on mathematical statistics.) This infatuation tended to focus interest away from the fact that, for real data, the normal distribution is often rather poorly realized, if it is realized at all. We are often taught, rather casually, that, on average, measurements will fall within $\pm\sigma$ of the true value 68 percent of the time, within $\pm 2\sigma$ 95 percent of the time, and within $\pm 3\sigma$ 99.7 percent of the time. Extending this, one would expect a measurement to be off by $\pm 20\sigma$ only one time out of 2×10^{88}. We all know that "glitches" are much more likely than *that*!

In some instances, the deviations from a normal distribution are easy to understand and quantify. For example, in measurements obtained by counting events, the measurement errors are usually distributed as a Poisson distribution, whose cumulative probability function was already discussed in §6.2. When the number of counts going into one data point is large, the Poisson distribution converges towards a Gaussian. However, the convergence is not uniform when measured in fractional accuracy. The more standard deviations out on the tail of the distribution, the larger the number of counts must be before a value close to the Gaussian is realized. The sign of the effect is always the same: The Gaussian predicts that "tail" events are much less likely than they actually (by Poisson) are. This causes such events, when they occur, to skew a least-squares fit much more than they ought.

Other times, the deviations from a normal distribution are not so easy to understand in detail. Experimental points are occasionally just *way off*. Perhaps the power flickered during a point's measurement, or someone kicked the apparatus, or someone wrote down a wrong number. Points like this are called *outliers*. They can easily turn a least-squares fit on otherwise adequate data into nonsense. Their probability of occurrence in the assumed Gaussian model is so small that the maximum likelihood estimator is willing to distort the whole curve to try to bring them, mistakenly, into line.

The subject of *robust statistics* deals with cases where the normal or Gaussian model is a bad approximation, or cases where outliers are important. We will discuss robust methods briefly in §15.7. All the sections between this one and that one assume, one way or the other, a Gaussian model for the measurement errors in the data. It it quite important that you keep the limitations of that model in mind, even as you use the very useful methods that follow from assuming it.

Finally, note that our discussion of measurement errors has been limited to *statistical* errors, the kind that will average away if we only take enough data. Measurements are also susceptible to *systematic* errors that will not go away with any amount of averaging. For example, the calibration of a metal meter stick might depend on its temperature. If we take all our measurements at the same wrong temperature, then no amount of averaging or numerical processing will correct for this unrecognized systematic error.

Chi-Square Fitting

We considered the chi-square statistic once before, in §14.3. Here it arises in a slightly different context.

If each data point (x_i, y_i) has its own, known standard deviation σ_i, then equation (15.1.3) is modified only by putting a subscript i on the symbol σ. That subscript also propagates docilely into (15.1.4), so that the maximum likelihood

estimate of the model parameters is obtained by minimizing the quantity

$$\chi^2 \equiv \sum_{i=1}^{N} \left(\frac{y_i - y(x_i; a_1 \ldots a_M)}{\sigma_i} \right)^2 \tag{15.1.5}$$

called the "chi-square."

To whatever extent the measurement errors actually *are* normally distributed, the quantity χ^2 is correspondingly a sum of N squares of normally distributed quantities, each normalized to unit variance. Once we have adjusted the $a_1 \ldots a_M$ to minimize the value of χ^2, the terms in the sum are not all statistically independent. For models that are linear in the a's, however, it turns out that the probability distribution for different values of χ^2 at its minimum can nevertheless be derived analytically, and is the *chi-square distribution for $N - M$ degrees of freedom*. We learned how to compute this probability function using the incomplete gamma function `gammq` in §6.2. In particular, equation (6.2.18) gives the probability Q that the chi-square should exceed a particular value χ^2 by chance, where $\nu = N - M$ is the *number of degrees of freedom*. The quantity Q, or its complement $P \equiv 1 - Q$, is frequently tabulated in appendices to statistics books, but we generally find it easier to use `gammq` and compute our own values: $Q = \texttt{gammq}(0.5\nu, 0.5\chi^2)$. It is quite common, and usually not too wrong, to assume that the chi-square distribution holds even for models that are not strictly linear in the a's.

This computed probability gives a quantitative measure for the goodness-of-fit of the model. If Q is a very small probability for some particular data set, then the apparent discrepancies are unlikely to be chance fluctuations. Much more probably either (i) the model is wrong — can be statistically rejected, or (ii) someone has lied to you about the size of the measurement errors σ_i — they are really larger than stated.

