LEAST-SQUARES AND CHI-SQUARE FITTING FOR THE BUDDING, AND ALSO THE POST-BUDDING, AFICIONADO
February 4, 2000

Carl Heiles

In our never-ending attempt to make your life easier, we present you with this highly instructive, time-saving, and labor-saving informative document! Here we give the derivations and, also, the prescription for doing least-squares the easy way using IDL. This prescription is given as an example in §5, and the budding-aficionado *power-user* can skip the details and go directly there. The sections after the numerical example are for the more advanced post-budding aficionado.

1. LEAST-SQUARE FITTING FOR TWO PARAMETERS, AS WITH A STRAIGHT LINE.

1.1. The closed-form expressions in Taylor.

First consider the least squares fit to a straight line. Let θ_i be the i^{th} measurement of the observed quantity (e.g., the forward measurement of zenith distance angle); t_i be the time of the i^{th} measurement; N = the total number of observations, i.e. i runs from 1 to N. Remember that in the least-squares technique, quantities such as t_i are regarded to be known with high accuracy while the quantity θ_i has uncertainties in its measurement.

We expect the zenith distance θ_i to change linearly with time as follows:

$$A + Bt_i = \theta_i \ . \tag{1}$$

Given this, one looks at Taylor's equations (8.10) and (8.11) and sees how to solve for the unknowns A and B; these equations are the closed form of the solution for the pair of equations (8.8) and (8.9).

1.2. Better is the following generalized notation.

Taylor's closed-form expressions are simply longhand ways to write matrix math. We will express them as matrices below in §3. But to ease into this gently, we first will carry through an intermediate stage of notation.

First rewrite equation (1) in the more general form

$$As_i + Bt_i = \theta_i \ . \tag{2}$$

To correspond to equation (1), $s_i = 1$ for all i, but for a more general application s might be some parameter other than time that varies from one measurement to another. We have N equations like equation (2). They are known as the *equations of condition* because they are the equations that specify the theoretical model to which we are fitting the data.

There are N equations of condition and only two unknowns (A and B). This is too many equations! We have to end up with a system in which the number of equations is equal to the number of unknowns.

To accomplish this, from equation (2) we form the *normal equations*. The number of normal equations is equal to the number of unknowns, so in this case we will have two. To do this, first define the following notation:

$$[ss] = \sum_{i=0}^{N-1} s_i s_i \ , \tag{3a}$$

$$[st] = \sum_{i=0}^{N-1} s_i t_i \ , \tag{3b}$$

$$[s\theta] = \sum_{i=0}^{N-1} s_i \theta_i \ , \text{ etc.} \tag{3c}$$

Note: Above and from now on, we consider $i = 0 \rightarrow (N-1)$ instead of $i = 1 \rightarrow N$ for consistency with IDL's convention. Then the two normal equations are

$$[ss]A + [st]B = [s\theta] \ , \tag{4a}$$

$$[ts]A + [tt]B = [t\theta] \ . \tag{4b}$$

These are precisely analogous to Taylor's equations (8.8) and (8.9). You could write the solution in closed form, as in Taylor's equations (8.10) and (8.11).

2. LEAST-SQUARE FITTING FOR MANY PARAMETERS, AS WITH A CUBIC.

With this notation it's easy to generalize to more (M) unknowns: the method is obvious because in each equation of condition (like equation (2)) we simply add equivalent additional terms such as Cc_i, Dd_i, etc; and in the normal equations (equation 4) we have more products and also more normal equations.

Let's take an example with four unknowns ($M = 4$), which we will denote by A, B, C, D; this would be like fitting a cubic. With $M = 4$ we need at least five data points ($N = 5$), so there must be at least five equations of condition. The generalization of equation (2) is the N equations

$$As_i + Bt_i + Cu_i + Dv_i = \theta_i \ , \tag{5}$$

with $i = 0 \rightarrow (N-1)$. Again, the least squares fitting process assumes that the s_i, t_i, u_i, v_i are known with zero uncertainty; all of the uncertainties are in the measurements of θ_i. We then form the four normal equations; the generalization of equation (4) is:

$$\begin{bmatrix} [ss] & [st] & [su] & [sv] \\ [ts] & [tt] & [tu] & [tv] \\ [us] & [ut] & [uu] & [uv] \\ [vs] & [vt] & [vu] & [vv] \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} [s\theta] \\ [t\theta] \\ [u\theta] \\ [v\theta] \end{bmatrix} \tag{6}$$

Note that this equation is really a matrix equation, and that the $M \times M$ matrix on the left is symmetric; this means that you can actually *solve* for A, B, C, D!

