NOAA Technical Report ERL 421-WPL 57

Nonlinear Least Squares Fitting on a Minicomputer: Method and Example

Ronald L. Schwiesow

May 1982

U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
Environmental Research Laboratories

NOAA Technical Report ERL 421-WPL 57

Nonlinear Least Squares Fitting on a Minicomputer: Method and Example

Ronald L. Schwiesow

Wave Propagation Laboratory
Boulder, Colorado

May 1982

U.S. Department of Commerce
Malcolm Baldrige, Secretary

National Oceanic and Atmospheric Administration
John V. Byrne, Administrator

Environmental Research Laboratories
Boulder, Colorado
George H. Ludwig, Director

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA Environmental Research Laboratories. Use for publicity or advertising purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

CONTENTS

Nonlinear Least Squares Fitting on a Minicomputer: Method and Example

Ronald L. Schwiesow

ABSTRACT. A method for fitting an analytic expression to data when the expression is not linear in the unknown parameters is outlined. Elements of the method include linearization of the expression, iterative improvement of the fit, and solution, based on the Gauss-Seidel approach, of a set of equations. A realization of the method, which does not require double precision, is given in FORTRAN. The method and code are applied to a specific problem of fitting a spectrum analyzer baseline with the expectation that readers can modify the program for their particular data reduction problems.

1. BACKGROUND AND PURPOSE

In many data reduction and analysis problems it is helpful or necessary to fit a parameterized analytic expression to a set of (usually noisy) data. Often the analytic expression is known from physical reasoning, and often it is not linear in the unknown parameters. For example, the calibration curve of the output of a radiometer versus target temperature depends on a number of parameters such as case temperature and amplifier gain. Rather than fit the calibration curve to a polynomial through data taken at known temperatures, it is more satisfying to fit the data to an expression that includes the physics of the device. This example is one of a class where the data analyst needs to estimate values between known values of an expression.

Another example (used for illustration in this report) is to determine an analytic representation of a spectrum analyzer baseline, i.e., a zero-signal spectrum from noisy zero-signal data. In subtracting the zero-signal spectrum from the signal spectrum, one obtains a less noisy difference spectrum if an analytic representation is used rather than the original data for the reference spectrum. In addition, the zero-signal spectrum can drift with time during an experiment, and a properly designed baseline fitting program can make adaptive changes to the parameters of the analytic form to adjust for baseline changes from an initial value. This example is representative of a class of fitting problems where a large amount of data severely overconstrains the parameterized analytic expression. When the data are noisy, knowledge of the physics of the situation allows a reduction in effective noise by the use of a derived function rather than raw data.

The intent of the study summarized here is to provide a least squares method that is simple enough to be included as part of a data reduction program run on typical minicomputers. In this setting, the nonlinear least

squares part of the operation supports the overall objectives of the program but is not an end in itself. The subroutines presented here are not general but are intended to be guides to the scientist or programer assembling a special-purpose package for a particular data analysis problem. The analysis techniques are suited to cases where a good initial guess for the fitting parameters is available and the data to be fit are not extremely noisy. The notions of good guess and moderate noise are difficult to quantify and are best checked by testing the method for reasonable results in each particular application. Fitting problems with difficult noise or first-guess conditions require more sophisticated minimum-locating and convergence procedures. I have chosen to use comparatively simple procedures that apply to straightforward data analysis applications rather than to attempt solution methods for especially difficult situations and pay the price of increased complexity for routine problems.

Generalized capabilities for nonlinear least squares fitting exist for large computers. For example, the STATLIB 3 family of subroutines for the CYBER 170/750 at the Boulder Laboratories of the Department of Commerce is based on the widely accepted NL2SOL algorithm written by Dennis, Gay, and Welsch (1980). Nonlinear least squares fitting for difficult cases is an active area of research by R. B. Schnabel (Computer Science Department, University of Colorado, Boulder, Colo.), for example. Remaining problems include convergence, scaling of parameters, and weighting functions for data points other than least squares. The question of confidence intervals for the fitted parameters is also not completely resolved. In spite of its usefulness, the NL2SOL approach is too big to work on small machines. A simpler method is required on the minicomputer equipment typically used in an interactive mode. This technical report addresses that need.