It is an important point that the chi-square probability Q does not directly measure the credibility of the assumption that the measurement errors are normally distributed. It assumes they are. In most, but not all, cases, however, the effect of nonnormal errors is to create an abundance of outlier points. These decrease the probability Q, so that we can add another possible, though less definitive, conclusion to the above list: (iii) the measurement errors may not be normally distributed.

Possibility (iii) is fairly common, and also fairly benign. It is for this reason that reasonable experimenters are often rather tolerant of low probabilities Q. It is not uncommon to deem acceptable on equal terms any models with, say, $Q > 0.001$. This is not as sloppy as it sounds: Truly *wrong* models will often be rejected with vastly smaller values of Q, 10^{-18}, say. However, if day-in and day-out you find yourself accepting models with $Q \sim 10^{-3}$, you really should track down the cause.

If you happen to know the actual distribution law of your measurement errors, then you might wish to *Monte Carlo simulate* some data sets drawn from a particular model, cf. §7.2–§7.3. You can then subject these synthetic data sets to your actual fitting procedure, so as to determine both the probability distribution of the χ^2 statistic, and also the accuracy with which your model parameters are reproduced by the fit. We discuss this further in §15.6. The technique is very general, but it can also be very expensive.

At the opposite extreme, it sometimes happens that the probability Q is too large, too near to 1, literally too good to be true! Nonnormal measurement errors cannot in general produce this disease, since the normal distribution is about as "compact"

as a distribution can be. Almost always, the cause of too good a chi-square fit is that the experimenter, in a "fit" of conservativism, has *overestimated* his or her measurement errors. Very rarely, too good a chi-square signals actual fraud, data that has been "fudged" to fit the model.

A rule of thumb is that a "typical" value of χ^2 for a "moderately" good fit is $\chi^2 \approx \nu$. More precise is the statement that the χ^2 statistic has a mean ν and a standard deviation $\sqrt{2\nu}$, and, asymptotically for large ν, becomes normally distributed.

In some cases the uncertainties associated with a set of measurements are not known in advance, and considerations related to χ^2 fitting are used to derive a value for σ. If we assume that all measurements have the same standard deviation, $\sigma_i = \sigma$, and that the model does fit well, then we can proceed by first assigning an arbitrary constant σ to all points, next fitting for the model parameters by minimizing χ^2, and finally recomputing

$$\sigma^2 = \sum_{i=1}^{N} [y_i - y(x_i)]^2/(N-M) \tag{15.1.6}$$

Obviously, this approach prohibits an independent assessment of goodness-of-fit, a fact occasionally missed by its adherents. When, however, the measurement error is not known, this approach at least allows *some* kind of error bar to be assigned to the points.

If we take the derivative of equation (15.1.5) with respect to the parameters a_k, we obtain equations that must hold at the chi-square minimum,

$$0 = \sum_{i=1}^{N} \left(\frac{y_i - y(x_i)}{\sigma_i^2} \right) \left(\frac{\partial y(x_i; \ldots a_k \ldots)}{\partial a_k} \right) \qquad k = 1, \ldots, M \tag{15.1.7}$$

Equation (15.1.7) is, in general, a set of M nonlinear equations for the M unknown a_k. Various of the procedures described subsequently in this chapter derive from (15.1.7) and its specializations.

CITED REFERENCES AND FURTHER READING:

Bevington, P.R. 1969, *Data Reduction and Error Analysis for the Physical Sciences* (New York: McGraw-Hill), Chapters 1–4.

von Mises, R. 1964, *Mathematical Theory of Probability and Statistics* (New York: Academic Press), §VI.C. [1]

15.2 Fitting Data to a Straight Line

A concrete example will make the considerations of the previous section more meaningful. We consider the problem of fitting a set of N data points (x_i, y_i) to a straight-line model

$$y(x) = y(x; a, b) = a + bx \tag{15.2.1}$$

This problem is often called *linear regression*, a terminology that originated, long ago, in the social sciences. We assume that the uncertainty σ_i associated with each measurement y_i is known, and that the x_i's (values of the dependent variable) are known exactly.

To measure how well the model agrees with the data, we use the chi-square merit function (15.1.5), which in this case is

$$\chi^2(a,b) = \sum_{i=1}^{N} \left(\frac{y_i - a - bx_i}{\sigma_i} \right)^2 \tag{15.2.2}$$

If the measurement errors are normally distributed, then this merit function will give maximum likelihood parameter estimations of a and b; if the errors are not normally distributed, then the estimations are not maximum likelihood, but may still be useful in a practical sense. In §15.7, we will treat the case where outlier points are so numerous as to render the χ^2 merit function useless.

