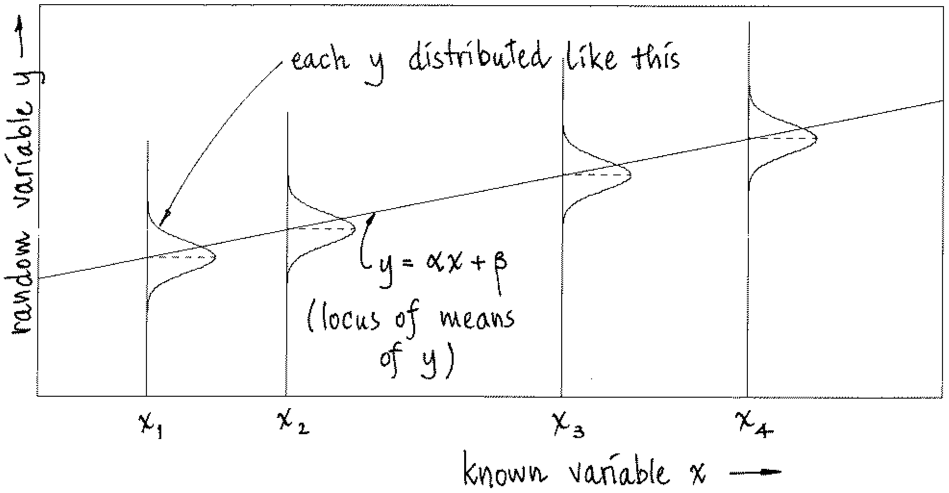

Figure 2 — Graphical representation of the model for linear regression.

Now we hypothesize a straight line of the form

$$f(x) = ax + b \tag{3}$$

Our job is to determine values for a and b that are best estimates for the unknown population parameters α and β. To do this, we form the quantity Φ, the weighted sum of the squares of the residuals. It will be a function of a and b:

$$\Phi(a,b) = \sum_{i=1}^{N} w_i r_i^2 = \sum_{i=1}^{N} w_i [f(x_i) - y_i]^2 = \sum_{i=1}^{N} w_i (ax_i + b - y_i)^2 \tag{4}$$

Here $r_i \equiv f(x_i) - y_i$ is the *residual* for the i^{th} data point. The concept of a *residual* may be illustrated with a graph:

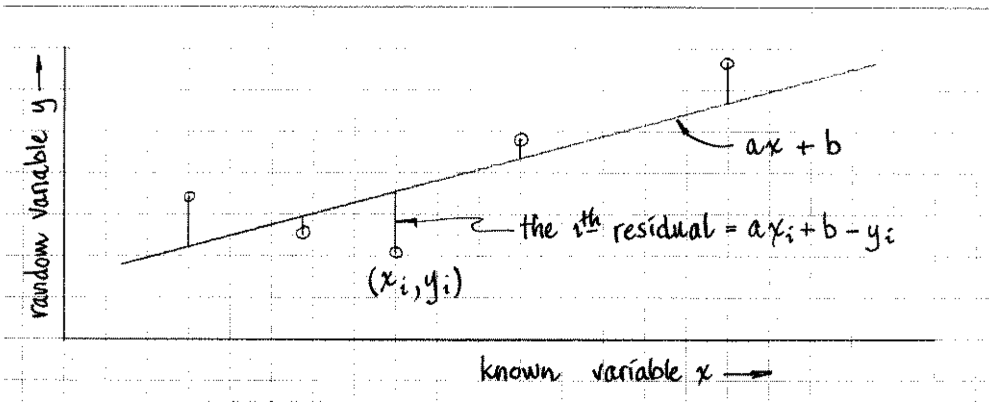

Figure 3 — Graphical representation of residuals.

There are N such residuals.

w_i is the *weight* to be given to the i^{th} data point. It will be proportional to the inverse of the variance (*i.e.*, proportional to $1/\sigma_i^2$) for that point. If w_i is taken to be *equal* to $1/\sigma_i^2$, where σ_i^2 is *independently* estimated for each point, the quantity Φ becomes equal to χ^2 (*chi-square*), a useful statistical quantity. For this reason a "least-squares fit" is sometimes called a "chi-square fit".

In practice, we shall often know only the *relative* weights, not having available independent estimates of the σ_i^2. If we know only that all σ_i are equal, we may take $w_i = 1$ for each point.

Note that Φ is a function of a and b, the parameters to be determined. The best values for these parameters will be those for which Φ is a minimum. This is the meaning of *least squares*. Moreover, the smaller $\Phi_{\min}$ is, the better the fit.[4]

To minimize Φ, we differentiate Φ with respect to a and b, and set each of those derivatives equal to zero. That is, we set

$$\frac{\partial\Phi}{\partial a} = 0\,; \qquad\qquad \frac{\partial\Phi}{\partial b} = 0$$

Working out the derivatives, we find

$$\begin{aligned} X_2\, a + X_1\, b &= P \\ X_1\, a + W\, b &= Y_1 \end{aligned} \tag{5}$$

where we represent relevant sums like this:

$$\begin{aligned} W &\equiv \sum w_i\,; & X_1 &\equiv \sum w_i x_i\,; & Y_1 &\equiv \sum w_i y_i \\ X_2 &\equiv \sum w_i x_i^2\,; & Y_2 &\equiv \sum w_i y_i^2\,; & P &\equiv \sum w_i x_i y_i \end{aligned}$$

with all sums running from $i = 1$ to N.

Equations 5 may be solved[5] for a and b:

$$a = \frac{1}{\Delta}(W \cdot P - X_1 \cdot Y_1) \qquad b = \frac{1}{\Delta}(X_2 \cdot Y_1 - X_1 \cdot P) \tag{6}$$

where Δ is the determinant of the coefficients:

$$\Delta \equiv W \cdot X_2 - X_1^2$$

[4] The reasoning leading to the statement that the best fit is obtained by the method of "least squares" (*e.g.*, by minimizing chi-square) stems from the consideration of *maximum likelihood estimators*. Chapter 15 of *Numerical Recipes* and Appendix 5A of the book by Bennett and Franklin contain good discussions of this topic.

[5] As an exercise, try deriving Eqs. 5 and solving them to get Eq. 6.

With these values of a and b, Φ assumes its minimum value:[6]

$$\Phi_{\min} = Y_2 - aP - bY_1 \tag{7}$$

We have thus determined values for a and b, the best estimates of the slope and intercept. We must now estimate the statistical uncertainties, or *confidence limits* for each parameter. Without confidence limits our results are meaningless.