2. A SAMPLE PROBLEM

The zero-signal output spectrum of a scanning spectrum analyzer such as an HP 8553B/8552B consists of a spike at zero frequency and a frequency-independent noise level associated with the detector preamplifier feeding the analyzer input. This spectrum can be modeled as a Gaussian filter applied to a delta function (Simpson and Barr, 1975), with the result added to a level baseline. The spectrum S as a function of frequency f is then

$$S(f) = b \exp[-f^2/c^2] + d = b\{\exp[-f^2/c^2] + d/b\} , \qquad (1)$$

where b, c, and d are fitting parameters and the amplitudes are such that $d/b << 1$. The recorded output is actually the negative of the logarithm of the spectrum, and an optional zero offset is in the recorder, so that the zero-signal spectral output is

$$Y(f) = a(1) - \log\{\exp[-f^2/a(2)] + a(3)\} , \qquad (2)$$

which is a three-parameter expression, nonlinear in a(2) and a(3). This unsmoothed function is shown as the dashed line in fig. 1. The output of the

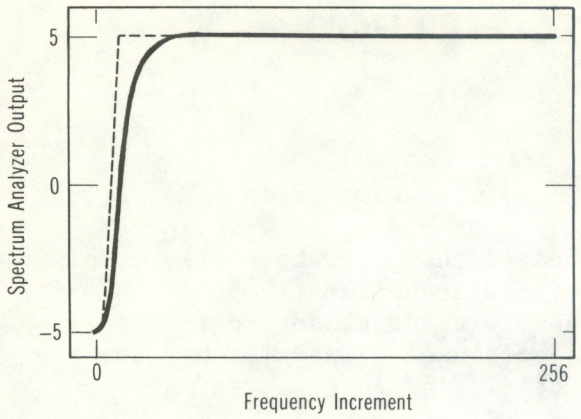

Figure 1.--Zero-signal spectral output of a spectrum analyzer. These baselines are calculated from eqs. (2) and (4) in the text with y(minimum) = -5 = a(1), y(maximum) = 5 = a(2), and a(4) = 3 (solid line, filtered) or a(4) = 0 (dashed line, unfiltered). If Δy = y(maximum) - y(minimum) = 10, then a(3) is set to 10^{10}.

spectrum analyzer is filtered by smoothing (video) filters in the analyzer and in the recorder, so that the final baseline analytic form is

$$y(f) = (1/N) \int_0^\infty Y(f-f') \exp[-f'^2/W] \, df' \ , \qquad (3)$$

where W is an adjustable filter coefficient (a fourth parameter), f' is a dummy variable of integration, and N is a normalizing coefficient given by $N = (\pi W/4)^{\frac{1}{2}}$. Uppercase notation refers to the unfiltered baseline, and lowercase to the filtered.

The baseline can be recast in digitized form by replacing frequency f by an integer i that counts the number of frequency increments from 0 frequency. Integer i also serves as the digitizing index for the digitized data to which the analytic form is to be fit. The digital baseline expression derived from the exponentially filtered spectral output for zero signal is

$$y(i) = y(i-1) + 2^{-a(4)} \, [Y(i) - y(i-1)] \ . \qquad (4)$$

Parameter a(4) depends on the exponential filter and ranges from 0 (no filter) to larger numbers. Its magnitude represents the number of previous values affecting the filter output at frequency i. (Parameter a(4) serves the same function in the digital form as W does in the analytic form.) In (2), f is replaced by i, and the units and scale for a(2) are appropriately adjusted in the digital form for Y(i) given as

$$Y(i) = a(1) - \log\{\exp[-i^2/a(2)] + a(3)\} \ . \qquad (5)$$

The filtered function is shown as the solid line in fig. 1.