3. FAR, FAR BEST AND EASIEST: MATRIX ALGEBRA.

The above equations are terribly cumbersome to solve in a computer code because they require lots of loops. However, it becomes trivial if we use matrices. Here we designate a **matrix** by **boldface** type.

We illustrate the matrix method by carrying through the above $M = 4$ example, and we assume that there are 5 independent measurements ($N = 5$). We first define the matrices

$$\mathbf{S} = \begin{bmatrix} s_0 & t_0 & u_0 & v_0 \\ s_1 & t_1 & u_1 & v_1 \\ s_2 & t_2 & u_2 & v_2 \\ s_3 & t_3 & u_3 & v_3 \\ s_4 & t_4 & u_4 & v_4 \end{bmatrix} \tag{7a}$$

$$\mathbf{A} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \tag{7b}$$

$$\mathbf{T} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \tag{7c}$$

so, in matrix form, the equations of condition (equation (5)) are simply the single matrix equation

$$\mathbf{S} \cdot \mathbf{A} = \mathbf{T} \; . \tag{8}$$

Again, there are more equations than unknowns so we can't solve this matrix equation directly. So next we form the normal equations from these matrices. In matrix form, the normal equations (equation (6)) reduce to the single equation

$$\mathbf{SS} \cdot \mathbf{A} = \mathbf{ST} \; , \tag{9}$$

where

$$\mathbf{SS} = \begin{bmatrix} [ss] & [st] & [su] & [sv] \\ [ts] & [tt] & [tu] & [tv] \\ [us] & [ut] & [uu] & [uv] \\ [vs] & [vt] & [vu] & [vv] \end{bmatrix} \tag{10a}$$

and

$$\mathbf{ST} = \begin{bmatrix} [s\theta] \\ [t\theta] \\ [u\theta] \\ [v\theta] \end{bmatrix} \tag{10b}$$

Now the number of equations is equal to the number of unknowns, so the solution of the matrix equation is easy (in IDL, that is).

In IDL, we generate the matrix **SS** by

$$\mathbf{SS} = \mathbf{transpose(S)\#\#S} \tag{11a}$$

and the matrix **ST** by

$$\mathbf{ST} = \mathbf{transpose(S)\#\#transpose(T)}\ . \tag{11b}$$

In IDL, ## is the matrix multiplication operator.[1]

To solve the matrix equation (9) for **A**, first rewrite it by multiplying both sides by the inverse of **SS** (that is, by $\mathbf{SS^{-1}}$); this gives

$$\mathbf{A} = \mathbf{SS^{-1}} \cdot \mathbf{ST}\ . \tag{12}$$

To compute $\mathbf{SS^{-1}}$ in IDL we use IDL's invert function and we multiply matrices by typing ##. Thus, in IDL, first calculate the inverse matrix $\mathbf{SS^{-1}}$, which we denote by **SSI**:

$$\mathbf{SSI} = \mathbf{invert(SS)}\ , \tag{13}$$

and then carry out the multiplication in equation 12:

$$\mathbf{A} = \mathbf{SSI}\ \#\#\ \mathbf{ST}\ . \tag{14}$$

4. UNCERTAINTIES IN THE DERIVED COEFFICIENTS.

How about the uncertainties in the derived quantities contained in the matrix **A**?

The first thing to do is derive the variance σ^2 (the square of standard deviation, or mean error, or dispersion, etc) of the individual data points using the generalization of Taylor's equation (8.14). The generalization is, simply, to replace the $N-2$ in the denominator by $N-M$. For

[1] In equation 11b we could have used **T** instead of **transpose T**; however, in the unusual case in which you are solving for a single unknown, **transpose T** works and **T** doesn't.

example, with the fit to a linear line there are 2 parameters, Taylor's A and B, so $M = 2$ and the denominator is $N - 2$. So the generalization is

$$\sigma^2 = \frac{1}{N-M} \sum_{i=0}^{N-1} (\theta_i - \bar{\theta}_i)^2 \ , \tag{15}$$

where $\bar{\theta}_i$ are the values for θ_i *predicted by the derived quantities.* Note the difference: θ_i are the *observed* values, while $\bar{\theta}_i$ are the values *predicted by the least squares fit.* The predicted values are those that are computed from the derived coefficients $A, B, C \ldots$ The N quantities $(\theta_i - \bar{\theta}_i)$ are called the *residuals* or *deviations* from the fit (Taylor, page 98).