The idea is this: a and b are each functions of the y_i. Hence statistical fluctuations in the y_i will lead to statistical fluctuations in a and b. Later on (see Fig. 4 on page 3–20) we shall provide a concrete illustration of such fluctuations. For now, we simply note that if σ_i^2 is the variance of y_i, the variances of a and b will be, according to the theory described for the propagation of uncertainty in the preceding chapter,[7]

$$\sigma_a^2 = \sum \left(\frac{\partial a}{\partial y_i}\right)^2 \sigma_i^2 \qquad \text{and} \qquad \sigma_b^2 = \sum \left(\frac{\partial b}{\partial y_i}\right)^2 \sigma_i^2 \tag{8}$$

Now $\sigma_i^2 = C/w_i$, where C is some constant, and

$$\frac{\partial a}{\partial y_i} = \frac{w_i}{\Delta}(Wx_i - X_1)\,; \qquad \frac{\partial b}{\partial y_i} = \frac{w_i}{\Delta}(X_2 - X_1 x_i)$$

This leads to

$$\sigma_a^2 = C\frac{W}{\Delta}\,; \qquad \sigma_b^2 = C\frac{X_2}{\Delta} \tag{9}$$

What is C? If we are fortunate enough to have available independent estimates s_i^2 of the y-variances σ_i^2, then $w_i \approx 1/s_i^2$, that is, $C \approx 1$. In that case our best estimates for σ_a^2 and σ_b^2 are just

$$s_a^2 = \frac{W}{\Delta}\,; \qquad s_b^2 = \frac{X_2}{\Delta} \tag{10}$$

The square roots of these quantities are estimates of the standard deviations of each of the parameters, providing confidence limits of approximately 68 per cent.

Furthermore, since it is known that $C \approx 1$, the value of $\Phi_{\min}$ is equal to a sample value of χ^2, a statistical quantity that may be used to test "goodness of fit", that is,

[6] This compact expression for the minimum value of Φ can be deduced by writing Φ in the form

$$\Phi = \sum w_i(ax_i + b - y_i)r_i = a\sum w_i x_i r_i + b\sum w_i r_i - \sum w_i y_i r_i$$

The first two sums on the right side of this equation will vanish by virtue of Eqs. 5, leaving only the last sum, which is the right side of Eq. 7.

[7] Estimates of the variances σ_a^2 and σ_b^2 lead to "one-parameter" confidence limits—those discussed here. Such confidence limits ignore any correlation between a and b. Later on in this chapter (see page 3–18) we discuss the consequences of taking into account such correlations.

whether the data may be appropriately described by a straight line. For more details on the use of χ^2, see the following chapter (Chapter 4).

More frequently, estimates of only the *relative* weights are available, in which case the best we can do is to estimate the value of C. We do this by *assuming* that our data are well-fit by the straight line, that is, we assume that χ^2 is equal to its mean value, which turns out to be the number of "degrees of freedom", or $N - 2$.[8] Since $\Phi_{\min} = C\chi^2$, we simply replace C by $\Phi_{\min}/(N-2)$, so that our best estimates for σ_a^2 and σ_b^2 become

$$s_a^2 = \frac{W}{\Delta}\frac{\Phi_{\min}}{(N-2)} \qquad \text{and} \qquad s_b^2 = \frac{X_2}{\Delta}\frac{\Phi_{\min}}{(N-2)} \tag{11}$$

The square roots of the quantities given by either Eqs. 10 or 11 are estimates of the standard deviations of each of the parameters, which provide confidence limits of approximately 68 per cent.

Thus our results may be stated succinctly:[9]

$$\text{SLOPE} = a \pm s_a\,; \qquad\qquad \text{INTERCEPT} = b \pm s_b \tag{12}$$

3. About the use of standard "Linear Regression" statistics routines

The usual "statistics" software routines available for use on personal computers do not usually calculate the standard deviations s_a and s_b of the slope and intercept of the straight line resulting from a "linear regression" calculation. Instead, they provide a calculation of either the *covariance* s_{xy} and/or the *correlation coefficient* $r = s_{xy}/s_x s_y$, where[10]

$$s_x^2 = \frac{1}{N-1}\sum(x_i - \overline{x})^2\,; \qquad s_y^2 = \frac{1}{N-1}\sum(y_i - \overline{y})^2$$

and

$$s_{xy} = \frac{1}{N-1}\sum(x_i - \overline{x})(y_i - \overline{y})$$

Such a procedure is more general than that we have been considering, in that the *covariance* and the *correlation coefficient* may be calculated (and have meaning) even if the x_i are *not* known exactly.

[8] The number is $N-2$ and not N because 2 degrees of freedom have been used up in the determination of a and b.

[9] The expressions given by Eq. 12 are not quite correct, for reasons similar to those noted in the footnote on page 2 – 10. To be correct, we should include the appropriate "Student" t-factor t_{N-2}, which depends on the number of data points and the confidence level chosen. Hence the correct forms for Eqs. 12 are:

$$\text{SLOPE} = a \pm t_{N-2}s_a\,; \qquad \text{INTERCEPT} = b \pm t_{N-2}s_b \tag{12a}$$

The least-squares program `fitline` described in Chapter 5 incorporates "Student" t-factors in this way. A table of "Student" t-factors appears on page 2 – 13.

[10] Weighting of points is implicit in these expressions.

However, if the x_i are assumed to be known exactly, s_a and s_b may be expressed in terms of the correlation coefficient r. Here are the appropriate equations:

$$s_a^2 = \frac{a^2}{N-2}\left(\frac{1}{r^2}-1\right) \; ; \qquad s_b^2 = \frac{X_2}{W} s_a^2 \tag{13}$$

with the correlation coefficient being

$$r \equiv \frac{s_{xy}}{s_x s_y} = \frac{W \cdot P - X_1 \cdot Y_1}{(W \cdot X_2 - X_1^2)^{1/2}(W \cdot Y_2 - Y_1^2)^{1/2}}$$

Note that as $r \rightarrow \pm 1$, the correlation becomes exact, and both s_a and $s_b \rightarrow 0$. On the other hand, as $r \rightarrow 0$, y_i becomes uncorrelated with x_i, and both s_a/a and s_b/a increase without bound, as expected.

4. Continuing with our example, involving the mass on the spring

We start with the raw data, given in the table on page 3 – 2. We assume that σ_T^2, the variance in T, is independent of T, *i.e.*, the same for all the data points. Hence the variance in T^2 will *not* be the same for all data points, and if we want to perform a proper least-squares fit to the straight line of T^2 versus M, we should weight each point with its appropriate weight w_i.

(Actually, in the majority of cases any physicist will encounter, the fit obtained, assuming that all points are equally weighted, will not be very different from that in which correct weights are included. We do this example, however, including the correct weighting factors, to illustrate the method in detail).