3. APPLICATION OF THE FITTING METHOD

3.1 Nonlinear Least Squares

The expression for the baseline y(i) in (4) is nonlinear in a(2), a(3), and a(4). We linearize the expression in (4) by expanding in a Taylor's series using approximate or first-guess values of the parameters $\bar{a}(k)$. This development is similar to that in Sokolnikoff and Redheffer (1966, pp. 673-680) and uses y(f) and y(i) as an example case to guide those programing for their own functions. Functions y(f) and y(i) are useful examples because the nonlinearities appear in logarithmic, exponential, and integral forms. In linearized form, the baseline is

$$y(f) \cong y[f,\bar{a}(k)] + \sum_k [\partial y/\partial \bar{a}(k)]\, \Delta a(k) \ , \qquad (6)$$

where the small corrections $\Delta a(k)$ to the initial $\bar{a}(k)$ are the quantities to be solved for by the least squares procedure. The notation for the partial derivatives of y(f) means that they are evaluated with the initial values of the parameters.

To evaluate the partial derivatives in (6) for the four-parameter example case, we use Leibniz's formula on the continuous function Y(f) under the integral in (3). The result of this operation is to give an expression for the exponential average of the partial derivatives in integral form. Writing the partial derivatives as P(k,i), where k is the parameter index and i the frequency index, we obtain after considerable algebra in analogy to (4)

$$P(1,i) = 1 \qquad (7a)$$

$$P(2,i) = P(2,i-1) + 2^{-a(4)}\left\{\frac{-0.43429[x^2/a^2(2)]\exp[-x^2/a(2)]}{\exp[-x^2/a(2)] + a(3)} - P(2,i-1)\right\} \qquad (7b)$$

$$P(3,i) = P(3,i-1) + 2^{-a(4)}\left\{\frac{-0.43429}{\exp[-x^2/a(2)] + a(3)} - P(3,i-1)\right\}, \qquad (7c)$$

with x = i-1 so that the i = 1 component of the digitized baseline function now occurs at zero frequency. For a(4) it is easier to evaluate the partial derivative from (4) rather than (3). This procedure yields

$$P(4,i) = -0.69314[2^{-a(4)}][Y(i) - y(i-1)] \ . \qquad (7d)$$

The recursive forms in (4) and (7) are convenient for machine calculations. In each case, the a(k) shown in (7) are estimated $\bar{a}(k)$ in simplified notation.

To evaluate the corrections $\Delta a(k)$ to the analytic fitting parameters in a least squares sense, we form the residuals

$$v(i) = y(i) - d(i) \tag{8}$$

between the analytic expression for the baseline at the ith frequency point and the data d(i) at that point. In the usual fashion we minimize the sum of the $v^2(i)$, which can be written with the help of (6) as

$$S = \sum_i v^2(i) = \sum_i \{y[i,\bar{a}(k)] + \sum_k P(k,i)\ \Delta a(k) - d(i)\}^2 \ . \tag{9}$$

At a minimum, the partial derivatives $\partial S/\partial[\Delta a(k)] = 0$. We differentiate S term by term (each i separately) to obtain the set of k equations in the k unknowns, $\Delta a(k)$,

$$\Delta a(k)\ B(j,k) = C(j) \ , \tag{10}$$

where j is the row index and k the column index. In the example case, the coefficients are

$$B(j,k) = \sum_i P(j,i)\ P(k,i) \tag{11a}$$

$$C(j) = \sum_i P(j,i)\{d(i) - y[i,\bar{a}(k)]\} \ , \tag{11b}$$

with i ranging over the number of data points, and k and j ranging over 1 to 4.

The problem of obtaining an analytic baseline expression that is a best fit in a least squares sense to the zero-signal spectral data reduces to the problem of solving set (10) for the $\Delta a(k)$ corrections to the first-guess values of the parameter $\bar{a}(k)$. The physics of the problem enters in obtaining a useful function to represent the data in terms of a few parameters. The form or forms of the function must be selected to allow calculation of the partial derivatives of the function with respect to the parameters. In many cases, such as in the example used here, analytical derivatives can be obtained after recasting the function in an appropriate form. Computing the derivatives numerically is beyond the scope of this study and would complicate the analysis significantly. The physics of each problem is reflected in the functional form of the derivatives, as in (7) for example. The example problem also illustrates a practical digital representation of a fitting function and its derivatives.

