NASA Technical Memorandum 4012

Interactive Application of Quadratic Expansion of Chi-Square Statistic to Nonlinear Curve Fitting

F. F. Badavi and Joel L. Everhart

OCTOBER 1987
Interactive Application
of Quadratic Expansion
of Chi-Square Statistic
to Nonlinear Curve Fitting

F. F. Badavi
PRC Kentron, Inc.
Hampton, Virginia

Joel L. Everhart
Langley Research Center
Hampton, Virginia
The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.
Abstract

This report contains a detailed theoretical description of an all-purpose, interactive curve-fitting routine that is based on P. R. Bevington's description of the quadratic expansion of the $\chi^2$ statistic. The method is implemented in the associated interactive, graphics-based computer program.

The Taylor's expansion of $\chi^2$ is first introduced, and justifications for retaining only the first term are presented. From the expansion, a set of $n$ simultaneous linear equations are derived, which are solved by matrix algebra.

A brief description of the code is presented along with a limited number of changes that are required to customize the program for a particular task. To evaluate the performance of the method and the goodness of nonlinear curve fitting, two typical engineering problems are examined and the graphical output and the tabular output of each are discussed. A complete listing of the entire package is included as an appendix.

Symbols

$\alpha_j$, $\beta_k$, $\sigma_i$, $\chi^2$, $\chi^2_0$, $\chi^2_2$:

- $\alpha_j$: coefficient of fitting function
- $\beta_k$: algebraic notation for row matrix
- $\sigma_i$: uncertainty in data
- $\chi^2$: global chi-square
- $\chi^2_0$: first term in the expansion of $\chi^2$
- $\chi^2_2$: reduced chi-square

(i, j, k, l): indexes

Introduction

In any area of engineering or physical science, suggested analytical models are accepted only when good statistical correlation exists with a set of experimentally measured values. The correlation is often measured by fitting the mathematical model to a set of experimental data.

Two common methods for fitting data are moving averages and least-squares fit. In the moving averages method, each data point is replaced by the average of itself and $n$ neighboring points on either side of it. The advantage of this method is that it is rather easy to program. One disadvantage is unequal smoothing of the first and the last data points compared with the rest of the data set because of the lack of neighbors on both sides. Another, more important, disadvantage is that the smoothing process is strictly an averaging one and does not produce any analytical representation of the smoothed data.

In the least-squares method, a user-specified fitting function is utilized in such a way to minimize the sum of the squares of distances between the data points and the fitting curve. The advantages of this method are that it permits the generation of statistical information on the goodness of the fit and does not require the data to be collected at regular intervals. The disadvantages are that the method assumes that the basic form of the smoothing equation is known and also, since it is a global procedure, it may be disproportionately biased by a few bad data points, which will twist the shape of the fit to spread the error over the entire data set.

Considering the advantages of the least-squares fitting method and the decreasing expense of computation time, it is often desirable to have a consolidated software package in the form of a single computer program to perform nonlinear curve fitting to a given set of data. This approach should provide the user with statistical information such as goodness of fit and estimated values of parameters that produce the highest degree of correlation between the experimental data and the mathematical model.

The purpose of this paper is to furnish such a software package. The section “Fitting Algorithm Description” describes the mathematical formulation of the quadratic expansion of $\chi^2$, which fundamentally follows the work of Bevington (ref. 1) and in many cases closely parallels his discussion. The section “Program Description” briefly describes the modular characteristics of the program and its associated subroutines and function subprograms. These program elements are formulated around a nonlinear optimization algorithm that calculates the best statistically weighted values of the parameters of the fitting function and the $\chi^2$ that is to be minimized. The program needs as input the mathematical form of the fitting function and the initial values of the parameters to be estimated. The “Notes to Users”
section describes the limited changes a user must make to implement the program for a particular application. The section “Sample Cases” describes two sample cases.

**Fitting Algorithm Description**

Consider the function \( y(z) \) with parameters \( a_j \). For example, \( y(x) \) can be an exponentially decaying sinusoidal function, plus a constant, of the form

\[
y(x) = a_1 e^{-a_2 x} \cos(a_3 x + a_4) + a_5
\]

or, a double Gaussian function, plus a quadratic, of the form

\[
y(x) = a_1 e^{-\frac{1}{2} \left( \frac{x-a_2}{a_3} \right)^2} + a_4 e^{-\frac{1}{2} \left( \frac{x-a_5}{a_6} \right)^2} + a_7 + a_8 x + a_9 x^2
\]

or some other function such that some of the parameters cannot be separated into different terms of a sum.