It's easy to calculate the $\bar{\theta}_i$ with matrices. First define the matrix $\bar{\mathbf{T}}$ that contains these values:

$$\bar{\mathbf{T}} = \begin{bmatrix} \bar{\theta}_0 \\ \bar{\theta}_1 \\ \bar{\theta}_2 \\ \bar{\theta}_3 \\ \bar{\theta}_4 \end{bmatrix} \tag{16}$$

Calculating $\bar{\mathbf{T}}$ is simple:

$$\bar{\mathbf{T}} = \mathbf{S} \cdot \mathbf{A} \ . \tag{17}$$

Note that $\mathbf{S}$ is already defined (equation 7a) and $\mathbf{A}$ was solved for in equation (12) and (14).

Let's do all this in IDL. To do equation (17) in IDL (in IDL we designate $\bar{\mathbf{T}}$ by $\mathbf{BT}$):

$$\mathbf{BT} = \mathbf{S} \ \#\# \ \mathbf{A} \ . \tag{18}$$

To get σ^2 (equation (15)) in IDL:

$$sigsq = \mathbf{total}((\mathbf{T} - \mathbf{BT}) \wedge 2)/(N - M) \ . \tag{19}$$

$(\mathbf{T} - \mathbf{BT})$ is the array of residuals; it is usually a good idea to plot all three quantities $\mathbf{T}$, $\mathbf{BT}$, and $(\mathbf{T} - \mathbf{BT})$ to make sure your fit looks reasonable and to check for bad data points.

OK, now the error of the derived coefficients is given by the analogy to Taylor's equations (8.15) and (8.16). In IDL, let's designate the array that contains the squares of errors of the N

derived coefficients by **sigdcsq** (think of "**sig**ma of **d**erived **c**oefficients **sq**uared"); you get this from the diagonal elements of **SSI**, whose indices are given in IDL by[2]

$$\mathbf{sigdcsq} = sigsq * \mathbf{SSI}[(\mathbf{M}+\mathbf{1}) * \mathbf{indgen}(\mathbf{M})] \; . \tag{20}$$

Or, to put it simply in words: to get the square of the "error of coefficient j" in the matrix $\mathbf{A}$, multiply σ^2 (that's the single number $sigsq$ in IDL) by the j^{th} diagonal element of the inverse matrix $\mathbf{SS^{-1}}$. And it should be obvious that the standard deviation of the coefficients is just $sqrt(\mathbf{sigdcsq})$.

Although the above equation is correct, there is more to the story because of covariance.

4.1. The covariance matrix and its normalized counterpart

Finally, you can calculate the *normalized covariance matrix*[3] **ncov** using the following two IDL statements:

$$\mathbf{dc} = \mathbf{SSI}[(\mathbf{M}+\mathbf{1}) * \mathbf{indgen}(\mathbf{M})] \tag{21a}$$

$$\mathbf{ncov} = \mathbf{SSI}/sqrt(\mathbf{dc\#dc}) \tag{21b}$$

In the above, **dc#dc** is an $M \times M$ matrix consisting of products of the diagonals of **SSI**, so dividing **SSI** by $sqrt(\mathbf{dc\#dc})$ generates a "normalized" version of **SSI**. So you might guess that if **ncov** is a *normalized* covariance matrix, it's non-normalized parent is **SSI**—and you'd be right.

In **ncov**, the diagonal elements are all unity and the off-diagonal elements reflect the interdependence of the derived coefficients (called the "covariance") on each other; the off-diagonal elements can range from $-1 \rightarrow 1$. Values significantly different from zero show a high covariance. The normalized covariance matrix is useful because you can tell at a glance how bad things are.

Here's what the covariance means. Suppose you are least-squares fitting for two derived coefficients (A_0 and A_1). When you do a least-squares fit to a set of data, you are fitting one set of data out of a possible infinity of possible sets that you'd get by repeating the experiment, and your particular set of data happens to produce specific values of A_0 and A_1, which differ from the *true* values by amounts δA_0 and δA_1. If their covariance is zero, then in the infinity of data sets

---

[2]Specifically, for any M-sized square array **Q** the vector of its diagonal elements is given by **Q[(M+1)*indgen(M)]**.