To determine the w_i, let $y = T^2$. Then

$$\sigma_y^2 = \left(\frac{\partial y}{\partial T}\right)^2 \sigma_T^2$$

as described on page 2 – 14. Now $\partial y/\partial T = 2T$, so that

$$\sigma_y^2 = 4T^2\sigma_T^2 \qquad \text{and} \qquad \frac{1}{\sigma_y^2} = \frac{1}{4T^2}\frac{1}{\sigma_T^2}$$

Now since w_i is proportional to $1/\sigma_y^2$, w_i will be proportional to $1/T_i^2$, and in fact may be taken equal to $1/T_i^2$, since for these data an estimate of σ_T^2 is not available.

With a simple computer program[11] we may calculate the sums

$$W = \sum w_i\,, \quad X_1 = \sum w_i x_i\,, \quad Y_1 = \sum w_i y_i\,, \quad X_2 = \sum w_i x_i^2\,, \quad \text{and} \quad P = \sum w_i x_i y_i$$

and then, using Eqs. 6 and 11, calculate the values of a, b, s_a and s_b. The results, using the measured data from page 3 – 2, are

$$a = \text{SLOPE} = \frac{4\pi^2}{k} = (3.331 \pm 0.015) \times 10^{-3} \quad \text{sec}^2/\text{gm}$$

and

$$b = \text{INTERCEPT} = \frac{4\pi^2 m}{k} = 0.0642 \pm 0.0032 \quad \text{sec}^2$$

These expressions for the slope and intercept, complete with the standard deviations s_a and s_b, imply that the “true” values of the slope and intercept have about a 68 per cent chance of falling within the specified limits, assuming that the experiments were carried out without any *systematic* error such as a miscalibration of a stopwatch.

Now, what do these values of a and b imply for k and m? We note that $k = 4\pi^2/a$. What is δk, the uncertainty in k, in terms of s_a?

Again we propagate uncertainty through the functional relationship to find

$$(\delta k)^2 = \left(\frac{\partial k}{\partial a}\right)^2 s_a^2$$

so

$$\delta k = \frac{4\pi^2}{a^2} s_a$$

Hence we obtain the result for our measured spring constant:

$$k \pm \delta k = (1.1852 \pm 0.0053) \times 10^4 \quad \text{dynes/cm}$$

Similarly, since $m = b/a$,

$$(\delta m)^2 = \left(\frac{\partial m}{\partial a}\right)^2 s_a^2 + \left(\frac{\partial m}{\partial b}\right)^2 s_b^2 = \frac{b^2}{a^4} s_a^2 + \frac{1}{a^2} s_b^2$$

leading to a measured effective mass for the spring of

$$m \pm \delta m = 19.27 \pm 0.97 \quad \text{grams}$$

Finally we ask, are the data consistent with the hypothesis that $m = M_s/3$? Since the measured $M_s \approx 48.2$ grams, $M_s/3 \approx 16.07$ grams. This value is not bracketed by the uncertainty in our experimentally deduced value of m, and in fact is less than our deduced value of m by over three standard deviations. Therefore, we conclude that the value of m derived from the data is *inconsistent* with the hypothesis that $m = M_s/3$.

Either (a) the hypothesis is incorrect, or (b) there is a systematic error in the measurements. We might want to check the accuracy of our apparatus, or try the experiment again, or rethink the theoretical analysis.

[11] See, for example, the program `fitline` described in Chapter 5.

5. Fitting curves nonlinear in the parameters: the Marquardt algorithm[12]

The *least-squares* method is not limited to fitting a straight line, that is, a linear combination of 1 and x, to a set of data points. The method may be generalized to fit either (a) a linear combination of *any* K specified functions of x, or (b) *any* function of x and a specified number of parameters, even one nonlinear in those parameters. The former may be accomplished in one step through the simple inversion of a $K \times K$ square matrix, while the latter requires an iterative technique, one that we describe below.

In practice, an iterative method may be used for both (a) and (b), and the particular method we shall describe—the Marquardt algorithm—is widely used as a technique for modeling data in a broad range of situations.

For example, when silver is irradiated with neutrons, the resulting intensity of radioactivity $I(t)$ decreases with time according to the expression

$$I(t) = \beta_1 + \beta_2 e^{-\beta_3 t} + \beta_4 e^{-\beta_5 t} \tag{14}$$

Although $I(t)$ is linear in β_1, β_2 and β_4, it is nonlinear in both β_3 and β_5, and there is no way to transform this function into any two-parameter straight line. It is possible, however, to use the *least squares* technique to provide estimates for each of the five parameters, along with estimates for the uncertainty in each.

Other examples encountered in the advanced laboratory include the transient oscillatory decay that arises in the Cavendish experiment, and the determination of resonance curves or line shapes that may be encountered in other experiments.

The method we describe is used in a computer program called `fit` that we wrote a few years ago in an effort to improve a Bell Labs program called `nllsq` (nllsq: "**n**on**l**inear **l**east **sq**uares). Further discussion of the `fit` program may be found in Chapter 5.

The general idea

Suppose we have a set of N data points with coordinates (x_i, y_i), that we wish to describe using a function

$$y = f(x, \beta_1, \beta_2, \ldots, \beta_K)$$

an expression we'll often write as

$$y = f(x, \boldsymbol{\beta}) \tag{15}$$

Here x is the independent variable whose values are presumed precisely known, and the β_j are the K parameters whose values we desire. $\boldsymbol{\beta}$ can be thought of as a

[12] An understanding of the material in this final section is not necessary for work in the Intermediate Laboratory course. It will, however, be needed in the Advanced Laboratory course, where several experiments involve the fitting of data by nonlinear mathematical functions.

In general, the reader who takes the time to understand the nonlinear curve-fitting algorithm will be well-rewarded. It is nothing short of astonishing to watch a data set from the Mössbauer experiment—one with six spectral lines buried in lots of noise—be fit by a 19-parameter mathematical function, with each of the parameters homing in on its optimum value.

K-dimensional vector. (In the example of Eq. 14, x is replaced by t, y is replaced by I, and $K = 5$, so that $\boldsymbol{\beta}$ is a five-dimensional vector.)

Our goal is to adjust the values of the parameters so that y_i is well-approximated by $f(x_i, \mathbf{b})$, where $\mathbf{b}$ is our best estimate of the desired parameter vector $\boldsymbol{\beta}$.

To do this we minimize the quantity Φ—the weighted sum of the squares of the residuals—just as we did in fitting a straight line to a set of data points. Φ will be a function of the b_j:

$$\Phi(\mathbf{b}) = \sum_{i=1}^{N} w_i r_i^2 = \sum_{i=1}^{N} w_i [f(x_i, \mathbf{b}) - y_i]^2 = \sum_{i=1}^{N} w_i (f_i - y_i)^2 \tag{16}$$

Here f_i is an abbreviation for $f(x_i, \mathbf{b})$. As with our fitting of a straight line, $r_i \equiv f_i - y_i$ is the i^{th} residual, and w_i is the weight of the i^{th} point, a number that is ideally equal to, or at least proportional to $1/\sigma_i^2$, the inverse of the observed y-variance of that data point.