Equation (15.2.2) is minimized to determine a and b. At its minimum, derivatives of $\chi^2(a,b)$ with respect to a, b vanish.

$$\begin{aligned} 0 &= \frac{\partial \chi^2}{\partial a} = -2 \sum_{i=1}^{N} \frac{y_i - a - bx_i}{\sigma_i^2} \\ 0 &= \frac{\partial \chi^2}{\partial b} = -2 \sum_{i=1}^{N} \frac{x_i (y_i - a - bx_i)}{\sigma_i^2} \end{aligned} \tag{15.2.3}$$

These conditions can be rewritten in a convenient form if we define the following sums:

$$\begin{aligned} S &\equiv \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \quad S_x \equiv \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} \quad S_y \equiv \sum_{i=1}^{N} \frac{y_i}{\sigma_i^2} \\ S_{xx} &\equiv \sum_{i=1}^{N} \frac{x_i^2}{\sigma_i^2} \quad S_{xy} \equiv \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_i^2} \end{aligned} \tag{15.2.4}$$

With these definitions (15.2.3) becomes

$$\begin{aligned} aS + bS_x &= S_y \\ aS_x + bS_{xx} &= S_{xy} \end{aligned} \tag{15.2.5}$$

The solution of these two equations in two unknowns is calculated as

$$\begin{aligned} \Delta &\equiv SS_{xx} - (S_x)^2 \\ a &= \frac{S_{xx} S_y - S_x S_{xy}}{\Delta} \\ b &= \frac{S S_{xy} - S_x S_y}{\Delta} \end{aligned} \tag{15.2.6}$$

Equation (15.2.6) gives the solution for the best-fit model parameters a and b.

We are not done, however. We must estimate the probable uncertainties in the estimates of a and b, since obviously the measurement errors in the data must introduce some uncertainty in the determination of those parameters. If the data are independent, then each contributes its own bit of uncertainty to the parameters. Consideration of propagation of errors shows that the variance σ_f^2 in the value of any function will be

$$\sigma_f^2 = \sum_{i=1}^{N} \sigma_i^2 \left(\frac{\partial f}{\partial y_i} \right)^2 \tag{15.2.7}$$

For the straight line, the derivatives of a and b with respect to y_i can be directly evaluated from the solution:

$$\begin{aligned} \frac{\partial a}{\partial y_i} &= \frac{S_{xx} - S_x x_i}{\sigma_i^2 \Delta} \\ \frac{\partial b}{\partial y_i} &= \frac{S x_i - S_x}{\sigma_i^2 \Delta} \end{aligned} \tag{15.2.8}$$

Summing over the points as in (15.2.7), we get

$$\begin{aligned} \sigma_a^2 &= S_{xx}/\Delta \\ \sigma_b^2 &= S/\Delta \end{aligned} \tag{15.2.9}$$

which are the variances in the estimates of a and b, respectively. We will see in §15.6 that an additional number is also needed to characterize properly the probable uncertainty of the parameter estimation. That number is the *covariance* of a and b, and (as we will see below) is given by

$$\mathrm{Cov}(a, b) = -S_x/\Delta \tag{15.2.10}$$

The coefficient of correlation between the uncertainty in a and the uncertainty in b, which is a number between -1 and 1, follows from (15.2.10) (compare equation 14.5.1),

$$r_{ab} = \frac{-S_x}{\sqrt{S S_{xx}}} \tag{15.2.11}$$

A positive value of r_{ab} indicates that the errors in a and b are likely to have the same sign, while a negative value indicates the errors are anticorrelated, likely to have opposite signs.

We are *still* not done. We must estimate the goodness-of-fit of the data to the model. Absent this estimate, we have not the slightest indication that the parameters a and b in the model have any meaning at all! The probability Q that a value of chi-square as *poor* as the value (15.2.2) should occur by chance is

$$Q = \texttt{gammq}\left(\frac{N-2}{2}, \frac{\chi^2}{2} \right) \tag{15.2.12}$$

Here `gammq` is our routine for the incomplete gamma function $Q(a,x)$, §6.2. If Q is larger than, say, 0.1, then the goodness-of-fit is believable. If it is larger than, say, 0.001, then the fit *may* be acceptable if the errors are nonnormal or have been moderately underestimated. If Q is less than 0.001 then the model and/or estimation procedure can rightly be called into question. In this latter case, turn to §15.7 to proceed further.

