

The resulting definitions for β_k and a_{jk} are identical to those of Equation (8.26) obtained by expanding the χ^2 function, and the χ^2 -expansion method is therefore equivalent to a second-order expansion of the fitting function.

Let us compare Equations (8.37) with the analogous Equations (7.14) and (7.15) for linear least-squares fitting. The definitions of α_{jk} in Equations (8.37) and (7.15) are equivalent in the linear approximation [See Equation (7.22)] and thus α corresponds to the curvature matrix. The definition of β_k in Equation (8.37) is equivalent, in the linear approximation, to the definition of β_k in Equations (7.14) except for the substitution of $y_i - y'(x_i)$ for y_i . We can justify this substitution by noting that the solutions of Equation (8.34) are the parameter increments δa_j , whereas those of Equation (7.14) are the parameters themselves. In essence, we are applying linear least-squares methods to fit the parameter increments to difference data Δy_i between the actual data and the starting values of the fitting $y'(x_i)$:

$$\Delta y_i = y_i - y'(x_i) \quad (8.38)$$

Thus, the expression given in Equation (8.35) for α_{jk} is a first-order approximation to the curvature matrix that is given to second order in Equation (8.37). For linear functions, the second-order term vanishes. It is convenient to use the first-order approximation for fitting nonlinear functions and thus avoid the necessity of calculating the second derivatives in Equation (8.37). We note that this procedure can be somewhat justified on the grounds that, in the vicinity of the χ^2 minimum, we should expect the factor of $y_i - y'(x_i)$ in the expression for α of Equation (8.37) to be close to 0 so that the first term in the expression will dominate.¹

Program 8.3. EXPNDFIT (Appendix E) Routine CHFIT illustrates non-linear fitting by expansion of the fitting function. The program is called repeatedly from the main program NONLINFIT, until χ^2 passes through a minimum. EXPNDFIT calls the following routines to set up and manipulate the matrices.

Program 8.6. MAKEAB8 (Appendix E) Sets up the α and β matrices. The routine uses the first-order approximation of Equation (8.35) to calculate the components α_{jk} of the curvature matrix. This is equivalent to neglecting terms in the second derivatives of the fitting function $y(x)$ in the expression for α_{jk} in Equation (8.37). The routines in this program unit use numerical derivatives and therefore differ from those with the same names in Chapter 7, which use analytic derivatives.

Program 8.7. NUMDERIV (website) Numerical derivatives. Derivatives of χ^2 (XISQ) are calculated numerically by the functions DXISQ_DA, D2XISQ_DA2, and D2XISQ_DAJK in this program unit. To avoid repetitive calculations, the values of the derivatives at each value of x and for the variation of each of the m parameters are calculated once for each trial and stored in arrays. If available, analytic expressions for the derivatives could be substituted directly for the functions to increase the speed and accuracy of the calculation.

Program B.1. MATRIX (Appendix E) Matrix multiplication and inversion.

TABLE 8.4

Two exponentials plus constant background: χ^2 expansion method

Trial	χ^2	a_1	a_2	a_3	a_4
0	406.6	10.0	900.0	80.0	27.0
1	86.2	11.1	933.8	140.4	33.8
2	66.6	10.8	861.2	128.9	33.9
3	66.1	10.4	958.2	131.2	34.0
Uncertainties		1.8	49.9	21.7	2.5

$\chi^2/\text{dof} = 1.22$; probability = 12.4%; relative time = 1.0

Note: All stages in the fit to counts from the decay of excited states of silver. The uncertainties in the parameters respond to the square roots of the diagonal terms in the error matrix.

At the conclusion of the search, the inverse ϵ of the final value of the curvature matrix α is treated as the error matrix, and the errors in the parameters are obtained from the square roots of the diagonal terms by calls to the routine SIGMATRIX in the unit FitFunc8. Table 8.4 shows values of χ^2 and the parameters a_1 through a_5 for all stages of the calculation.

8.6 THE MARQUARDT METHOD

Convergence

One disadvantage inherent in the analytical methods of expanding either the function $y(x)$ or χ^2 is that although they converge quite rapidly to the point of minimum χ^2 from points nearby, they cannot be relied on to approach the minimum with any accuracy from a point outside the region where the χ^2 hypersurface is approximately parabolic. In particular, if the curvature of the χ^2 hypersurface as in Equation (8.37) or (8.26), the analytical solution is clearly unreliable when the curvature becomes negative. Symptomatic of this problem is the need to truncate the diagonal elements α_{jj} of the matrix α so that all curvatures are treated as if they were positive.

In contrast, the gradient search of Section 8.4 is ideally suited for approaching the minimum from far away, but does not converge rapidly near the minimum. Therefore, we need an algorithm that behaves like a gradient search for the minimum and behaves more like an analytical solution as the search converges. In fact, it can be shown (see Marquardt 1963) that the path direction of a gradient and analytical searches are nearly perpendicular to each other, and the optimum direction is somewhere between these two vectors.