Bevington (ref. 1) defines \( \chi^2 \), a measure of the goodness of the fit, as

\[
\chi^2 = \sum \left\{ \frac{1}{\sigma_i^2} [y_i - y(x_i)]^2 \right\}
\]

where \( \sigma_i^2 \), the uncertainties in the data points \( y_i \), is defined as

\[
\sigma_i^2 = \frac{1}{n} \sum_{j=1}^{n} (y_{ij} - \bar{y}_i)^2
\]

According to the method of least squares, the simultaneous minimization of \( \chi^2 \) with respect to each of the parameters produces the optimum values of parameters \( a_j \) as

\[
\frac{\partial}{\partial a_j} \chi^2 = \frac{\partial}{\partial a_j} \sum \left\{ \frac{1}{\sigma_i^2} [y_i - y(x_i)]^2 \right\} = 0
\]

Because of the difficulty in deriving an analytical expression to calculate the parameters of \( y(x) \), \( \chi^2 \) is considered as a continuous function of \( n \) parameters \( a_j \) describing a hypersurface in a space of \( n + 1 \) dimensions, where \( a_j \), \( j = 1, 2, \ldots, n \), are the abscissa and \( \chi^2 \) is the ordinate. This space is searched to locate the minimum value of \( \chi^2 \).

In the present paper the search is accomplished through the expansion of \( \chi^2 \) by using an analytical expression for the variation of \( \chi^2 \) to map its variation with respect to parameters \( a_j \). The goal will be to find an approximate analytical function describing the \( \chi^2 \) hypersurface and to use this function to locate the minimum.

**Description of \( \chi^2 \) Expansion**

Consider the linear terms of a Taylor expansion of \( \chi^2 \) as a function of parameters \( a_j \)

\[
\chi^2 \approx \chi^2_0 + \sum_{j=1}^{n} \left( \frac{\partial \chi^2_0}{\partial a_j} \delta a_j \right)
\]

where \( \delta a_j \) are the increments in \( a_j \) required to reach the point at which \( y(x) \) and \( \chi^2 \) are to be evaluated. The \( \chi^2_0 \) is the starting value of \( \chi^2 \) at the point where the value of \( y(x) \) is \( y_0(x) \) such that

\[
\chi^2_0 = \sum \left\{ \frac{1}{\sigma_i^2} [y_i - y_0(x_i)]^2 \right\}
\]

and

\[
y_0(x) = y(x, a_{10}, a_{20}, \ldots, a_{n0})
\]

Since the optimum values for \( a_j \) are defined through the minimization of \( \chi^2 \) with respect to \( a_j \), then

\[
\frac{\partial \chi^2}{\partial a_k} = \frac{\partial \chi^2_0}{\partial a_k} + \sum_{j=1}^{n} \left( \frac{\partial^2 \chi^2_0}{\partial a_j \partial a_k} \delta a_j \right) = 0 \quad (k = 1, 2, \ldots, n)
\]

A set of \( n \) simultaneous linear equations in \( \delta a_j \) are obtained, which algebraically can be written as

\[
\beta_k = \sum_{j=1}^{n} (\delta a_j \alpha_{jk}) \quad (k = 1, 2, \ldots, n)
\]

where

\[
\beta_k = -\frac{1}{2} \frac{\partial^2 \chi^2_0}{\partial a_j \partial a_k}, \quad \alpha_{jk} = \frac{1}{2} \frac{\partial^2 \chi^2_0}{\partial a_j \partial a_k}
\]

One way of looking at equation (8b) is to state that \( \chi^2 \) through the first-order expansion is approximated by a parabolic surface. This is verified by a second-order Taylor expansion of \( \chi^2 \) as a function of \( a_j \)

\[
\chi^2 = \chi^2_0 + \sum_{j=1}^{n} \left( \frac{\partial \chi^2_0}{\partial a_j} \delta a_j \right) + \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \left( \frac{\partial^2 \chi^2_0}{\partial a_j \partial a_k} \delta a_j \delta a_k \right)
\]

which is a second-order function with respect to \( \delta a_j \) and describes a parabolic hypersurface.
Equation (9) indicates that the optimum values of \( \delta a_j \) for which \( \chi^2 \) is a minimum are obtained by requiring that the derivatives with respect to \( a_j \) be zero. Thus,

\[
\frac{\partial^2 \chi^2}{\partial a_k \partial a_k} = \frac{\partial^2 \chi_0^2}{\partial a_k \partial a_k} + \sum_{j=1}^{n} \left( \frac{\partial^2 \chi_0^2}{\partial a_j \partial a_j} \delta a_j \right) = 0 \quad (k = 1, 2, ..., n)
\]

which is the same as equation (7).