If you do not know the individual measurement errors of the points σ_i, and are proceeding (dangerously) to use equation (15.1.6) for estimating these errors, then here is the procedure for estimating the probable uncertainties of the parameters a and b: Set $\sigma_i \equiv 1$ in all equations through (15.2.6), and multiply σ_a and σ_b, as obtained from equation (15.2.9), by the additional factor $\sqrt{\chi^2/(N-2)}$, where χ^2 is computed by (15.2.2) using the fitted parameters a and b. As discussed above, this procedure is equivalent to *assuming* a good fit, so you get no independent goodness-of-fit probability Q.

In §14.5 we promised a relation between the linear correlation coefficient r (equation 14.5.1) and a goodness-of-fit measure, χ^2 (equation 15.2.2). For unweighted data (all $\sigma_i = 1$), that relation is

$$\chi^2 = (1-r^2)\text{NVar}\,(y_1 \ldots y_N) \tag{15.2.13}$$

where

$$\text{NVar}\,(y_1 \ldots y_N) \equiv \sum_{i=1}^{N} (y_i - \overline{y})^2 \tag{15.2.14}$$

For data with varying weights σ_i, the above equations remain valid if the sums in equation (14.5.1) are weighted by $1/\sigma_i^2$.

The following function, `fit`, carries out exactly the operations that we have discussed. When the weights σ are known in advance, the calculations exactly correspond to the formulas above. However, when weights σ are unavailable, the routine *assumes* equal values of σ for each point and *assumes* a good fit, as discussed in §15.1.

The formulas (15.2.6) are susceptible to roundoff error. Accordingly, we rewrite them as follows: Define

$$t_i = \frac{1}{\sigma_i}\left(x_i - \frac{S_x}{S}\right), \qquad i = 1, 2, \ldots, N \tag{15.2.15}$$

and

$$S_{tt} = \sum_{i=1}^{N} t_i^2 \tag{15.2.16}$$

Then, as you can verify by direct substitution,

$$b = \frac{1}{S_{tt}} \sum_{i=1}^{N} \frac{t_i y_i}{\sigma_i} \tag{15.2.17}$$

$$a = \frac{S_y - S_x b}{S} \tag{15.2.18}$$

$$\sigma_a^2 = \frac{1}{S}\left(1 + \frac{S_x^2}{SS_{tt}}\right) \tag{15.2.19}$$

$$\sigma_b^2 = \frac{1}{S_{tt}} \tag{15.2.20}$$

$$\mathrm{Cov}(a,b) = -\frac{S_x}{SS_{tt}} \tag{15.2.21}$$

$$r_{ab} = \frac{\mathrm{Cov}(a,b)}{\sigma_a \sigma_b} \tag{15.2.22}$$

```
#include <math.h>
#include "nrutil.h"

void fit(float x[], float y[], int ndata, float sig[], int mwt, float *a,
    float *b, float *siga, float *sigb, float *chi2, float *q)
```

Given a set of data points `x[1..ndata]`,`y[1..ndata]` with individual standard deviations `sig[1..ndata]`, fit them to a straight line $y = a + bx$ by minimizing χ^2. Returned are `a`,`b` and their respective probable uncertainties `siga` and `sigb`, the chi-square `chi2`, and the goodness-of-fit probability `q` (that the fit would have χ^2 this large or larger). If `mwt=0` on input, then the standard deviations are assumed to be unavailable: `q` is returned as `1.0` and the normalization of `chi2` is to unit standard deviation on all points.

```
{
    float gammq(float a, float x);
    int i;
    float wt,t,sxoss,sx=0.0,sy=0.0,st2=0.0,ss,sigdat;

    *b=0.0;
    if (mwt) {                                          Accumulate sums ...
        ss=0.0;
        for (i=1;i<=ndata;i++) {                        ...with weights
            wt=1.0/SQR(sig[i]);
            ss += wt;
            sx += x[i]*wt;
            sy += y[i]*wt;
        }
    } else {
        for (i=1;i<=ndata;i++) {                        ...or without weights.
            sx += x[i];
            sy += y[i];
        }
        ss=ndata;
    }
    sxoss=sx/ss;
    if (mwt) {
        for (i=1;i<=ndata;i++) {
            t=(x[i]-sxoss)/sig[i];
            st2 += t*t;
            *b += t*y[i]/sig[i];
        }
    } else {
        for (i=1;i<=ndata;i++) {
            t=x[i]-sxoss;
            st2 += t*t;
            *b += t*y[i];
        }
    }
    *b /= st2;                                          Solve for a, b, σa, and σb.
    *a=(sy-sx*(*b))/ss;
    *siga=sqrt((1.0+sx*sx/(ss*st2))/ss);
    *sigb=sqrt(1.0/st2);
```

```
    *chi2=0.0;                                          Calculate χ².
    *q=1.0;
    if (mwt == 0) {
        for (i=1;i<=ndata;i++)
            *chi2 += SQR(y[i]-(*a)-(*b)*x[i]);
        sigdat=sqrt((*chi2)/(ndata-2));                 For unweighted data evaluate typ-
        *siga *= sigdat;                                    ical sig using chi2, and ad-
        *sigb *= sigdat;                                    just the standard deviations.
    } else {
        for (i=1;i<=ndata;i++)
            *chi2 += SQR((y[i]-(*a)-(*b)*x[i])/sig[i]);
        if (ndata>2) *q=gammq(0.5*(ndata-2),0.5*(*chi2));       Equation (15.2.12).
    }
}
```