The best values of the b_j will be obtained when we have found the minimum value for $\Phi(\mathbf{b})$.[12] This will happen when

$$\frac{\partial \Phi}{\partial b_j} = 0 \tag{17}$$

for each b_j. If f is a linear function of the parameters b_j, the problem of minimizing Φ is straightforward. For example, in fitting a straight line to a set of data points there are two parameters, the intercept and the slope (b_1 and b_2). That is, $f(x, \mathbf{b}) = b_1 + b_2 x$. As described earlier, Φ may be minimized to find the best straight line by solving the two simultaneous equations represented by Eq. 17. For this case, Φ, plotted as a function of b_1 and b_2, will have the shape of a paraboloid, and contours of constant Φ will be ellipses. If we take ε_1 and ε_2 to be the excursions of b_1 and b_2 from their best values, Φ has this form:

$$\Phi = \Phi_{\min} + (\textstyle\sum w_i)\varepsilon_1^2 + 2(\sum w_i x_i)\varepsilon_1 \varepsilon_2 + (\sum w_i x_i^2)\varepsilon_2^2$$

In the general case, where $f(x, \mathbf{b})$ is a nonlinear function of the b_j, we can expect that near the minimum, Φ will also be at least approximately parabolic, that is, that Φ will have the approximate form

$$\Phi = \Phi_{\min} + \sum_j \sum_k A_{jk} \varepsilon_j \varepsilon_k \tag{18}$$

where the A_{jk} are constants—the elements of a K by K matrix called the *curvature* matrix:

$$A_{jk} = \frac{1}{2} \frac{\partial^2 \Phi}{\partial b_j \partial b_k} \qquad \text{(evaluated at } \Phi_{\min}\text{)} \tag{19}$$

The main problem is to find the values of the b_j that will minimize Φ. There are a number of methods for doing this; we'll describe three. They are called (a) the *Taylor*

[12] There is no guarantee that Φ will have only one minimum. Our solution may not be unique.

expansion, or *Newton* method, (b) the *gradient*, or *steepest descent* method, and (c) the *Marquardt* method, which combines the Taylor expansion and gradient methods, making use of the best features of each.

Useful references include a Bell Labs memo describing the `nllsq` program by Kornblit,[13] Marquardt's original paper,[14] and the more general references listed at the end of the chapter.

The Taylor expansion method

The Taylor expansion or Newton method is similar to the Newton method for finding the roots of a function. If we expand the function $f(x_i)$ in the vicinity of an arbitrary point $\mathbf{b}$ in parameter space, we obtain a linear approximation to f:

$$f(x_i, \mathbf{b} + \boldsymbol{\delta}) \approx f(x_i, \mathbf{b}) + \sum_{j=1}^{K} \frac{\partial f_i}{\partial b_j} \delta_j = f(x_i, \mathbf{b}) + \sum_{j=1}^{K} p_{ij} \delta_j \tag{20}$$

Here $\boldsymbol{\delta}$ is simply a small increment in the parameter vector. The derivatives $\partial f_i / \partial b_j$, which we abbreviate by p_{ij}, are evaluated at the point $\mathbf{b}$. If the chosen point $\mathbf{b}$ is sufficiently close to the desired final $\mathbf{b}$ vector, the right side of Eq. 20 will be a reasonable approximation to f near the minimum of Φ.

Using this linear approximation, we can proceed to minimize Φ using straightforward methods:

$$\frac{\partial \Phi}{\partial \delta_j} = \frac{\partial}{\partial \delta_j} \sum w_i r_i^2 = 2 \sum w_i r_i \frac{\partial r_i}{\partial \delta_j} = 2 \sum w_i r_i p_{ij} \tag{21}$$

since $\partial r_i / \partial \delta_j = \partial f_i / \partial \delta_j = \partial f_i / \partial b_j = p_{ij}$.

Hence we have, since we want $\partial \Phi / \partial \delta_j = 0$,

$$\sum_i w_i r_i p_{ij} = \sum_i w_i (f_i + \sum_k p_{ik} \delta_k - y_i) p_{ij} = 0$$

or

$$\sum_i \sum_k w_i p_{ij} p_{ik} \delta_k = - \sum_i w_i (f_i - y_i) p_{ij} \tag{22}$$

[13] A. Kornblit, *nllsq—Non Linear Least Square Fit in C*, Bell Laboratories Technical Memorandum (1979). This is not officially published, but a copy is available in the lab.

[14] D. W. Marquardt, *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*, J. Soc. Indust. Appl. Math. **11** 431 (1963).

Equation 22 represents a set of K linear equations that may be solved for the increments δ_j. The left side of this equation may be put in simple form by writing

$$A_{jk} = \sum_i w_i p_{ij} p_{ik}$$

Note that

$$A_{jk} = \frac{1}{2} \frac{\partial^2 \Phi}{\partial \delta_k \partial \delta_j}$$

as can be seen by differentiating Eq. 21 with respect to δ_k.[15] The curvature matrix A_{jk} is descriptive of the shape of Φ, particularly near $\Phi_{\min}$.

As can be seen from Eq. 16, the right side of Eq. 22 is $-(1/2)(\partial\Phi/\partial b_j)$, that is, half of the negative gradient vector component of the Φ surface, calculated at $\boldsymbol{\delta} = 0$. Denoting this quantity by $-g_j$, we see that Eq. 22 may be written

$$\sum_k A_{jk} \delta_k = -g_j \tag{23}$$

or in truly shorthand notation,

$$\mathbf{A}\boldsymbol{\delta} = -\mathbf{g}$$

Here the square symmetric curvature matrix $\mathbf{A}$ is of dimension K by K, while $\boldsymbol{\delta}$ and $\mathbf{g}$ are K-dimensional vectors. If $\mathbf{A}$ is nonsingular we can solve for $\boldsymbol{\delta}$ by inverting $\mathbf{A}$:

$$\boldsymbol{\delta} = -\mathbf{A}^{-1}\mathbf{g} \tag{24}$$

With luck, the δ_j, when added to the initial values of the parameters b_j, will produce a new set of parameters that lie closer to the point in b-space where Φ is a minimum. If this is the case, the process can be repeated (iterated) until a desired accuracy is achieved. If our starting point in parameter space is close to the point where Φ takes on its minimum value, this process will converge rapidly to the desired solution. On the other hand, if our initial guesses for the parameters are too far from the minimum of Φ, we may be led off to a never-never land, and the process will fail, leading to new points that actually increase the value of Φ. An alternative method, one that ensures that we find a new point for which Φ decreases, is the gradient method, which we describe next.