[3]It is a pleasure to thank Doug Finkbiener for introducing me to this concept.

you'd find that δA_0 is uncorrelated with δA_1. But if it is nonzero, these two quantities would be correlated.

A high covariance is "bad" because the derived variables depend on each other. If you're evolving from being a *budding* aficionado to a real one, then you should see section 6.3 below. Often a high covariance results from a poor choice of coefficients that you are fitting or even a bad choice of the zero point of the independent variable (Sound like gobbledygook? See the numerical example in §5.). And, as in that example, you can often (not always!) eliminate the bad covariance by reformulating the problem—you don't even need to take more data!

5. A NUMERICAL EXAMPLE AND ITS SOLUTION IN IDL.

(1) First, generate the numerical example. Suppose that we make four measurements of the angle θ and we want to fit to a parabolic function in time. In the notation of equation 5, s would be unity, t the time, and u the time squared, so the number of unknowns is three ($M = 3$). Because there are four independent measurements ($N = 4$) the subscripts run from $i = 0 \rightarrow 3$. Suppose that the four values of time are 5, 7, 9, 11.

First we create the matrix **S** in IDL

$$\mathbf{S} = \mathbf{fltarr(M, N)} = \mathbf{fltarr(3, 4)} \tag{22}$$

and then we populate it with numbers. In your own work, you would normally do this by reading a data file and transferring the numbers to the matrix using IDL commands; to work through this example, you might manually type them in. After populating the matrix, in direct correspondence to equation 7a we have $s_i = 1$, $t_i = time_i$, $u_i = time_i^2$:

$$\mathbf{S} = \begin{bmatrix} 1 & 5 & 25 \\ 1 & 7 & 49 \\ 1 & 9 & 81 \\ 1 & 11 & 121 \end{bmatrix} \tag{23a}$$

and suppose that four measured values of θ are (equation 7c)

$$\mathbf{T} = \begin{bmatrix} 142 \\ 168 \\ 211 \\ 251 \end{bmatrix} . \tag{24a}$$

(2) Now solve it in IDL. In IDL we calculate the normal equation matrices (equations 10a, 11a):

$$\mathbf{SS} = \mathbf{transpose(S)\#\#S} \;, \tag{25a}$$

and (equations 10b, 11b)

$$\mathbf{ST} = \mathbf{transpose(S)\#\#transpose(T)} \;. \tag{25b}$$

In IDL we take the inverse of **SS** (equation 13):

$$\mathbf{SSI} = \mathbf{invert(SS)} \;. \tag{26}$$

The least-square fitted quantities are in the matrix **A** (equations 12, 14), which we obtain in IDL with

$$\mathbf{A} = \mathbf{SSI}\ \#\#\ \mathbf{ST} \;. \tag{27}$$

To get the uncertainties, follow the prescription in §4. More specifically, the IDL statements of equations 18, 19, and 20 provide the matrix of squares of the uncertainties, **sigdcsq**.

For this numerical example, the solution (the array of derived coefficients) is

$$\mathbf{A} = \left[\begin{array}{c} 96.6250 \\ 4.49805 \\ 0.875122 \end{array} \right] \tag{28a}$$

and the errors in the derived coefficients [the square root of the σ^2's of the derived coefficients, i.e. $(\sigma_j^2)^{1/2}$ or, in IDL, $sqrt$(**sigdcsq**) in equation 20] are:

$$\sigma_{\mathbf{A}} = \left[\begin{array}{c} 34.0118 \\ 8.99995 \\ 0.559014 \end{array} \right] \;. \tag{28b}$$

The normalized covariance matrix (equation 21) is

$$\mathbf{ncov} = \begin{bmatrix} 1 & -.989848 & .969717 \\ -.989848 & 1 & -.993808 \\ .969717 & -.993808 & 1 \end{bmatrix} . \tag{28c}$$

These results look *horrible*! The uncertainties are large fractions of the derived coefficients, and the normalized covariance matrix has large off-diagonal elements.

In this seemingly innocuous example we have an excellent case of a poor choice of zero point for the independent variable (the time). Retry this example making the zero point of the time equal to the mean of all the times, that is set ($time_i = time_i - 8$), and try it again; note the amazingly better results! Different *answers* for $\mathbf{A}$, too! In doing this, you are taking a partial step towards fitting coefficients of the orthonormal set of Chebyshev polynomials—it's always a good idea to fit orthogonal functions if you can express the problem in those terms.