CITED REFERENCES AND FURTHER READING:

Bevington, P.R. 1969, *Data Reduction and Error Analysis for the Physical Sciences* (New York: McGraw-Hill), Chapter 6.

15.3 Straight-Line Data with Errors in Both Coordinates

If experimental data are subject to measurement error not only in the y_i's, but also in the x_i's, then the task of fitting a straight-line model

$$y(x) = a + bx \tag{15.3.1}$$

is considerably harder. It is straightforward to write down the χ^2 merit function for this case,

$$\chi^2(a,b) = \sum_{i=1}^{N} \frac{(y_i - a - bx_i)^2}{\sigma_{y\,i}^2 + b^2\sigma_{x\,i}^2} \tag{15.3.2}$$

where $\sigma_{x\,i}$ and $\sigma_{y\,i}$ are, respectively, the x and y standard deviations for the ith point. The weighted sum of variances in the denominator of equation (15.3.2) can be understood both as the variance in the direction of the smallest χ^2 between each data point and the line with slope b, and also as the variance of the linear combination $y_i - a - bx_i$ of two random variables x_i and y_i,

$$\mathrm{Var}(y_i - a - bx_i) = \mathrm{Var}(y_i) + b^2\mathrm{Var}(x_i) = \sigma_{y\,i}^2 + b^2\sigma_{x\,i}^2 \equiv 1/w_i \tag{15.3.3}$$

The sum of the square of N random variables, each normalized by its variance, is thus χ^2-distributed.

We want to minimize equation (15.3.2) with respect to a and b. Unfortunately, the occurrence of b in the denominator of equation (15.3.2) makes the resulting equation for the slope $\partial\chi^2/\partial b = 0$ nonlinear. However, the corresponding condition for the intercept, $\partial\chi^2/\partial a = 0$, is still linear and yields

$$a = \left[\sum_i w_i(y_i - bx_i)\right] \Bigg/ \sum_i w_i \tag{15.3.4}$$

where the w_i's are defined by equation (15.3.3). A reasonable strategy, now, is to use the machinery of Chapter 10 (e.g., the routine `brent`) for minimizing a general one-dimensional function to minimize with respect to b, while using equation (15.3.4) at each stage to ensure that the minimum with respect to b is also minimized with respect to a.