[15] We neglect the term $2\sum_i w_i r_i (\partial^2 r_i / \partial\delta_k \partial\delta_j)$. It vanishes when f_i is linear in the δ_j, as it surely will be near $\Phi_{\min}$.

The gradient method

In the gradient method, we simply use a correction vector whose components are proportional to the components of the negative gradient vector:

$$\delta_k = -\alpha_k g_k$$

where α_k is a constant of proportionality. Note that α_k will depend on which parameter is being corrected, that is, on k. How large should α_k be? If it's too large, the correction will overshoot the minimum of Φ. If it's too small, the approach to Φ_{min} is slow. Here's one way to think about what "too large" or "too small" mean: Note that δ_k has dimensions of b_k whereas g_k has dimensions of $1/b_k$. Therefore α_k has dimensions of b_k^2. The dimensions of $1/A_{kk}$ are also b_k^2, so we are led to write

$$\delta_k = -\frac{1}{\lambda A_{kk}} g_k \tag{25}$$

Here λ is a dimensionless factor, independent of k. (The reason for putting λ in the denominator will become clear shortly.)

Now A_{kk} is the curvature of the Φ surface in the k direction. If A_{kk} is small, we take a big step, whereas if it's big, we take a small step, which is what we want. The dimensionless parameter λ can be adjusted to achieve the optimum step length, that is, optimum convergence.

The gradient method will work well if we are so far from Φ_{min} that the Taylor expansion method fails. Near Φ_{min}, however, the gradient method generally converges much more slowly than the Taylor expansion method since **g**, and hence the correction vector that's proportional to it, become vanishingly small there.

The Marquardt method

The Marquardt method combines the best features of each of the above methods, emphasizing the gradient method at first if necessary, then switching to the Taylor expansion method as the minimum of Φ is approached. In what follows, we will explain in detail how Marquardt combines the two methods. As we shall see, the factor λ will play a key role.

Equation 25 may be written

$$\lambda A_{kk}\delta_k = -g_k \qquad \text{or} \qquad \lambda \sum_k A_{jk}\delta_{jk}\delta_k = -g_k \tag{26}$$

where δ_{jk} (not to be confused with δ_k) is 1 if $j = k$ and 0 otherwise.

Marquardt combines Eq. 23 with Eq. 26 by adding the two together and ignoring the factor of 2:

$$\sum_k (1 + \lambda\delta_{jk}) A_{jk}\delta_k = -g_j \tag{27}$$

Given a value of λ, Eq. 27 may be solved for a correction vector $\boldsymbol{\delta}$.

As λ is decreased to 0 from a value much larger than 1, this $\boldsymbol{\delta}$ changes smoothly from a gradient-type to a Taylor expansion-type correction vector, since for $\lambda \gg 1$, Eq. 27 reduces to Eq. 26, while for $\lambda = 0$, Eq. 27 reduces to Eq. 23. Thus we may use a large value of λ to start if we are far from $\Phi_{\min}$, then reduce λ with each iteration, expecting Φ to decrease as we approach $\Phi_{\min}$. That is, with each iteration we solve for a correction vector $\boldsymbol{\delta}$, add it to $\mathbf{b}$ to produce a new vector $\mathbf{b}$ that is closer (we hope) to the desired parameter vector, then decrease λ and repeat the process. Eventually we hope that λ will be decreased to such a small value that we are in the Taylor expansion regime, so that convergence will be rapid. We continue this iterative process until the application of a suitable convergence test shows that we have achieved the optimum set of parameters.

In the following sections we describe the finer details of the Marquardt algorithm—scaling, testing for convergence, determining confidence limits for the parameters, and determining to what extent the parameters are correlated with each other.

The finer details

(a) Scaling

In performing the computation it is useful to scale Eq. 27 so as to eliminate the dimensions. As we noted above in our discussion of the gradient method, the diagonal elements of the curvature matrix provide a natural scale for this problem. Thus we are led to define a scaled matrix element A^*_{jk}:

$$A^*_{jk} \equiv \frac{A_{jk}}{\sqrt{A_{jj}}\sqrt{A_{kk}}} \tag{28}$$

Note that the diagonal elements of the scaled curvature matrix $\mathbf{A}^*$ are each equal to 1.

Substituting Eq. 28 into Eq. 27 we obtain

$$\sum_k \sqrt{A_{jj}}\sqrt{A_{kk}}(1 + \lambda\delta_{jk})A^*_{jk}\delta_k = -g_j$$

or, dividing both sides by $\sqrt{A_{jj}}$:

$$\sum_k (1 + \lambda\delta_{jk})A^*_{jk}\delta^*_k = \sum_k (A^*_{jk} + \lambda\delta_{jk})\delta^*_k = -g^*_j \tag{29}$$

Here the $\delta^*_k = \sqrt{A_{kk}}\,\delta_k$ form the scaled correction vector, while $g^*_j = g_j/\sqrt{A_{jj}}$ is the scaled gradient vector. Equation 29, which may also be written in compact form as

$$(\mathbf{A}^* + \lambda\mathbf{I})\boldsymbol{\delta}^* = -\mathbf{g}^* \tag{30}$$

may be solved for the scaled correction vector $\boldsymbol{\delta}^*$ by inverting the matrix $\mathbf{A}^* + \lambda\mathbf{I}$. Then each component of the correction vector is *unscaled*:

$$\delta_k = \frac{\delta^*_k}{\sqrt{A_{kk}}} \tag{31}$$

δ_k is then added to the appropriate component of the parameter vector at the n^{th} iteration to obtain a new value for this component:

$$b_k^{(n+1)} = b_k^n + \delta_k \tag{32}$$

We expect this new set of parameters to be one for which the fit is improved, *i.e.*, one yielding a smaller value for Φ.

Given an initial guess for the parameter vector $\mathbf{b}$, we may summarize the steps of the algorithm so far as follows:

(1) Compute the initial $\Phi(\mathbf{b})$.

(2) Pick a modest value for λ, say $\lambda = 10^{-4}$.

(3) Solve Eq. 30 and use Eq. 31 to obtain the correction vector $\boldsymbol{\delta}$ and compute $\Phi(\mathbf{b} + \boldsymbol{\delta})$.

(4) If $\Phi(\mathbf{b} + \boldsymbol{\delta}) \geq \Phi(\mathbf{b})$, *increase* λ by a factor of 10 (or any other substantial factor) and return to step 3.

(5) If $\Phi(\mathbf{b} + \boldsymbol{\delta}) < \Phi(\mathbf{b})$, test for convergence (see below). If convergence is not yet achieved, *decrease* λ by a factor of 10, calculate a new (scaled) $\mathbf{A}$ matrix, and return to step 3.