The method of quadratic expansion is accurate and precise if the minimum is close to the starting point in such a way that higher order terms in equation (9) can be neglected. But, if the starting point is not close enough, the parabolic approximation of \( \chi^2 \) hypersurface is generally not valid, and in the direction of increasing \( \delta a_j \) the result will be in error. Hence to achieve convergence the algorithm requires meaningful initial estimates for \( a_j \). The initial estimates can often be obtained by visual inspection of data.

**Description of Computational Method**

The analytical methods of the previous section can be used for computational purposes by recognizing that a matrix inversion operation will yield the solution of equation (8) as

\[
\delta a_j = -\sum_{k=1}^{n} \left( \beta_k \varepsilon_{jk} \right)
\]

where \( \varepsilon_{jk} = \sigma_{jk}^{-1} \), and the computation of equation (8b) can be approximated by calculating the variation of \( \chi^2 \) near the starting point \( \chi_0^2 \) and using the standard finite difference equations of first, second, and cross product derivatives of \( \chi_0^2 \) with respect to \( \partial a_j \) and \( \partial a_j \partial a_k \) as

\[
\frac{\partial^2 \chi_0^2}{\partial a_j \partial a_j} \approx \frac{1}{2\Delta a_j} \left[ \chi_0^2 (a_j + \Delta a_j, a_k) - \chi_0^2 (a_j, a_k) \right]
\]

\[
\frac{\partial^2 \chi_0^2}{\partial a_j \partial a_k} \approx \frac{1}{\Delta a_j \Delta a_k} \left[ \chi_0^2 (a_j + \Delta a_j, a_k) - \chi_0^2 (a_j + \Delta a_j, a_k) - \chi_0^2 (a_j, a_k + \Delta a_k) + \chi_0^2 (a_j, a_k) \right]
\]

Finally, the quantity \( \nu \), the number of degrees of freedom left after fitting \( N \) data points to a function of \( n + 1 \) parameters, is defined as

\[
\nu = N - n - 1
\]

Therefore, for \( \nu \) degrees of freedom, the quantity \( \chi_\nu^2 \), the reduced chi-square, is defined as

\[
\chi_\nu^2 = \frac{\chi^2}{\nu}
\]

\( \chi_\nu^2 \) will be used in the computations where \( N \) and \( n \) have specific numerical values.

**Program Description**

The program evolved from the idea of having an interactive package that requires minimum modification by the user. The main program and each subroutine or function subprogram begins with a description of its purpose and a definition of the variables used. The program is 882 lines long and is written in FORTRAN 77. It was developed on the CDC® CYBER 750 scalar mainframe under the NOS 2.3 Level 617 Operating System and requires a minimum of 124008 60-bit words of storage. The entire package is divided into a main program (NLNFIT), five subroutines (CHIFIT, MATINV, PRETTY, CHAR, ERRBAR), and three function subprograms (FCHISQ, FUNCTN, TEXP), with the main program (NLNFIT) containing all EXTERNAL Tektronix (PLOT-10) CALLS (refs. 2 and 3) and Character Generator System (C.G.S.) CALLS (ref. 4). Subroutines CHIFIT and MATINV and function subprogram FCHISQ were originally developed in reference 1 and were modified by the authors. A brief description of the function of main module NLNFIT follows.

**Main Program (NLNFIT)**

NLNFIT assumes that the input data file named "RAWDAT" is written on logical unit 1 (LU = 1) as is specified by the PARAMETER statement. This can easily be changed to another suitable value if LU = 1 is a reserved unit.