The value of the fitting procedure for baseline or calibration data compared with simply using the data directly as a reference function is that the effect of noise is reduced in the fitted function. By introducing a priori information on the expected functional dependence of the reference data, we are able to use the data to determine a severely overconstrained set

of parameters. The fitting procedure is also preferred over simply smoothing the data along the frequency axis because simple smoothing introduces a distortion between the zero-signal spectrum analyzer output, for example, and the smoothed reference data. In a sense, the least squares fitting procedure is an efficient filter for the baseline data, reducing the number of variables from the number of data points to the number of fitting parameters.

3.2 Solution of a System of Equations

There are a number of ways to solve system (10). For the study reported on here, we have chosen a successive approximation technique called the Gauss-Seidel method. This method was chosen because it is suited to small computers, it is fairly simply coded, and it is consistent with the iterative approach adopted for the nonlinear least squares fitting procedure. Gauss-Seidel is particularly well suited to the case where the unknowns are near zero, as we shall see. This means the method is ideal for our problem because in nonlinear least squares fitting the expected values of the $\Delta a(k)$ are zero if the $\underline{a}(k)$ are best first estimates.

Alternative approaches to solving a system of equations exist. Cramer's rule is important in illustrating matrix algebra, but is of little help in practical computation (Margenau and Murphy, 1964) when the system contains more than two unknowns. Direct methods of solution based on elimination of unknowns are exact in principle but are subject to roundoff error. One direct method is Gauss elimination with pivotal condensation, which is programed for small computers in subroutine GAUSL by M. Ackley (NOAA Environmental Research Laboratories, Boulder, Colo.; personal communication, 1981). GAUSL runs in double precision and uses approximately two pages of code.

In the Gauss-Seidel method of successive approximation or iteration (Sokolnikoff and Redheffer, 1966, pp. 661-663) the given coefficient matrix is approximated by a matrix with every element above the diagonal replaced by zero (Margenau and Murphy, 1964). The resulting approximate system is more easily solved than the original. In practice, (10) is rewritten in the form

$$\Delta a(1) = [C(1) - B(1,2)\Delta a(2) - B(1,3)\Delta a(3) - \ldots - B(1,n)\Delta a(n)]/B(1,1) \qquad (12a)$$

$$\Delta a(2) = [C(2) - B(2,1)\Delta a(1) - B(2,3)\Delta a(3) - \ldots - B(2,n)\, a(n)]/B(2,2) \qquad (12b)$$

. . .
. . .
. . .

The mth equation for $\Delta a(m)^{(p)}$, the pth approximation to the solution, is evaluated with the pth approximate value for all $\Delta a(k)^{(p)}$ with $k < m$ and with the (p-1)th approximation for all $\Delta a(k)^{(p-1)}$ with $k > m$. In the Gauss-Seidel

method all $\Delta a(k)^{(0)}$ are zero. For the first time through (12), $\Delta a(1)^{(1)}$ is evaluated with a single nonzero term, $\Delta a(2)^{(1)}$ is evaluated with two nonzero terms, one containing the factor $\Delta a(1)^{(1)}$, and so on.

Equations (12) are easy to code, but the B(j,k) matrix must generally be rearranged if best results are to be obtained. The rearrangement is guided by the form of the approximation; the smaller elements are placed above the diagonal. In addition, (12) demands that the coefficient matrix be arranged by swapping rows so that there are no zeros on the diagonal.

3.3 Slowly Changing Data

In principle the nonlinear least squares fitting procedure of sec. 3.1 could be iterated, using the data set d(i), until the corrections $\Delta a(k)$ converged by some fractional change criterion. An alternative approach is to perform a fitting just once for each data set. If the data sets are repeated and there is only a small change from set to set, the corrected parameter values

$$a(k)^{(p)} = \bar{a}(k)^{(p)} + \Delta a(k)^{(p)} \tag{13a}$$

from the previous pth data set serve as the first guess

$$\bar{a}(k)^{(p+1)} = a(k)^{(p)} \tag{13b}$$

for the (p+1)th data set fitting.