(6) If convergence is achieved, set $\lambda = 0$ and recalculate the (scaled) $\mathbf{A}$ matrix, which will be useful in determining the uncertainties in the final parameter estimates, along with correlations among the parameters, as described below.

(b) Testing for convergence

At each iteration, it is necessary to test whether convergence has been achieved. In general it is inadvisable to iterate until the machine roundoff limit is reached, since this would lead to results containing more significant figures than are implied by ordinary data.

Furthermore, it is fairly common, as one nears the minimum of Φ, to find parameters wandering around in small steps, searching for an ill-defined minimum in a flat valley of complicated topology. This is particularly likely when there are large correlations among the parameters.

Thus it is necessary to establish appropriate criteria for stopping. The simplest (and crudest) is just to stop after some preset number of iterations. Of course then we won't know, except by monitoring how the parameters change at each iteration, whether convergence has been achieved. Nevertheless it is advisable to set some maximum number of iterations, just in case no ordinary convergence is reached. An appropriate number will depend on the particular problem at hand, but a number on the order of 20 or 30 is usually suitable.

In normal situations, we should stop when the changes in the parameters (the δ_j) are smaller than some specified value. Marquardt suggests stopping whenever

$$|\delta_j^*| < \varepsilon(\tau + |b_j^*|) \qquad \text{for all} \;\; j \tag{33}$$

where ε and τ are constants. Suitable values might be $\varepsilon = 10^{-5}$ and $\tau = 1.0$. The constant τ is there to take care of situations where a final parameter value might be close to zero. Kornblit calls this the "epsilon test". Scaled values of δ_j and b_j are used because they are dimensionless and likely to be of comparable magnitude.

Sometimes, particularly when the parameters are highly correlated so that the Φ surface in the vicinity of the minimum is quite flat, it may be found that λ will increase to values larger than would seem necessary to ensure a decreasing Φ. It is possible then that the correction vector is too large. In such situations it is helpful to monitor the angle γ between the scaled correction vector $\boldsymbol{\delta}^*$ and the scaled gradient vector $\mathbf{g}^*$. This angle can be calculated:

$$\gamma = \cos^{-1}\left(\frac{\mathbf{g}^* \cdot \boldsymbol{\delta}^*}{|\,\mathbf{g}^*||\,\boldsymbol{\delta}^*|}\right)$$

The technique is to increase λ until γ is less than some chosen value (typically 45 degrees). This can always be achieved since γ will decrease monotonically toward zero as λ increases. Then we do not increase λ further, but halve the correction vector, replacing Eq. 32 by

$$\mathbf{b}^{(n+1)} = \mathbf{b}^{(n)} + \frac{1}{2}\boldsymbol{\delta}^{(n)} \tag{34}$$

We continue to halve the correction vector until either the epsilon test is passed or Φ decreases. Kornblit calls this the "gamma-epsilon" test.

Finally there are two tests for non-convergence. The first is an attempt to deal with a singular Marquardt matrix $\mathbf{A}^* + \lambda\mathbf{I}$. If λ is too small and the parameters are too highly correlated this matrix may be judged singular by the `fit` program. If this is found, λ is automatically increased by a factor of 10 and the matrix recalculated. This will happen up to five times (resulting in a possible increase in λ by a factor of up to 10^5) before the program gives up in disgust.

The second is called the "gamma-lambda" test, which causes the program to stop if γ cannot be reduced to less than 90 degrees even with a λ of 10. The completion of this test is also an indication that the parameters are too highly correlated for a solution to be found.

It is not uncommon to find that the parameter values just grow larger without bound, as Φ decreases (usually slowly) toward an imagined minimum in outer space. If this is the case the searching will probably continue until the maximum number of iterations is reached, but with no meaningful results. The most likely cause for such behavior is that poor initial guesses have been made for the parameters.

(c) Confidence limits

The simplest and most often used method for estimating confidence limits for the parameters—the *one-parameter* confidence limits—is identical to that outlined earlier for the fitting of a straight line. Thus to estimate the variance in the j^{th} parameter we start with an expression analogous to Eq. 8:

$$\sigma_{b_j}^2 = \sum_i \left(\frac{\partial b_j}{\partial y_i}\right)^2 \sigma_i^2 \tag{35}$$

where b_j is the optimum value of the j^{th} parameter, and σ_i^2 is, as before, the y-variance at the i^{th} point, with σ_i^2 being inversely proportional to w_i: $\sigma_i^2 = C/w_i$, where C is a constant.

We need to calculate $\partial b_j/\partial y_i$ at the point where Φ is a minimum, that is, where $\mathbf{g} = 0$, or where for each m

$$g_m = \sum_k w_k(f_k - y_k)p_{km} = 0$$

To find $\partial b_j/\partial y_i$ we differentiate this equation with respect to y_i to get

$$\sum_k w_k p_{km} \sum_n \frac{\partial f_k}{\partial b_n}\frac{\partial b_n}{\partial y_i} - w_i p_{im} = 0$$

Thus

$$\sum_n \left(\sum_k w_k p_{kn} p_{km}\right) \frac{\partial b_n}{\partial y_i} = w_i p_{im}$$

or

$$\sum_n A_{nm}\frac{\partial b_n}{\partial y_i} = w_i p_{im}$$

which may be solved for $\partial b_j/\partial y_i$ by inverting (once again) the curvature matrix $\mathbf{A}$:

$$\frac{\partial b_j}{\partial y_i} = w_i \sum_m A_{jm}^{-1} p_{im}$$

where A_{jm}^{-1} is the jm^{th} element of $\mathbf{A}^{-1}$. Squaring this expression, multiplying by C/w_i, and summing over i yields

$$\sigma_{b_j}^2 = C A_{jj}^{-1} \tag{36}$$

If independent estimates s_i^2 of the y-variances σ_i^2 are available,[16] then $w_i \approx 1/s_i^2$ and $C \approx 1$, so $s_{b_j}^2 = A_{jj}^{-1}$ is a good estimate for the variance of the j^{th} parameter b_j, and

[16] For many measurement situations such estimates are not available. However if the y_i consist of *counts* drawn from a Poisson distribution, such as might be obtained using a Geiger counter, y_i itself is an estimate of σ_i^2, so that $1/y_i$ is an absolute estimate of w_i. For the least-squares theory described in this chapter, which assumes the y_i are drawn from a *normal*, or *Gaussian* distribution, y_i must be large enough so that the Poisson distribution may be assumed Gaussian. A common rule-of-thumb is to ensure that y_i is greater than or equal to 10.

$s_{b_j} = (A^{-1}_{jj})^{1/2}$ is a good estimate of the standard deviation of the j^{th} parameter.