One advantage of combining these two methods into one algorithm is that the simpler first-order expansion of the analytical method will certainly suffice for the expansion need only be valid in the immediate neighborhood of the minimum. Thus, to calculate the curvature matrix α , we can use the approximation of

Gradient-Expansion Algorithm

A convenient algorithm (see Marquardt 1963), which combines the best features of the gradient search with the method of linearizing the fitting function, can be obtained by increasing the diagonal terms of the curvature matrix α by a factor $1 + \lambda$ that controls the interpolation of the algorithm between the two extremes. Equation (8.34) becomes

$$\beta = \delta a \alpha' \quad \text{with} \quad \alpha'_{jk} = \begin{cases} \alpha_{jk}(1 + \lambda) & \text{for } j = k \\ \alpha_{jk} & \text{for } j \neq k \end{cases} \quad (8.39)$$

If λ is very small, Equations (8.39) are similar to the solution of Equation (8.34) developed from the Taylor expansion. If λ is very large, the diagonal terms of the curvature matrix dominate and the matrix equation degenerates into m separate equations

$$\beta_j \approx \lambda \delta a_j \alpha_{jj} \quad (8.40)$$

which yield the vector increment δa in the same direction as the vector β of Equation (8.37) (or opposite to the gradient of χ^2).

The solution for the parameter increments δa_j follows from Equations (8.39) after matrix inversion

$$\delta a_j = \sum_{k=1}^m (\beta_k \epsilon'_{jk}) \quad (8.41)$$

where the β_k are given by Equation (8.37) and the matrix ϵ' is the inverse of the matrix α' with elements given by Equations (8.39).

The initial value of the constant factor λ should be chosen small enough to take advantage of the analytical solution, but large enough that χ^2 decreases. Because this algorithm approaches the gradient-search method with small steps for large λ , there should exist a value of λ such that $\chi^2(a + \delta a) < \chi^2(a)$. The recipe given by Marquardt is:

1. Compute $\chi^2(a)$.
2. Start initially with $\lambda = 0.001$.
3. Compute δa and $\chi^2(a + \delta a)$ with this choice of λ .
4. If $\chi^2(a + \delta a) > \chi^2(a)$, increase λ by a factor of 10 and repeat step 3.
5. If $\chi^2(a + \delta a) < \chi^2(a)$, decrease λ by a factor of 10, consider $a' = a + \delta a$ to be the new starting point, and return to step 3, substituting a' for a .

For each iteration it may be necessary to recompute the parameter increments δa_j from Equation (8.41), and the elements α_{jk} and β_j of the matrices, several times to optimize λ . As the solution approaches the minimum, the value of λ will decrease and the program should locate the minimum with a few iterations. A lower limit may be set for the value λ , but in practice this limit will seldom be reached.

TABLE 8.5

Two exponentials plus constant background: Marquardt method

Trial	χ^2	a_1	a_2	a_3	a_4
0	406.6	10.0	900.0	80.0	27.0
1	82.9	11.0	933.5	139.3	33.9
2	66.4	10.8	960.1	130.6	33.8
3	66.1	10.4	958.3	131.4	33.9
Uncertainties		1.8	49.9	21.7	2.5

$\chi^2/\text{dof} = 1.22$; probability = 12.4%; relative time = 1.0

Note: All stages in the fit to counts from the decay of excited states of silver. The uncertainties in the parameters respond to the square roots of the diagonal terms in the error matrix.

TABLE 8.6

Elements of the error matrix (Marquardt method)

1/k	1	2	3	4
1	3.38	-3.69	27.98	-2.34
2	-3.69	2492.26	81.89	-69.21
3	27.98	81.89	468.99	-44.22
4	-2.34	-69.21	-44.22	6.39
5	-49.24	-3.90	-615.44	53.80

Note: Error matrix from a fit to the radioactive silver data. The diagonal terms are the variances σ_k^2 and the off-diagonal terms are the covariances σ_{ki}^2 of the parameters a_k .

Program 8.4. MARQFIT (Appendix E) Routine MARQUARDT illustrating the gradient-expansion algorithm.

The procedure uses the same program units as those in Program 8.3, and is identical to that program except for the adjustment of the diagonal elements α_{jj} of the matrix α according to Equation (8.39).

At the conclusion of the search, the inverse ϵ of the final value of the curvature matrix α is treated as the error matrix, and the errors in the parameters are obtained from the square roots of the diagonal terms by calls to the function SIGMATRIX in the unit FitFunc8. Table 8.5 shows values of χ^2 and the parameters a_1 through a_5 for all stages of the calculation. Table 8.6 shows the error matrix for the fit.

8.7 COMMENTS

Although the Marquardt method is the most complex of the four fitting routines, it is also the clear winner for finding fits most directly and efficiently. It has the advantage of being reasonably insensitive to the starting values of the parameters, although in a peak-over-background example (Chapter 9), it does have difficulties when the starting parameters of the function for the peak are outside reasonable bounds.

ranges. The Marquardt method also has the advantage over the grid- and gradient-search methods of providing an estimate of the full error matrix and better calculation of the diagonal errors.