For the sake of transportability, the data file is limited to only four sets of input. The first card is an integer specifying the number of data pairs.
and is optionally set at 200. The second card is an integer flag with values +1, 0, or -1, depending on whether the input data are to be weighted or not. For instrumental weight, where the uncertainty in each measurement of \( y_i \) generally comes from fluctuations in repeated readings of instrumental scale, the input weight flag should be set to +1. The choice of instrumental weight requires that the user input data points \( (x_i, y_i) \) and uncertainty \( \Delta y_i \). If it is decided not to weight the input data, integer flag 0 must be chosen. For statistical weight, where it is generally true that the uncertainty in each measurement \( y_i \) is proportional to \( |y_i|^{-1} \) and therefore the standard deviations \( \sigma_i \) associated with these measurements cannot be considered equal over any reasonable range of values, an integer flag of -1 must be chosen. The third card is the form of the fitting equation and will be read by the main module in an 80A1 format. The actual data pairs are the fourth input and are read in the form \( (x_i, y_i) \) for no weight or statistical weight or \( (x_i, y_i, \Delta y_i) \) for instrumental weight.

**Notes to Users**

This section describes what changes a user must make to each routine (appendix A) to use the program for a different fitting function.

**NLFIT**

The PARAMETER statement is the only change that is required for the main program. In the PARAMETER statement, II indicates the maximum number of data pairs, JJ must always be \( 4*II \), KK is the maximum number of characters in the X and Y title statements, LL is the number of \( a_j \), IBAUD is the baud/10 rate of graphics display device, ITER is the maximum number of iterations allowed, and LU = 1 is the logical unit for input data.

**CHIFIT**

In CHIFIT, only the value of LL in the PARAMETER statement must be changed.

**FUNCTN**

In FUNCTN, the value of LL in the PARAMETER statement and the form of the FUNCTN statement must be changed.

**MATINV**

In MATINV, only the value of LL in the PARAMETER statement must be changed.

**Sample Cases**

Two sample cases in classical and fluid mechanics, weighted statistically (-1) and instrumentally (+1), respectively, are analyzed with the program package. Each case is described below, and its computer output is given as an appendix.

**Sample Case 1: Classical Mechanics—Physical Pendulum**

The circles in figure 1 are 166 data pairs obtained through an 8-bit A/D converter in a pendulum calibration test conducted by the authors. A 5-parameter nonlinear fitting function of the form

\[
A(t) = A_1 e^{-t/t_m^1} \cos(\omega t + \delta) + A_2
\]

was applied to the data. Equation (15) is similar to equation (1a), with \( A_2 = t_m^{-1} \), \( \omega \) and \( \delta \) as angular frequency and phase, and \( A_2 \) as contribution due to damping factors such as the frictional forces in the support bearings. The solid line is the best fit to the data. This particular functional form (eq. (15)), with
initial $a_j$ estimates listed in appendix B, produced $\chi^2 \approx 0.12$ in six iterations. Appendix B lists the interactive session for sample case 1.

Sample Case 2: Fluid Mechanics—Far-Field Wind-Tunnel Pressure Analysis

The circles in figure 2 are 22 data pairs representing the nondimensional pressure coefficients measured near the top wall of a two-dimensional wind tunnel. A 6-in-chord airfoil model was mounted on the tunnel centerline between $x = -3$ in. and $x = +3$ in. The variation of the data is the result of the expansion of the flow about the model and a flow angularity probe inserted in the airstream at $x = 6$ in. near the top wall. The data were measured approximately 3.5 chord lengths above the model.

A 9-parameter nonlinear fitting function of the form

$$A(x) = A_1 e^{-\frac{1}{2} \left( \frac{x-\mu_1}{\sigma_1} \right)^2} + A_4 e^{-\frac{1}{2} \left( \frac{x-\mu_2}{\sigma_2} \right)^2} + A_7 + A_8 x + A_9 x^2$$

(16)

was applied to the data. Equation (16) is similar to equation (1b), with $a_2 = \mu_1$, $a_3 = \sigma_1$, $a_5 = \mu_2$, and $a_6 = \sigma_2$ as the mean $\mu$ and standard deviation $\sigma$ of each Gaussian peak, and $A_7$, $A_8$, and $A_9$ are the background contributions due to the undisturbed flow in the tunnel in the absence of the airfoil. The solid line is the best fit to the data. This particular functional form (eq. (16)), with initial $a_j$ estimates listed in appendix C, produced $\chi^2 \approx 0.69$ in four iterations. Appendix C lists the interactive session for sample case 2 with initial data listed as X-DATA, Y-DATA, and fitted data listed as YFIT.