In addition, if estimates of the y-variances are available, the obtained value of Φ_{min} becomes a sample value of χ^2 and may be used to test “goodness of fit”, as described in Chapter 4.

If estimates of the σ_i^2 are not available (a frequent situation), then we are reduced, as described earlier for the fitting of a straight line to a set of points, to assuming that the fit is ideal, so that χ^2 is assumed to equal its mean value of $N-K$, the number of degrees of freedom. (In this case, any desire to test “goodness of fit” must be abandoned, since a sample χ^2 is *not* available.) Thus we use $\Phi_{\text{min}}/(N-K)$ as an estimate for C, so that our estimate of the standard deviation of the j^{th} parameter becomes

$$s_{b_j} = \left(\frac{\Phi_{\text{min}}}{N-K} A^{-1}_{jj}\right)^{1/2} \tag{37}$$

To estimate the uncertainty in b_j, we multiply s_{b_j} by the appropriate “Student” t-factor, as explained earlier for the fitting of a straight line. Thus, if we have b_j as an estimate of the parameter whose “true” value is β_j, we expect that with a probability of the chosen confidence level,

$$b_j - t_{N-K} s_{b_j} \le \beta_j \le b_j + t_{N-K} s_{b_j} \tag{38}$$

where t_{N-K} is the appropriate “Student” t-factor for $N-K$ degrees of freedom and the chosen confidence level, such as shown on page 2 – 11. For a 68.3 per cent confidence level, this factor is approximately 1, and so is often omitted.[17]

A different kind of confidence limit is one that defines the *joint confidence region*, that is, the region in parameter space within which *all* of the parameters will lie with some probability, say 68.3 per cent. Such a region will have a shape that is approximated by a K-dimensional ellipsoid, a surface of constant $\Phi > \Phi_{\text{min}}$.

To envision why this is appropriate, imagine that the experiment yielding our data sample is repeated many, many times, so that we have a very large number of data samples. Each sample will produce a distinct parameter set, and hence a distinct point in parameter space. Thus the collection of repeated experiments will produce a cluster of points, one that we expect will be roughly ellipsoidal in shape, and we can arrange an ellipsoidal surface that will contain some specified fraction of the points. Its size will depend on the number of degrees of freedom and the desired confidence level. The interior of the ellipsoid so chosen is the *joint confidence region.* Statistics wizard G. E. P. Box has shown[18] that if the fit is assumed ideal (so that χ^2 is assumed equal to its mean value) the

[17] Our computer programs `fit` and `fitline` do take into account appropriate “Student” t-factors, however. See Chapter 5.

[18] G. E. P. Box, in the two-volume set, *The Collected Works of George E. P. Box*, edited by George C. Tiao (Wadsworth, Inc., Belmont, Calif., 1985). See especially *The Experimental Study of Physical Mechanisms* Vol. I, pp. 137–156, and *Application of Digital Computers in the Exploration of Functional Relationships*, Vol. II, pp. 381–388.

best estimate of the boundary of the joint confidence region is given by

$$\Phi = \Phi_{\min}\left[1 + \frac{K}{N-K}F_p(K, N-K)\right] \tag{39}$$

where $F_p(K, N-K)$ is the upper p per cent point of the F-distribution with K and $N-K$ degrees of freedom. p is the desired confidence level, say 68.3 per cent or 95 per cent. Thus if there are 2 parameters and 9 data points and we choose a 68.3 per cent confidence level, the boundary in parameter space of the joint confidence region is obtained by increasing Φ to $[1 + \frac{2}{7}F_{0.683}(2, 7)] \approx 1.39$ times its minimum value.

If the chosen function is linear in the parameters this boundary will have the shape of a K-dimensional ellipsoid. If the chosen function is nonlinear in the parameters the boundary will be only approximately ellipsoidal in shape; its actual form may be computed if desired.

The projection of the joint confidence region onto each of the parameter axes defines what is called the *support plane.* For an ellipsoidal region defined by Eq. 39, such a projection is given by

$$\Delta b_j = [K \cdot F_p(K, N-K)]^{1/2} s_{b_j} \tag{40}$$

and is called the *support plane error* for the j^{th} parameter.

The diagram shown in Fig. 4 on the next page illustrates our discussion. To create this figure we have simulated 200 repetitions of the experiment described at the beginning of this chapter,[19] using our best (non-independently determined) estimates for the σ_i^2, the variance in $y_i = T_i^2$ for the i^{th} data point. Such a simulation is called a *Monte Carlo* simulation. Each simulated repetition produces an intercept-slope pair (b_1, b_2), a point in the two-dimensional parameter space. Note that the cluster of points has an elliptical shape, as expected.[20] The degree to which the ellipse is skewed depends on the correlation between b_1 and b_2. If there were no correlation, the axes of the ellipse would coincide with the b_1 and b_2 axes. In our example this would happen only if $X_1 = \sum w_i x_i$ were zero, that is, if the curvature matrix $\mathbf{A}$ were diagonal.

Figure 5 (on page 3 − 22) shows a three-dimensional plot of Φ versus b_1 and b_2 in the vicinity of $\Phi_{\min}$, for the same example as that shown in Fig. 4. Regularly spaced contours of constant Φ are shown projected onto the b_1–b_2 plane; the innermost ellipse is approximately that shown in Fig. 4, giving the boundary of the joint confidence region.

It is helpful to realize that contours of constant χ^2 play a definitive role in our discussion. The 68.3 per cent one-parameter confidence limits are roughly determined

[19] Of course the methods used for fitting nonlinear functions can also be used for fitting straight lines. No generality is lost by using this simple example to illustrate confidence limits.

[20] If there were three parameters, the cluster of points would have a three-dimensional ellipsoidal shape (easily visualized in 3-space), while more parameters would result in an ellipsoidal cluster of still higher dimensions, not so easy to visualize.

by projecting the contour that results from increasing χ^2 by 1.0 above $\chi^2_{\rm min}$ onto the parameter axes, whereas the joint confidence region is roughly determined by the contour resulting from an increase of χ^2 by an amount depending on the number of parameters. For two parameters this is approximately 2.70. Further discussion of this approach is contained in the book of Numerical Recipes cited at the end of this chapter.