The routines of Programs 8.3 and 8.4 were tested with both numerical and analytical derivatives. Typical search paths with numerical derivatives are shown in Tables 8.4 and 8.5. For the sample problem with the assumed starting conditions, the minimum χ^2 was found in only a few steps by either method with essentially no time difference. Both methods are reasonably insensitive to starting values of parameters in which the fit is linear, but can be sensitive to starting values of the non-linear parameters. Program 8.4 had remarkable success over a broad range of starting values, whereas Program 8.3 required better definition of the starting values of the parameters and generally required many more iterations.

The uncertainties in the parameters for these fits were calculated from the diagonal terms in the error matrices and are, in general, considerably larger than the uncertainties obtained in the grid- and gradient-search methods. Because the latter errors were obtained by finding the change in each parameter to produce a change of χ^2 of 1 from the minimum values, without reoptimizing the fit, there is a strong suggestion that correlations among the parameters play an important role in fitting Figure 8.1. This point of view is supported by examination of the error matrix from the method 4 fit (Table 8.6), which shows large off-diagonal elements.

With poorly selected starting values, the searches may terminate in local minima with unacceptably high values of χ^2 and, therefore, with unacceptable final values for the parameters. Termination in the sample programs is controlled simply by considering the reduction in χ^2 from one iteration to the next and stopping at a pre-selected difference. With this method, it is essential to check the results carefully to be sure that the absolute minimum has indeed been found.

SUMMARY

Nonlinear function: One that cannot be expressed as a sum of terms with the coefficients of the terms.

Minimum of χ^2 (parabolic approximation):

$$a'_j = a_{j3} - \Delta a_j \left[\frac{\chi_3^2 - \chi_2^2}{\chi_1^2 - 2\chi_2^2 + \chi_3^2} + \frac{1}{2} \right]$$

Estimate of standard deviation from $\Delta\chi^2 = 1$:

$$\sigma_j = \Delta a_j \sqrt{2(\chi_1^2 - 2\chi_2^2 + \chi_3^2)^{-1}}$$

Grid search: Vary each parameter in turn, minimizing χ^2 with respect to each parameter independently. Many successive iterations are required to locate the minimum of χ^2 unless the parameters are independent; that is, unless the variation of χ^2 with respect to one parameter is independent of the values of the other parameters.

Gradient search: Vary all the parameters simultaneously, adjusting relative magnitudes of the variations so that the direction of propagation in parameter space is

Direction of steepest descent: Opposite the gradient $\nabla\chi^2$:

$$(\nabla\chi^2)_j = \frac{\partial\chi^2}{\partial a_j} = \frac{\chi^2(a_j + f\Delta a_j) - \chi^2(a_j)}{f\Delta a_j}$$

$$\delta a_j = \frac{-((\partial\chi^2/\partial a_j)\Delta a_j^2)}{\sqrt{\sum_{j=1}^m ((\partial\chi^2/\partial a_j)\Delta a_j)^2}}$$

Parabolic expansion of χ^2 :

$$\delta\alpha = \beta\epsilon \quad \delta a_k = \sum_{j=1}^m (\epsilon_{kj}\beta_j)$$

with

$$\beta_k \equiv -\frac{1}{2} \frac{\partial\chi^2}{\partial a_k} \quad \text{and} \quad \alpha_{jk} \equiv \frac{1}{2} \frac{\partial^2\chi^2}{\partial a_j \partial a_k}$$

Linearization of the fitting function:

$$\beta_k \equiv \sum \left\{ \frac{1}{\sigma_i^2} [y_i - y(x_i)] \frac{\partial y(x_i)}{\partial a_k} \right\} = -\frac{1}{2} \frac{\partial\chi_0^2}{\partial a_k}$$

$$\alpha_{jk} \equiv \sum \frac{1}{\sigma_i^2} \left\{ \frac{\partial y(x_i)}{\partial a_j} \frac{\partial y(x_i)}{\partial a_k} - [y_i - y'(x_i)] \frac{\partial^2 y(x_i)}{\partial a_j \partial a_k} \right\}$$

$$= \frac{1}{2} \frac{\partial^2\chi^2}{\partial a_j \partial a_k}$$

Gradient-expansion algorithm—the Marquardt method: Make λ just large enough to insure that χ^2 decreases:

$$\alpha'_{jk} = \begin{cases} \alpha_{jk}(1 + \lambda) & \text{for } j = k \\ \alpha_{jk} & \text{for } j \neq k \end{cases}$$

$$\alpha_{jk} \approx \sum \left[\frac{1}{\sigma_i^2} \frac{\partial y(x_i)}{\partial a_j} \frac{\partial y(x_i)}{\partial a_k} \right] \quad \beta_k = -\frac{1}{2} \frac{\partial\chi^2}{\partial a_k}$$

$$\delta a_j = \sum_{k=1}^m (\beta_k \epsilon'_k)$$

Uncertainty in parameter a_j : $\alpha_{aj} = e_{jj}$ corresponds to $\Delta\chi^2 = 1$.

EXERCISES

- 8.1. Use an interpolation method (see Appendix A) to find the equation of the parabola that passes through the three points (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) . Find the value of x at the minimum of the parabola and thus verify Equation (8.12).