Concluding Remarks

The theoretical description of an all-purpose curve-fitting routine based on quadratic expansion of $\chi^2$ was presented. Taylor's expansion of $\chi^2$ was introduced, and from the expansion a set of $n$ simultaneous linear equations were derived and solved by matrix algebra. The associated interactive, graphics-based computer program and sample cases indicated the relatively fast convergence rate of the method. Guidelines on how to customize the program for a particular task were given and fully described.

NASA Langley Research Center
Hampton, Virginia 23665-5225
September 8, 1987
Figure 1. Application in classical mechanics.

Figure 2. Application in fluid mechanics.
Appendix A

Program Listing of Nonlinear Fitting Program NLNFIT

Appendix A contains the program listing of the nonlinear fitting program NLNFIT, which consists of the main program NLNFIT, five subroutines CHIFIT, MATINV, PRETTY, CHAR, ERRBAR, and three function subprograms FCHISQ, FUNCTN, TEXP.
PROGRAM NLNFIT

PURPOSE
MAIN PROGRAM TO MAKE A LEAST SQUARES FIT TO A NON-LINEAR
FUNCTION WITH A QUADRATIC EXPANSION OF CHI. SQUARE

DESCRIPTION OF PARAMETERS
II - MAX. NO. OF DATA POINTS (200)
JJ - 4 TIMES THE NUMBER OF DATA POINTS, USED FOR PLOTTING
A SMOOTH FIT THROUGH DATA POINTS
KK - MAX. NO. OF ALPHABETIC CHARACTERS IN TITLES STATEMENTS
LU - LOGIC UNIT OF I/O FOR INPUT DATA FILE
LL - NO. OF COEFFICIENTS OF FITTING FUNCTION
X - ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
Y - ARRAY OF DATA POINTS FOR DEPENDENT VARIABLE
XDATA - DUMMY ARRAY TO STORE INDEPENDENT DATA POINTS
YDATA - DUMMY ARRAY TO STORE DEPENDENT DATA POINTS
YBU - DUMMY ARRAY TO STORE SIGMAY
YBD - DUMMY ARRAY TO STORE SIGMAY
YFIT- ARRAY OF CALCULATED VALUES OF Y
SIGMAY - ARRAY OF STANDARD DEVIATIONS FOR Y DATA POINTS
A - ARRAY OF PARAMETERS
SIGMAA - ARRAY OF STANDARD DEVIATIONS FOR PARAMETERS A
DELTAA - ARRAY OF INCREMENTS FOR PARAMETERS A
XLABEL - ARRAY TO STORE TITLE OF Y-AXIS
YLABEL - ARRAY TO STORE TITLE OF X-AXIS
IXLAB - DUMMY ARRAY FOR X-TITLE
IYLAB - DUMMY ARRAY FOR Y-TITLE
TITLE - ARRAY TO STORE FITTING FUNCTION
NPTS - NUMBER OF PAIRS OF DATA POINTS
MODE - DETERMINES METHOD OF WEIGHTING LEAST SQUARES FIT
  +1 (INSTRUMENTAL) WEIGHT(I)=1./SIGMAY(I)**2
  0 (NO. WEIGHT) WEIGHT(I)=1.0
  -1 (STATISTICAL) WEIGHT(I)=1./Y(I)
INTERMS - NUMBER OF PARAMETERS
CHISQR - REDUCED CHI. SQUARE FOR FIT
IBAUD - BAUD/10 RATE OF GRAPHICS DISPLAY DEVICE
ITER - NO. OF ITERATIONS TO CONVERGE (20)

PROGRAM NLNFIT(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)

PARAMETER(II=200, JJ=4*II, KK=80, ITER=20, LU=1, LL=9, IBAUD=960)

DIMENSION X(JJ), Y(JJ), XDATA(JJ+1), YDATA(JJ+1), YFIT(JJ), SIGMAY(JJ)
DIMENSION YBU(JJ+1), YBD(JJ+1)
DIMENSION A(LL), SIGMAA(LL), DELTAA(LL)
DIMENSION YLABEL(KK), XLABEL(KK), IXLAB(KK), IYLAB(KK), TITLE(KK)

WRITE(6,10)

OPEN(UNIT=LU, ACCESS=‘SEQUENTIAL’, FILE=‘RAWDAT’)