The single fitting to repeated baseline data is particularly appropriate when the baseline is slowly changing, as it is in the sample problem of a spectrum analyzer in this report. Parameter a(1), in particular, changes with time as the experimental optical apparatus drifts, so it is necessary to have an adaptive routine to adjust the fitting parameters with time. By slowly changing data we mean that the parameter estimate of (13b) is a good estimate for the fitting procedure. There are two ways of obtaining baseline data sets that are almost continuous in time. One alternative is to run zero-signal data on a regular basis (e.g., every other spectrum). The other way is to identify and remove from the data spectrum those points corresponding to signal above the baseline. For our test case example we chose the latter method of obtaining repetitive baseline data sets because signal portions of the spectrum were intermittent in time and occupied only 10% or so of the total spectral scan.

Time averaging of the baseline data to reduce noise is also appropriate when the baseline is changing slowly. A useful averaging routine is to use the time scale analog of the frequency scale exponential filter in (4). If d'(i) is the averaged baseline data at frequency point i and d(i) is the present zero-signal spectrum data set, the average is

$$d'(i)_{new} = d'(i)_{old} + AV[d(i) - d'(i)_{old}] \ . \tag{14}$$

Coefficient AV is an averaging factor ranging between 1 (no averaging) and 0 (ignoring new data). Notice that there is no distortion or smoothing of the data in the frequency dimension by this averaging process.

4. SAMPLE SUBROUTINES

4.1 Baseline Subroutine

Baseline Subroutine in FORTRAN (shown on the following page) performs a nonlinear least squares fit of (4) to a set of baseline data. This subroutine was implemented on a Data General Corporation Eclipse S/250 in FORTRAN V, but it can be easily adapted to any system with minor modifications. In addition to fitting the baseline data as outlined in sec. 3.1, the subroutine conditions the successive spectrum-analyzer traces by (a) eliminating parts of the spectrum where signal occurs and (b) performing an exponential (in time) average of the zero signal or baseline parts of successive spectra.

The first part of the subroutine merely sets up an initial data average YBAR(I) from the initial baseline data Y(I) and an analytic function YFIT(I). The average and function exist from previous passes for all but the first iteration. The next section checks the data to eliminate parts of the spectrum where signal exists from the baseline fitting procedure. If a datum is more than a preset noise tolerance TOL from the analytic function, then the datum is replaced by the value of the analytic function at that point. The tolerance changes with the value of the baseline because the spectrum analyzer output is logarithmic and the output change for a given signal depends on the value of the baseline below the signal. Either the original or substituted datum is entered into the baseline data average YBAR(I) by modifying the average in accord with (14). This data initialization and selection procedure assumes that the first few spectra are zero-signal spectra, but the four-parameter fitting procedure quickly damps out the effect of a small initial signal on the baseline.

The actual least squares fitting occurs in the main part of the subroutine that starts by initializing values for the two partial derivatives that are subject to exponential averaging. The first subsection of the main part calculates and stores the values of the partial derivatives according to (7). From the P(k,i), the fitting routine calculates the B(j,k) and C(j) of (11), which are the coefficients in the simultaneous set (10) for fitting parameter corrections $\Delta a(k)$. This set of equations is solved by the solution subroutine SOLV discussed in the next section. The final part of the baseline fitting subroutine is to correct the fitting parameters a(k) with the $\Delta a(k)$ determined from the new data average and to use the corrected a(k) to calculate a new analytic baseline function, YFIT(I). The new parameters and baseline values serve as initial guesses for the next iteration (with new data) of the fitting operation.

The subroutine tracks slowly varying changes in the zero-signal baseline of the spectrum analyzer by means of an adaptive approach that updates an exponential data average with the zero-signal portions of each successive spectrum. A four-parameter analytic function is fit to each data average in a