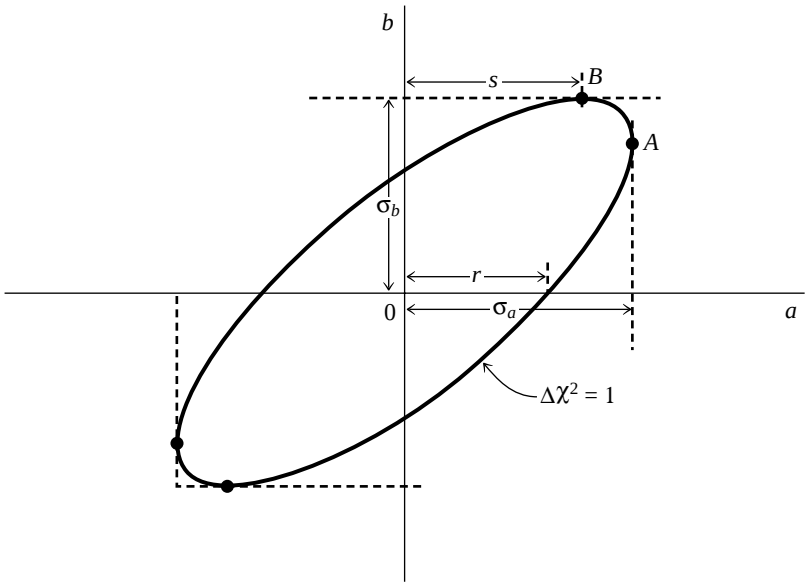

Figure 15.3.1. Standard errors for the parameters a and b. The point B can be found by varying the slope b while simultaneously minimizing the intercept a. This gives the standard error σ_b, and also the value s. The standard error σ_a can then be found by the geometric relation $\sigma_a^2 = s^2 + r^2$.

Because of the finite error bars on the x_i's, the minimum χ^2 as a function of b will be finite, though usually large, when b equals infinity (line of infinite slope). The angle $\theta \equiv \arctan b$ is thus more suitable as a parametrization of slope than b itself. The value of χ^2 will then be periodic in θ with period π (not 2π!). If any data points have very small σ_y's but moderate or large σ_x's, then it is also possible to have a maximum in χ^2 near zero slope, $\theta \approx 0$. In that case, there can sometimes be two χ^2 minima, one at positive slope and the other at negative. Only one of these is the correct global minimum. It is therefore important to have a good starting guess for b (or θ). Our strategy, implemented below, is to scale the y_i's so as to have variance equal to the x_i's, then to do a conventional (as in §15.2) linear fit with weights derived from the (scaled) sum $\sigma_{y\,i}^2 + \sigma_{x\,i}^2$. This yields a good starting guess for b if the data are even *plausibly* related to a straight-line model.

Finding the standard errors σ_a and σ_b on the parameters a and b is more complicated. We will see in §15.6 that, in appropriate circumstances, the standard errors in a and b are the respective projections onto the a and b axes of the "confidence region boundary" where χ^2 takes on a value one greater than its minimum, $\Delta\chi^2 = 1$. In the linear case of §15.2, these projections follow from the Taylor series expansion

$$\Delta\chi^2 \approx \frac{1}{2}\left[\frac{\partial^2\chi^2}{\partial a^2}(\Delta a)^2 + \frac{\partial^2\chi^2}{\partial b^2}(\Delta b)^2\right] + \frac{\partial^2\chi^2}{\partial a\partial b}\Delta a\Delta b \tag{15.3.5}$$

Because of the present nonlinearity in b, however, analytic formulas for the second derivatives are quite unwieldy; more important, the lowest-order term frequently gives a poor approximation to $\Delta\chi^2$. Our strategy is therefore to find the roots of $\Delta\chi^2 = 1$ numerically, by adjusting the value of the slope b away from the minimum. In the program below the general root finder `zbrent` is used. It may occur that there are no roots at all — for example, if all error bars are so large that all the data points are compatible with each other. It is important, therefore, to make some effort at bracketing a putative root before refining it (cf. §9.1).

Because a is minimized at each stage of varying b, successful numerical root-finding leads to a value of Δa that minimizes χ^2 for the value of Δb that gives $\Delta\chi^2 = 1$. This (see Figure 15.3.1) directly gives the tangent projection of the confidence region onto the b axis, and thus σ_b. It does not, however, give the tangent projection of the confidence region onto the a axis. In the figure, we have found the point labeled B; to find σ_a we need to find the

point A. Geometry to the rescue: To the extent that the confidence region is approximated by an ellipse, then you can prove (see figure) that $\sigma_a^2 = r^2 + s^2$. The value of s is known from having found the point B. The value of r follows from equations (15.3.2) and (15.3.3) applied at the χ^2 minimum (point O in the figure), giving

$$r^2 = 1 \Big/ \sum_i w_i \tag{15.3.6}$$

Actually, since b can go through infinity, this whole procedure makes more sense in (a, θ) space than in (a, b) space. That is in fact how the following program works. Since it is conventional, however, to return standard errors for a and b, not a and θ, we finally use the relation

$$\sigma_b = \sigma_\theta / \cos^2 \theta \tag{15.3.7}$$

We caution that if b and its standard error are both large, so that the confidence region actually includes infinite slope, then the standard error σ_b is not very meaningful. The function `chixy` is normally called only by the routine `fitexy`. However, if you want, you can yourself explore the confidence region by making repeated calls to `chixy` (whose argument is an angle θ, not a slope b), after a single initializing call to `fitexy`.