REWIND LU

READ(LU,20) NPTS
READ(LU,20) MODE
READ(LU,270)(TITLE(I), I=1, KK)
C DO 30 I=1,NPTS
IF((MODE.EQ.0).OR.(MODE.EQ.-1)) THEN
READ(LU,*) X(I),Y(I)
ELSE IF(MODE.EQ.1) THEN
READ(LU,*) X(I),Y(I),SIGMAY(I)
END IF
END IF
30 CONTINUE
C CLOSE(UNIT=LU)
C IF(NPTS.GT.11) THEN
WRITE(6,40)
STOP
END IF
WRITE(6,50) NPTS
NTERMS=LL
KEEN=0
WRITE(6,60)
WRITE(6,270)(TITLE(I),I=1,WW)
WRITE(6,70) NTERMS
C DO 80 I=1,NTERMS
WRITE(6,90) I
READ(5,*) A(I)
80 CONTINUE
C WRITE(6,100)
WRITE(6,110) (I,A(I),I=1,NTERMS)
WRITE(6,120)
C DO 130 I=1,NTERMS
DELTA(I)=A(I)*.01
130 CONTINUE
C KOUNT=0
C BEGIN ITERATION
C DO 140 K=1,ITER
CALL CHIFIT(X,Y,SIGMAY,NPTS,NTERMS,MODE,A,DELTA,SIGMAA,
- YFIT,CHISGR)
WRITE(6,150) K,CHISGR
IF(K.GT.1) THEN
GO TO 160
END IF
SAVE=CHISGR
KOUNT=1
GO TO 140
160 XCHI=CHISGR-SAVE
IF(ABS(XCHI).LT.0.01) THEN
WRITE(6,175)
GO TO 180
END IF
SAVE=CHISGR
KOUNT=KOUNT+1
140 CONTINUE
180 KOUNT = KOUNT + 1
WRITE(6, 170) KOUNT
WRITE(6, 190)
WRITE(6, 270) (TITLE(I), I = 1, KK)
WRITE(6, 45)
C
DO 200 I = 1, NTERMS
WRITE(6, 210) I, A(I)
200 CONTINUE
C
WRITE(6, 220) CHISGR
WRITE(6, 230)
READ(5, *) IANS1
IF(IANS1 .EQ. 1) THEN
GO TO 240
END IF
490 WRITE(6, 250)
READ(5, *) IANS2
IF(IANS2 .EQ. 0) THEN
STOP
END IF
WRITE(6, 260)
READ(5, 270) (YLABEL(I), I = 1, KK)
WRITE(6, 280)
READ(5, 270) (XLABEL(I), I = 1, KK)
WRITE(6, 290)
READ(5, *) IANS
IF(IANS .EQ. 2) THEN
GO TO 300
END IF
GO TO 310
300 WRITE(6, 320)
READ(5, *) INSL
310 CONTINUE
WRITE(6, 330)
READ(5, *) IANSRT
IF(IANSRT .EQ. 0) THEN
GO TO 340
END IF
WRITE(6, 350)
READ(5, *) ISYMB
340 CONTINUE
WRITE(6, 360)
READ(5, *) NSETX
IF(NSETX .NE. 1) THEN
GO TO 370
END IF
WRITE(6, 380)
READ(5, *) XMIN, XMAX
370 WRITE(6, 390)
READ(5, *) NSETY
IF(NSETY .NE. 1) THEN
GO TO 400
END IF
WRITE(6, 410)
READ(5, *) YMIN, YMAX
400 CONTINUE
START OF TEKTRONIX PLOT-10 GRAPHICS CALLS

CALL INITT( IBAUD)
CALL BINITT
CALL XNEAT(1)
CALL YNEAT(1)
XDATA(1)=FLOAT(4*NPTS)
YDATA(1)=FLOAT(4*NPTS)
YBU(1)=FLOAT(4*NPTS)
YBD(1)=FLOAT(4*NPTS)