Baseline Subroutine

```
      SUBROUTINE BASE(NPTS,Y,YFIT)
      FALL(X,A2)=EXP(-X**2/A2)
      FACT(X,A2,A3)=FALL(X,A2)+A3
      COMMON A(4),TOL,ICTR
      DIMENSION Y(512),YBAR(512),YFIT(0:512)
      DIMENSION B(4,4),C(4),P(4,0:512),DELA(4)
      CNST=-.434294
      AV=.0625
      WT=2.**(-A(4))
C     SET UP INITIAL DATA AVERAGE YBAR.  YFIT IS ANALYTICAL BASELINE
      IF(ICTR-1)10,10,20
   10 YFIT(0)=A(1)-ALOG10(A(3))
      DO 16 I=1,NPTS
      X=FLOAT(I)-1.
      YFIT(I)=YFIT(I-1)+WT*(A(1)-ALOG10(FACT(X,A(2),A(3)))-YFIT(I-1))
   16 YBAR(I)=Y(I)
   20 ICTR=ICTR+1
C     UPDATE PAST DATA FILE AND AVERAGE
      DO 40 I=1,NPTS
      IF(ABS(YFIT(I)-Y(I))-TOL)32,32,34
   32 YBAR(I)=YBAR(I)+AV*(Y(I)-YBAR(I))
      GO TO 40
   34 YBAR(I)=YBAR(I)+AV*(YFIT(I)-YBAR(I))
   40 CONTINUE
C     FIT YBAR TO 4-PARAMETER ANALYTIC FORM
      P(2,0)=0.
      P(3,0)=CNST/A(3)
      DO 60 I=1,NPTS
      X=FLOAT(I)-1.
      P(1,I)=1.
      QUANT=CNST*(X/A(2))**2*FALL(X,A(2))/FACT(X,A(2),A(3))
      P(2,I)=P(2,I-1)+WT*(QUANT-P(2,I-1))
      P(3,I)=P(3,I-1)+WT*(CNST*FACT(X,A(2),A(3))-P(3,I-1))
   60 P(4,I)=-.69314*WT*(A(1)-ALOG10(FACT(X,A(2),A(3)))-YFIT(I-1))
      DO 64 K=1,4
      DO 62 L=1,4
   62 B(K,L)=0.
   64 C(K)=0.
      DO 90 K=1,4
      DO 80 I=1,NPTS
      DO 70 L=1,4
   70 B(K,L)=B(K,L)+P(K,I)*P(L,I)
      X=FLOAT(I)-1.
   80 C(K)=C(K)+P(K,I)*(YBAR(I)-YFIT(I))
   90 CONTINUE
      CALL SOLV(B,C,DELA,4,.001)
      DO 100 K=1,4
  100 A(K)=A(K)+DELA(K)
      YFIT(0)=A(1)-ALOG(A(3))
      DO 110 I=1,NPTS
      X=FLOAT(I)-1.
  110 YFIT(I)=YFIT(I-1)+WT*(A(1)-ALOG10(FACT(X,A(2),A(3)))-YFIT(I-1))
      RETURN
      END
```

least-squares-difference sense by using a linear approximation to the dependence of the function on the parameters. The fitting process is well suited to the sample problem because the results of fitting to one baseline spectrum serve as a good initial guess for fitting the next, similar spectrum.

4.2 Solution Subroutine

Solution Subroutine in FORTRAN (shown on the following page) solves a set of N equations in N unknowns by the Gauss-Seidel method of successive approximation. Because the solution proceeds by an iterative process, it is well suited to small computers and does not require double precision for practical accuracy. The Gauss-Seidel method works best when the values of the unknowns are near zero because the method assumes zero for the initial value of all unknowns. It is possible to generalize the Gauss-Seidel method for expected final variable values far from zero by introducing first-guess values for the variables.

The first part of Solution Subroutine rearranges the coefficient matrix for system (10) by swapping rows until the biggest element above the diagonal is moved to the diagonal for each column. This is done because the Gauss-Seidel method approximates the given matrix with a matrix with zeros everywhere above the diagonal, and it is wise to make the given matrix as close to the approximate matrix as possible. The rearrangement starts with the rightmost column and moves the largest element to the diagonal. The swapping routine does not reorder the entire column because operations on other columns will deorder preceeding columns, in general. The process of moving the largest element above the diagonal to the diagonal is applied to each column in turn, moving to the left.