A final caution, repeated from §15.0, is that if the goodness-of-fit is not acceptable (returned probability is too small), the standard errors σ_a and σ_b are surely not believable. In dire circumstances, you might try scaling all your x and y error bars by a constant factor until the probability is acceptable (0.5, say), to get more plausible values for σ_a and σ_b.

```
#include <math.h>
#include "nrutil.h"
#define POTN 1.571000
#define BIG 1.0e30
#define PI 3.14159265
#define ACC 1.0e-3

int nn;                                          Global variables communicate with
float *xx,*yy,*sx,*sy,*ww,aa,offs;                  chixy.

void fitexy(float x[], float y[], int ndat, float sigx[], float sigy[],
    float *a, float *b, float *siga, float *sigb, float *chi2, float *q)
```

Straight-line fit to input data `x[1..ndat]` and `y[1..ndat]` with errors in both x and y, the respective standard deviations being the input quantities `sigx[1..ndat]` and `sigy[1..ndat]`. Output quantities are `a` and `b` such that $y = a + bx$ minimizes χ^2, whose value is returned as `chi2`. The χ^2 probability is returned as `q`, a small value indicating a poor fit (sometimes indicating underestimated errors). Standard errors on `a` and `b` are returned as `siga` and `sigb`. These are not meaningful if either (i) the fit is poor, or (ii) b is so large that the data are consistent with a vertical (infinite b) line. If `siga` and `sigb` are returned as `BIG`, then the data are consistent with *all* values of b.

```
{
    void avevar(float data[], unsigned long n, float *ave, float *var);
    float brent(float ax, float bx, float cx,
        float (*f)(float), float tol, float *xmin);
    float chixy(float bang);
    void fit(float x[], float y[], int ndata, float sig[], int mwt,
        float *a, float *b, float *siga, float *sigb, float *chi2, float *q);
    float gammq(float a, float x);
    void mnbrak(float *ax, float *bx, float *cx, float *fa, float *fb,
        float *fc, float (*func)(float));
    float zbrent(float (*func)(float), float x1, float x2, float tol);
    int j;
    float swap,amx,amn,varx,vary,ang[7],ch[7],scale,bmn,bmx,d1,d2,r2,
        dum1,dum2,dum3,dum4,dum5;

    xx=vector(1,ndat);
    yy=vector(1,ndat);
```

```
    sx=vector(1,ndat);
    sy=vector(1,ndat);
    ww=vector(1,ndat);
    avevar(x,ndat,&dum1,&varx);                    Find the x and y variances, and scale
    avevar(y,ndat,&dum1,&vary);                        the data into the global variables
    scale=sqrt(varx/vary);                             for communication with the func-
    nn=ndat;                                           tion chixy.
    for (j=1;j<=ndat;j++) {
        xx[j]=x[j];
        yy[j]=y[j]*scale;
        sx[j]=sigx[j];
        sy[j]=sigy[j]*scale;
        ww[j]=sqrt(SQR(sx[j])+SQR(sy[j]));         Use both x and y weights in first
    }                                                  trial fit.
    fit(xx,yy,nn,ww,1,&dum1,b,&dum2,&dum3,&dum4,&dum5);    Trial fit for b.
    offs=ang[1]=0.0;                               Construct several angles for refer-
    ang[2]=atan(*b);                                   ence points, and make b an an-
    ang[4]=0.0;                                        gle.
    ang[5]=ang[2];
    ang[6]=POTN;
    for (j=4;j<=6;j++) ch[j]=chixy(ang[j]);
    mnbrak(&ang[1],&ang[2],&ang[3],&ch[1],&ch[2],&ch[3],chixy);
    Bracket the \chi^2 minimum and then locate it with brent.
    *chi2=brent(ang[1],ang[2],ang[3],chixy,ACC,b);
    *chi2=chixy(*b);
    *a=aa;
    *q=gammq(0.5*(nn-2),*chi2*0.5);                Compute \chi^2 probability.
    for (r2=0.0,j=1;j<=nn;j++) r2 += ww[j];        Save the inverse sum of weights at
    r2=1.0/r2;                                         the minimum.
    bmx=BIG;                                       Now, find standard errors for b as
    bmn=BIG;                                           points where \Delta\chi^2 = 1.
    offs=(*chi2)+1.0;
    for (j=1;j<=6;j++) {                           Go through saved values to bracket
        if (ch[j] > offs) {                            the desired roots. Note period-
            d1=fabs(ang[j]-(*b));                      icity in slope angles.
            while (d1 >= PI) d1 -= PI;
            d2=PI-d1;
            if (ang[j] < *b) {
                swap=d1;
                d1=d2;
                d2=swap;
            }
            if (d1 < bmx) bmx=d1;
            if (d2 < bmn) bmn=d2;
        }
    }
    if (bmx < BIG) {                               Call zbrent to find the roots.
        bmx=zbrent(chixy,*b,*b+bmx,ACC)-(*b);
        amx=aa-(*a);
        bmn=zbrent(chixy,*b,*b-bmn,ACC)-(*b);
        amn=aa-(*a);
        *sigb=sqrt(0.5*(bmx*bmx+bmn*bmn))/(scale*SQR(cos(*b)));
        *siga=sqrt(0.5*(amx*amx+amn*amn)+r2)/scale;    Error in a has additional piece
    } else (*sigb)=(*siga)=BIG;                            r2.
    *a /= scale;                                   Unscale the answers.
    *b=tan(*b)/scale;
    free_vector(ww,1,ndat);
    free_vector(sy,1,ndat);
    free_vector(sx,1,ndat);
    free_vector(yy,1,ndat);
    free_vector(xx,1,ndat);
}
```

```
#include <math.h>
#include "nrutil.h"
#define BIG 1.0e30

extern int nn;
extern float *xx,*yy,*sx,*sy,*ww,aa,offs;

float chixy(float bang)
```