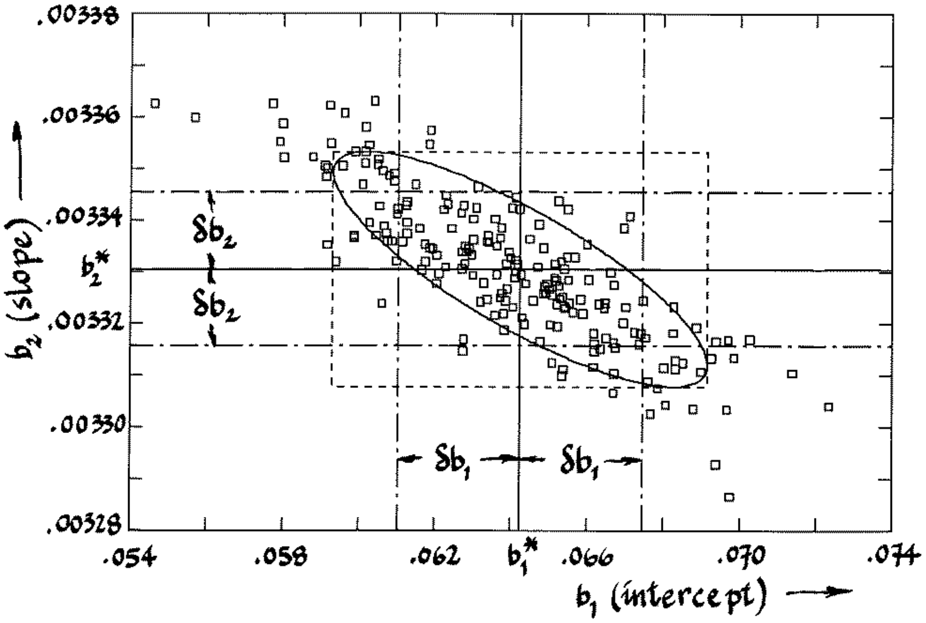

Figure 4—Confidence limits in parameter space, using the example of the straight line fit described at the start of this chapter. The best estimates of b_1 and b_2 are denoted here by b_1^* and b_2^*. The 200 points represent a Monte Carlo simulation of the experiment. δb_1 and δb_2 are estimates of the one-parameter confidence limits (see Eq. 38); approximately 68.3 per cent of the points will fall within either the horizontal band or the vertical band delineated by these limits. The joint confidence region (see Eq. 39), which contains about 68.3 per cent of the points, is delineated by the ellipse; the projection of this ellipse onto the parameter axes, indicated by the dashed rectangle, defines the *support plane.* The support plane will always extend beyond the one-parameter limits.

(d) The correlation matrix

It is useful, when fitting functions nonlinear in the parameters to data, to know the degree to which parameters are correlated with each other. Strong correlations are the rule rather than the exception, and if the correlations are too strong the method will fail because the curvature matrix **A** will be judged singular. Excess correlations frequently occur when data are attempted to be fit by a function containing too many parameters, so that the attempted fit is over-determined. In this case it is useful to know which parameters might fruitfully be abandoned. If two parameters are completely correlated then one of them may be eliminated with impunity.

Just as the *variance* of the j^{th} parameter is proportional to A^{-1}_{jj} (Eq. 36), the *covariance* $\sigma_{b_j b_k}$ relating the j^{th} and k^{th} parameters is proportional to A^{-1}_{jk}.

Thus it is not surprising that a matrix of correlation coefficients results from a scaling of the $\mathbf{A}^{-1}$ matrix. The diagonal elements of this scaled matrix will be 1.0 (each parameter will be completely correlated with itself), while an off-diagonal element such as $(A^{-1}_{jk})^*$ will indicate the degree of correlation between the j^{th} and k^{th} parameter. Off-diagonal elements close to ±1 indicate a high degree of correlation, while those close to 0 indicate hardly any correlation at all.

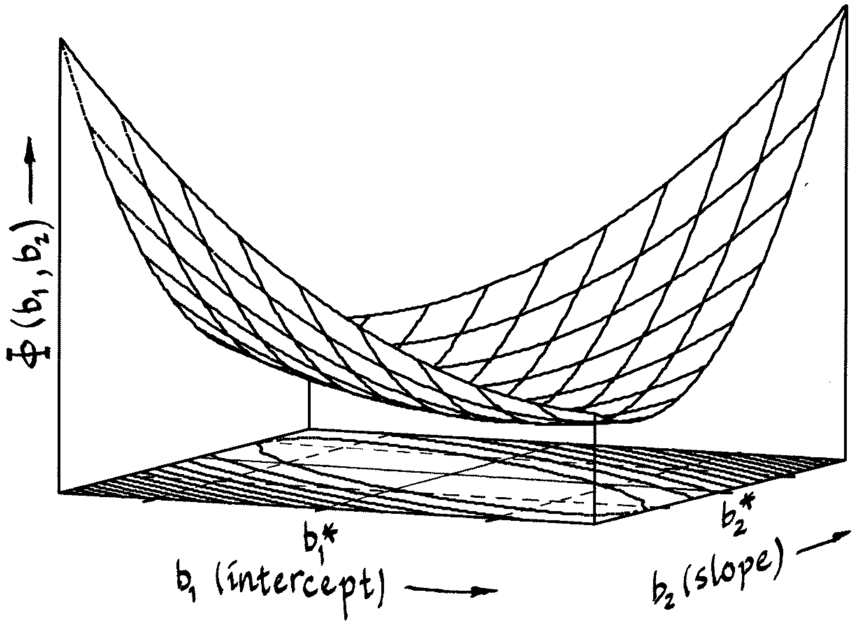

Figure 5—A plot of Φ vs b_1 and b_2 for the same example as that shown in Fig. 4. Equally spaced contours of Φ are shown projected onto the b_1–b_2 plane, with the innermost ellipse being approximately that shown in Fig. 4, giving the boundary of the joint confidence region. If you look carefully you can also see the dashed lines that show the one-parameter confidence limits. A gradient correction vector $-\mathbf{g}$ would be perpendicular to the contours of Φ, whereas a Taylor correction vector $\boldsymbol{\delta}$ would normally point toward the minimum in Φ. Note the hammock-shaped minimum, a commonly encountered situation.

References

1. Taylor, John R., *An Introduction to Error Analysis*, 2nd Ed. (University Science Books, 1997). Taylor is somewhat misleading in his discussion of the chi-square statistic (which he mistakenly calls "chi-squared"), but for most of the basic concepts this is a good place to start.

2. Bevington, Philip R., and Robinson, D. Keith, *Data Reduction and Error Analysis for the Physical Sciences*, 2nd Ed. (McGraw-Hill, 1992). Bevington's comprehensive book is commonly found on physicists' shelves. Unlike the *Numerical Recipes* book (see the following citation), it is, well, a little tedious. Think seriously about coffee if you want to delve into it.

3. Press, William H. et. al., *Numerical Recipes in C—The Art of Scientific Computing*, 2nd Ed. (Cambridge University Press, New York, 1992). Chapter 15 of this useful volume contains extensive discussion of methods for fitting data by a "model".

4. Bennett, Carl A., and Franklin, Norman L., *Statistical Analysis in Chemistry and the Chemical Industry* (Wiley, 1954). These authors are remarkably thorough in their treatment.