An important part of rearrangement is to make sure there are no zeros on the diagonal because diagonal elements appear in the denominator in (12). The probability of a zero diagonal element is highest in the first (leftmost) column and lowest in the last, so the zero check and rearrange procedure moves from left to right. After rearrangement, the sign of all the elements in one row is changed if the number of interchanges of rows is odd. Because there is no interchange of columns, the variables do not need to be relabeled.

After rearrangement, the main part of the solution proceeds by setting the initial (0th iteration) values of the variables to zero and then solving for $x(i)^{(1)}$ by (12). The value of each variable is checked in each iteration to see if the fractional change in the value exceeds a preset accuracy limit, ACU, which might be 10^{-4}, for example. If any variable exceeds this convergence limit, the flag ICONV is set and another iteration is performed.

In tests on a three-variable system the subroutine converged to four-digit accuracy in seven iterations and checked independently-derived results.

Solution Subroutine

```
      SUBROUTINE SOLV(B,C,X,N,ACU)
      DIMENSION B(N,N),C(N),X(N)
      ISWP=1
      DO 30 I1=1,N-1
      L=N+1-I1
      DO 20 I2=1,L-1
      IF(ABS(B(I2,L))-ABS(B(I2+1,L))) 20,20,10
10    DO 12 I3=1,N
      BTEMP=B(I2,I3)
      B(I2,I3)=B(I2+1,I3)
12    B(I2+1,I3)=BTEMP
      CTEMP=C(I2)
      C(I2)=C(I2+1)
      C(I2+1)=CTEMP
      ISWP=-ISWP
20    CONTINUE
30    CONTINUE
      DO 50 I1=1,N
      IF(B(I1,I1)) 50,42,50
42    DO 44 I2=I1+1,N
      IF(B(I2,I1)) 46,44,46
44    CONTINUE
46    DO 48 I3=1,N
      BTEMP=B(I2,I3)
      B(I2,I3)=B(I1,I3)
48    B(I1,I3)=BTEMP
      ISWP=ISWP+1
50    CONTINUE
      IF(ISWP) 52,58,58
52    DO 54 I=1,N
54    B(1,I)=-1.*B(1,I)
      C(1)=-1.*C(1)
58    CONTINUE
      DO 70 I=1,N
70    X(I)=0.
      IT=0
      ICONV=0
72    DO 90 I=1,N
      HOLD=X(I)
      X(I)=C(I)/B(I,I)
      IF(I-2) 84,74,74
74    DO 80 J=1,I-1
80    X(I)=X(I)-X(J)*B(I,J)/B(I,I)
      IF(N-I)90,90,84
84    DO 85 J=I+1,N
85    X(I)=X(I)-X(J)*B(I,J)/B(I,I)
      IF(ABS((X(I)-HOLD)/X(I))-ACU) 90,90,87
87    ICONV=1
90    CONTINUE
      IF(ICONV) 96,96,94
94    IT=IT+1
95    ICONV=0
      GO TO 72
96    RETURN
      END
```

5. CONCLUSIONS

A subroutine for nonlinear least squares fitting can be used on a minicomputer to fit analytical expressions to experimental data when the data are reasonably well behaved. As well as outlining the technique, this report presents an example of its use to fit a spectrum analyzer baseline to zero-signal data. In the fitting example we illustrate different techniques for obtaining analytical expressions for the partial derivatives of a nonlinear expression with respect to the fitting parameters.

It is possible to perform the fitting with approximately two pages of FORTRAN V computer code for the illustrative example used in this study. One page is the fitting subroutine itself, and the other is a successive approximation method for solving a set of simultaneous equations, which is a procedure needed for the nonlinear least squares fitting. The computer code for the Gauss-Seidel solution of a set of N equations in N unknowns is interesting in its own right because it provides a simple alternative to exact methods and is consistent with the iterative approach that is inherent in nonlinear least squares fitting.

Data analysts who have a need for least squares fitting to a function that is nonlinear in the fitting parameters should be able to adapt the subroutines in this report. Limitations on the subroutines are that an analytic expression for the partial derivatives must be available and that the data to be fitted must be reasonably well behaved. The sample case discussed shows the use of an adaptive application where the procedure for nonlinear least squares fitting is used, in conjunction with an exponential filter on the baseline data, to follow drifts in the baseline with time caused by drifts in the experimental apparatus.