Captive function of `fitexy`, returns the value of $(\chi^2 - \mathtt{offs})$ for the slope `b=tan(bang)`. Scaled data and `offs` are communicated via the global variables.

```
{
    int j;
    float ans,avex=0.0,avey=0.0,sumw=0.0,b;

    b=tan(bang);
    for (j=1;j<=nn;j++) {
        ww[j] = SQR(b*sx[j])+SQR(sy[j]);
        sumw += (ww[j] = (ww[j] < 1.0/BIG ? BIG : 1.0/ww[j]));
        avex += ww[j]*xx[j];
        avey += ww[j]*yy[j];
    }
    avex /= sumw;
    avey /= sumw;
    aa=avey-b*avex;
    for (ans = -offs,j=1;j<=nn;j++)
        ans += ww[j]*SQR(yy[j]-aa-b*xx[j]);
    return ans;
}
```

Be aware that the literature on the seemingly straightforward subject of this section is generally confusing and sometimes plain wrong. Deming's [1] early treatment is sound, but its reliance on Taylor expansions gives inaccurate error estimates. References [2-4] are reliable, more recent, general treatments with critiques of earlier work. York [5] and Reed [6] usefully discuss the simple case of a straight line as treated here, but the latter paper has some errors, corrected in [7]. All this commotion has attracted the Bayesians [8-10], who have still different points of view.

CITED REFERENCES AND FURTHER READING:

Deming, W.E. 1943, *Statistical Adjustment of Data* (New York: Wiley), reprinted 1964 (New York: Dover). [1]

Jefferys, W.H. 1980, *Astronomical Journal*, vol. 85, pp. 177–181; see also vol. 95, p. 1299 (1988). [2]

Jefferys, W.H. 1981, *Astronomical Journal*, vol. 86, pp. 149–155; see also vol. 95, p. 1300 (1988). [3]

Lybanon, M. 1984, *American Journal of Physics*, vol. 52, pp. 22–26. [4]

York, D. 1966, *Canadian Journal of Physics*, vol. 44, pp. 1079–1086. [5]

Reed, B.C. 1989, *American Journal of Physics*, vol. 57, pp. 642–646; see also vol. 58, p. 189, and vol. 58, p. 1209. [6]

Reed, B.C. 1992, *American Journal of Physics*, vol. 60, pp. 59–62. [7]

Zellner, A. 1971, *An Introduction to Bayesian Inference in Econometrics* (New York: Wiley); reprinted 1987 (Malabar, FL: R. E. Krieger Pub. Co.). [8]

Gull, S.F. 1989, in *Maximum Entropy and Bayesian Methods*, J. Skilling, ed. (Boston: Kluwer). [9]

Jaynes, E.T. 1991, in *Maximum-Entropy and Bayesian Methods, Proc. 10th Int. Workshop*, W.T. Grandy, Jr., and L.H. Schick, eds. (Boston: Kluwer). [10]

Macdonald, J.R., and Thompson, W.J. 1992, *American Journal of Physics*, vol. 60, pp. 66–73.