You might also want to try out another example in Taylor's §8.5.

6. CHI-SQUARE FITTING, AND WEIGHTED FITTING

Chi-square fitting is very much like our least-squares fitting above, with one additional wrinkle. Chi-square (χ^2) is equal to the ratio of the variance of the *fit* (σ^2) to the intrinsic variances of the datapoints (σ_{data}^2). So it should be obvious that In Chi-square fitting, you must know the measurement uncertainties of the individual data points beforehand. If you want to give the various datapoints different weights, then that is just like Chi-square fitting except that you can adopt an arbitrary scale factor for the uncertainties (section 6.2).

6.1. The Chi-square when all datapoints have identical dispersions

Suppose, first, that all datapoints have the same intrinsic measurement uncertainty σ_{data}.[4] This is a function *only* of the *instrument* and the *data-taking procedure*: if you know the properties of the instrument, and if you understand how it works, and if there are no systematic errors, then you should be able to specify σ_{data}. Specifically, σ_{data} is most certainly *not* the same thing as the dispersion of the points from the least-squares fit.

This case, in which all datapoints have the same intrinsic measurement uncertainty σ_{data}, occurs frequently, for example when the datapoints are taken under identical conditions.

---

[4]Our discussion follows *Numerical Recipes* §14.4.3, *Solution by Use of the Normal Equations*, which we denote by NR.

Least-squares assumes a Gaussian distribution of datapoints around the true value, which is characterized by the dispersion σ_{data}. In this case, Chi-square fitting is just like least-squares fitting except for the following:

(1) Divide each equation of condition by σ_{data}. You are generating a new matrix $\mathbf{S_X}$, which is identical to $\mathbf{S}$ except that each row is divided by σ_{data}. This new matrix is the same as NR's *design matrix.*

(2) Divide each datapoint by σ_{data}. You are generating a new matrix $\mathbf{T_X}$, which is identical to $\mathbf{T}$ except that each row is divided by σ_{data}. This new matrix is the same as NR's vector $\mathbf{b}$.

(3) In our equation 19, obtain the *reduced* observed chi-square $\widehat{\chi_{obs}}^2$ by replacing $\mathbf{T}$ by $\mathbf{T_X}$ and $\mathbf{BT}$ by $\mathbf{BT_X} = \mathbf{S_X}\#\#\mathbf{A_X}$ (cf equation 18). That is, in IDL

$$\widehat{\chi^2_{obs}} = \mathbf{total}((\mathbf{T_X} - \mathbf{BT_X}) \wedge 2)/(N - M) \ . \tag{29}$$

The reduced chi-square $\widehat{\chi_{obs}}^2$ is equal to the ordinary chi-square χ^2_{obs} except that it is divided by the number of degrees of freedom, ordinarily denoted by ν, which is equal to $(N - M)$.

Look at the effect of all these three steps somewhat carefully. In steps **(1)** and **(2)**, you've taken our entire matrix equation 8 and divided both sides by the same quantity, σ_{data}. Obviously, this doesn't change the derived results for $\mathbf{A}$—as it shouldn't. So $\mathbf{A_X} = \mathbf{A}$. However, in equation 29 the vectors $\mathbf{T_X}$ and $\mathbf{BT_X}$ are multiplied by the factor $\frac{1}{\sigma_{data}}$, so equation 29 calculates $\widehat{\chi^2_{obs}} = \frac{\sigma^2}{\sigma^2_{data}}$.

This is exactly what we *should* get! Suppose, for example, that the least-squares fit model is perfect and the only deviations from the fitted curve result from measurement error. Then by necessity we have $\sigma = \sigma_{data}$ and $\widehat{\chi^2_{obs}} = 1$.

In the case $\widehat{\chi^2_{obs}} = 1$ the dispersion of the observed points σ is equal to the intrinsic dispersion of the datapoints σ_{data} and the mathematical model embodied in the least-squares fit is perfect. That, at least, is the *theoretical* conclusion. In practice, however, you're obtaining such a low, good value for χ^2 might mean instead that you are using *too large* a value for σ_{data}: you are ascribing more error to your datapoints than they really have, perhaps by not putting enough faith in your instrument. Specifically, if the datapoints really have intrinsic measurement dispersion σ_{data} and you specify that they have $2\sigma_{data}$ when computing $(\mathbf{S_X}, \mathbf{T_X})$, then you'll find $\widehat{\chi^2_{obs}} = 0.25$.