6. ACKNOWLEDGMENTS

I appreciate D. A. Merritt's help in clarifying FORTRAN errors. R. S. Lawrence encouraged a test of the approaches reported here.

7. REFERENCES

Dennis, J. E., Jr., D. M. Gay, and R. E. Welsch, 1980. An adaptive nonlinear least-squares algorithm. Tech. Rep. TR-20, Alfred P. Sloan School of Management, Massachusetts Institute of Technology, Cambridge, 45 pp.

Margenau, H., and G. M. Murphy, 1964. The Mathematics of Physics and Chemistry, Vol. 2. 2nd ed., Van Nostrand, Princeton, N. J., 58-61.

Simpson, R. L., and P. W. Barr, 1975. Laser Doppler velocimeter signal processing using sampling spectrum analysis. Rev. Sci. Instrum. 46: 835-837.

Sokolnikoff, I. S., and R. M. Redheffer, 1966. Mathematics of Physics and Modern Engineering. 2nd ed., McGraw-Hill, New York, 752 pp.

U. S. Government Printing Office 1982 - 576-001/1243 Reg. 8

Environmental Research LABORATORIES

The mission of the Environmental Research Laboratories (ERL) is to conduct an integrated program of fundamental research, related technology development, and services to improve understanding and prediction of the geophysical environment comprising the oceans and inland waters, the lower and upper atmosphere, the space environment, and the Earth. The following participate in the ERL missions:

AL *Aeronomy Laboratory.* Studies the physics, dynamics, and chemistry of the stratosphere and the surrounding upper and lower atmosphere.

AOML *Atlantic Oceanographic and Meteorological Laboratories.* Study the physical, chemical, biological, and geological characteristics and processes of the ocean waters, the sea floor, and the atmosphere above the ocean, including tropical meteorology such as hurricanes and tropical weather systems.

ARL *Air Resources Laboratories.* Study the diffusion, transport, dissipation, and chemistry of atmospheric pollutants; develop methods of predicting and controlling atmospheric pollution; monitor the global physical environment to detect climatic change.

GFDL *Geophysical Fluid Dynamics Laboratory.* Studies the dynamics of geophysical fluid systems (the atmosphere, the hydrosphere, and the cryosphere) through theoretical analysis and numerical simulation using powerful, high-speed digital computers.

GLERL *Great Lakes Environmental Research Laboratory.* Studies hydrology, waves, currents, lake levels, biological and chemical processes, and lake-air interaction in the Great Lakes and their watersheds; forecasts lake ice conditions.

NSSL *National Severe Storms Laboratory.* Studies severe-storm circulation and dynamics, and develops techniques to detect and predict tornadoes, thunderstorms, and squall lines.

OWRM *Office of Weather Research and Modification.* Conducts a program of basic and applied research to advance the understanding and define the structure of mesoscale phenomena, to improve short-range weather predictions and warnings, and to identify and test hypotheses for beneficially modifying weather processes.

PMEL *Pacific Marine Environmental Laboratory.* Monitors and predicts the physical and biochemical effects of natural events and human activities on the deep-ocean and coastal marine environments of the Pacific region.

PROFS *Prototype Regional Observing and Forecasting Service.* Evaluates and integrates advanced meteorological measurement, forecasting, and communication/dissemination technologies into functional mesoscale weather forecast system designs for transfer to operational agencies such as NWS, NESS, and FAA.

RFC *Research Facilities Center.* Operates instrumented aircraft for environmental research programs; provides scientific measurement tools, logged data, and associated information for meteorological and oceanographic research programs.

SEL *Space Environment Laboratory.* Studies solar-terrestrial physics (interplanetary, magnetospheric, and ionospheric); develops techniques for forecasting solar disturbances; provides real-time monitoring and forecasting of the space environment.

WIM *Weather Modification Program Office.* Plans and coordinates ERL weather modification projects for precipitation enhancement and severe storms mitigation.

WPL *Wave Propagation Laboratory.* Develops, and applies to research and services, new methods for remote sensing of the geophysical environment.

U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
BOULDER, COLORADO 80302