High values of $\widehat{\chi^2_{obs}}$ indicate that the model is not perfect and could be improved by the use of a different model, such as the addition of more parameters—or, alternatively, that you think your equipment works better than it really does and you are ascribing *less* error to your datapoints than they really have.

We don't discuss the interpretation of χ^2 here, because that is done very adequately in textbooks on statistics. In particular, we recommend Taylor's chapter on Chi-square (§12), which provides a nice introduction and discussion.

6.2. The case in which datapoints have different dispersions, or different weights: like a weighted average

In this case, each row of the equation-of-condition matrix **S** and each element of the data vector **T** get divided by their corresponding σ_{data}. The equation embodied in each row of the matrix equation 8 remains unchanged, but the different rows are weighted differently with respect to each other.

Consider two measurements with intrinsic measurement uncertainties (σ_1, σ_2); suppose $\sigma_1 < \sigma_2$. After being divided by their respective σ_{data}'s, all of the numbers in row 1 are larger than those in row 2. In all subsequent matrix operations, these larger numbers contribute more to all of the matrix-element products and sums. Thus, the measurement with smaller uncertainty has more influence on the final result, as it should.

Suppose that the above two measurements were taken under identical conditions except that measurement 1 received more integration time than measurement 2; we have $\frac{\sigma_1}{\sigma_2} = \left(\frac{t_1}{t_2}\right)^{-1/2}$, so the rows of $\mathbf{S_X}$ are weighted as $t^{1/2}$. This means that during the computation of $\mathbf{SS_X}$, the self-products of row 1 are weighted as t (and each row multiplies *only* itself; see equations 3 and 6). This means that each datapoint is weighted as t, which is exactly what you'd expect! Note that this is also exactly the same weighting scheme used in a weighted average, in which the weights are proportional to $\left(\frac{1}{\sigma_{data}}\right)^2$. We conclude that weighting scheme of the three steps in section 6.1 agrees with common sense.

Suppose you don't know the intrinsic measurement dispersion σ_{data}, but you *do* know the *relative* dispersion of the various measurements. For example, this would be the case if the datapoints were taken under identical conditions except for integration time; then $\sigma_{data} \propto t^{-1/2}$. In this case, multiply each row by its weight $w \propto \frac{1}{\sigma_{data}}$ and proceed as above.

6.3. The uncertainties of the derived coefficients

The above two subsections deal with the Chi-square of the fitted points, and provide information about how well the model fits the data. It says nothing about the uncertainties in the derived coefficients. To get this[5], you re-express equation 20 in terms of the new parameters, namely

[5]Our discussion follows *Numerical Recipes* §14.5, the first three italicized subsections, which we denote by NR.

$$\mathbf{sigdcsq} = \widehat{\chi^2_{obs}} * \mathbf{SSI_X}[(\mathbf{M}+\mathbf{1}) * \mathbf{indgen}(\mathbf{M})] \,, \tag{30}$$

where $\widehat{\chi^2_{obs}}$ is from equation 29. However, *this is most definitely not the whole story* because you are doing a multiparameter fit, and generally speaking the derived parameters have nonzero covariance.

Consider the example of the first two coefficients in our above example. In this example, the fit gives $y = A_0 + A_1 t + A_2 t^2$, where the numerical values are given in vector form by equation 28. The coefficient A_0 is the y-intercept and A_1 is the slope. They have derived values $A_0 = 96 \pm 34$ and $A_1 = 4 \pm 9$.

Remember what these uncertainties really mean: in an infinity of similar experiments, you'll obtain an infinity of values of (A_0, A_1) that are normally distributed with dispersions (34,9). Loosely speaking, this means that A_0 lies between $(96 - 34 = 62)$ and $(96 + 34 = 130)$ and A_1 lies between -4.5 and 13.5. If you are interested in knowing about A_0 *without regard to* A_1, then its uncertainty is indeed 96; ditto for A_1. In other words, equations 20 and 30 apply. Thus, if you don't care about A_1, then you can conclude $A_0 = 130$; and if you don't care about A_0, then you can conclude $A_1 = 13.5$ (both with 68.3% probability).

However, if you are interested in knowing about *both*, you must include their covariance. In our example, the large negative covariance follows logically just from looking at a graph: if you fit some points, all of which lie at positive t, then a more negative derived slope will raise the y-intercept.

Specifically, the large negative covariance means that positive departures of A_0 are associated with negative departures of A_1. So even though the *individual* values $A_0 = 130$ and $A_1 = 13.5$ are acceptable, you *cannot* conclude that the *pair* of values $(A_0, A_1) = (130, 13.5)$ is acceptable, because this pair has large positive departures of both. In contrast, what *is* acceptable here would be something like $(A_0, A_1) = (130, -4.5)$.

We stress that the acceptable ranges of values depend on what you are interested in. This is sort of like the observer's influence in quantum mechanics. If you are interested in A_1 alone, then you can say $A_1 = 4 \pm 9$ and, in making this statement, you have to realize that, as A_1 varies over this range, A_0 can vary over (formally, at least) the range $(\infty \rightarrow -\infty)$: you just don't give a damn *what* happens to A_0 because you're not interested. But the moment you become interested and restrict its possible range, that influences the possible range for A_1, too.

There is no simple relationship between the covariance matrix elements and the acceptable ranges. For two variables, the best way to express this matters is to construct the ellipses that define the loci of constant $\Delta\chi^2$ and present them on a graph as in NR's Figure 14.5.4. For three variables, these ellipses become ellipsoids; for four, they become four-dimensional volumes, etc.

To construct these curves/surfaces/volumes/etc., follow the prescription defined by the six bullets at the end of the section NR's *Probability Distribution of Parameters in the Normal Case.* This description is a bit curt, but it is accurate. To make sure that you understand it, take a numerical example and try applying the prescription to the case of deriving the uncertainties of a single coefficient (the case $\nu = 1$); that case is the case covered by equations 20 and 30 above. In this case, you have the usual normal distribution function for that variable, in which its formal error (as defined in equations 20 and 30 above) defines a range within which the derived value will fall 68.3% of the time in the infinity of experiments—it's just like any other normal distribution.

7. YOU WANT MORE? YOU GOT IT!

In our above example, a true expert (there aren't many of these, even among professional astronomers) would often do something more involved, namely to fit an orthogonal set of polynomials (i.e., in this case the first three terms of Chebyshev polynomials) instead of just a cubic. Because this set is orthogonal, the off-diagonal elements of **ncov** would automatically equal zero.

There are two "bibles" of least-square fitting. Most important is Press *et al.*'s *Numerical Recipes*, which provides a wide-ranging treatment of computational techniques. Our treatment follows theirs quite closely, but I wrote this before knowing of their treatment so didn't use their notation. Our matrices correspond to theirs pairwise (ours, theirs) as follows: $(\mathbf{S}, \mathbf{A})$; $(\mathbf{T}, \beta)$; $(\mathbf{S^T} \cdot \mathbf{S}\ [= \mathbf{SS}], \alpha)$; $(\mathbf{SSI}\ [= \mathbf{SS}^{-1}], \mathbf{C}\ [= \alpha^{-1}])$.

An ancient (1863!—and in this case, age is beauty!) treatise by William Chauvenet, the Appendix of *A Manual of Spherical and Practical Astronomy*, treats complicated cases such as least-square fitting in the presence of known constraints (e.g. if three angles of a triangle are measured, their sum is constrained to be 180°). But it doesn't use matrices.

8. ADDENDUM: FITTING A LINE WHEN BOTH X AND Y HAVE UNCERTAINTIES

We've mentioned that one of the essential assumptions of least squares is that the independent variables are known with high precision and the errors occur only in the measured data. Suppose you're fitting two variables, t and θ, as in equation 1. This essential assumption means that t is known with high precision and all the uncertainty is in θ, and you are minimizing the squares of the residuals in the θ-direction only.

If *both* variables have uncertainties, then you have to be careful because the essential assumption is violated. Before doing anything, read (lightly) Isobe, Feigelson, Akritas, and Babu (1990, ApJ 364, 104). Then adopt one of the two following procedures:

(1) You may be able to reformulate the problem by defining two new variables, one of which has much less fractional uncertainty than the other. The most straightforward reformulation is to define $x = (\theta + t)$ and $y = (\theta - t)$. The sum, x, has smaller fractional uncertainty than y so its dispersion *might* not matter as much. In such a redefinition, you should include a scaling factor on one of the original variables to make the two original dispersions about equal.

Before proceeding with this option, re-read Isobe et al!

(2) More generally, use the prescription in Isobe et al. Lucky for me, I haven't had to delve into this method—but their formulation looks good!