FILL DUMMY ARRAY DATA POINTS

DO 420 I=2,4*NPTS+1,4
  KEEN=KEEN+1
  XDATA(I)=X(KEEN)
  XDATA(I+1)=X(KEEN)
  XDATA(I+2)=X(KEEN)
  XDATA(I+3)=X(KEEN)
  YDATA(I)=Y(KEEN)
  YDATA(I+1)=Y(KEEN)
  YDATA(I+2)=Y(KEEN)
  YDATA(I+3)=Y(KEEN)
  IF(MODE.EQ.1) THEN
    YBD(I)=Y(KEEN)-SIGMAY(KEEN)
    YBD(I+1)=Y(KEEN)-SIGMAY(KEEN)
    YBD(I+2)=Y(KEEN)-SIGMAY(KEEN)
    YBD(I+3)=Y(KEEN)-SIGMAY(KEEN)
    YBU(I)=Y(KEEN)+SIGMAY(KEEN)
    YBU(I+1)=Y(KEEN)+SIGMAY(KEEN)
    YBU(I+2)=Y(KEEN)+SIGMAY(KEEN)
    YBU(I+3)=Y(KEEN)+SIGMAY(KEEN)
  END IF
  CONTINUE

IF(INSL.EQ.1) CALL YTYPE(2)
IF(INSL.EQ.2) CALL XTYPE(2)
IF(IANS.EQ.3) CALL YTYPE(2)
IF(IANS.EQ.3) CALL XTYPE(2)
CALL ZLINE(-4)
CALL SYMBL(ISYMB)
CALL XFRM(3)
CALL XMFRM(3)
CALL YFRM(3)
CALL YMFRM(3)
IF(NSETX.EQ.1) CALL XNEAT(0)
IF(NSETY.EQ.1) CALL YNEAT(0)
IF(NSETX.EQ.1) CALL DLIMX(XMIN, XMAX)
IF(NSETY.EQ.1) CALL DLIMY(YMIN, YMAX)
CALL CHECK(XDATA, YDATA)
CALL DSPLAY(XDATA, YDATA)

IF(MODE.EQ.1) THEN
  CALL ERRBAR(XDATA, YBU, YBD)
END IF
X1 = X(1)
X2 = X(NPTS)
XINC = (X2 - X1) / FLOAT(4*NPTS)
IN = 0

DO 430 XV = X1, X2, XINC
IN = IN + 1
X(IN) = XV
YFIT(IN) = FUNCTN(X, IN, A)
CONTINUE

DO 440 I = 2, IN
XDATA(I) = X(I)
YDATA(I) = YFIT(I)
CONTINUE

CALL ZLINE(0)
CALL SYMBOL(0)
CALL CPLOT(XDATA, YDATA)

CALL PRETTY(YLABEL, XLABEL, IYLEN, IXLEN, IXLAB, IYLAB)
IVY = IFIX((575. - 13. * IYLEN) / 2.) + 125
CALL KA12AS(50, YLABEL, IYLAB)
CALL CHAR(20, IVY, IYLAB, 50, 90., 1.)
IVX = IFIX((750. - 13. * IXLEN) / 2.) + 150
CALL KA12AS(50, XLABEL, IXLAB)
CALL CHAR(IVX, 20, IXLAB, 50, 90., 1.)
CALL FRAME
CALL BELL
CALL TINPUT(I)
CALL ERASE
CALL FINITT(0, 700)
STOP

DO 460 I = 1, NPTS
DIFF = ((Y(I) - YFIT(I)) / Y(I)) * 100.0
TRACK = TRACK + DIFF
WRITE(6, 470) X(I), Y(I), YFIT(I), DIFF
CONTINUE

WRITE(6, 480) TRACK
GO TO 490

FORMAT(/, 'N O N L I N E A R C U R V E-F I T T I N G C O D E',/) 
FORMAT(/, 'N U M B E R O F D A T A P A I R S = ', I3) 
FORMAT(/, 'C H O S E N F I T T I N G F U N C T I O N I S : ',/)
70 FORMAT( 'ENTER INITIAL GUESSES FOR THE A1->A', I1, ' PARAMETERS', '/)
90 FORMAT( 'FOR A', I1, ' ENTER GUESS?', '/)
150 FORMAT( 'FINISHED ITERATION #', I2, ' WITH REDUCED CHI. SQ. = ', IPE13.4)
175 FORMAT( 'ITERATION STOPPED BECAUSE ABS(XCHI) LT. 0.01', '/)
260 FORMAT( 'INPUT TITLE OF Y - AXIS', '/)
280 FORMAT( 'INPUT TITLE OF X - AXIS', '/)
270 FORMAT(80A1)
110 FORMAT(5X, 'A(', I1, ')=', IPE14.6)
100 FORMAT( 'STARTING VALUES' '/)
170 FORMAT( 'THERE WERE ', I3, ' ITERATIONS', '/)
190 FORMAT( 'USING ', '/)
45 FORMAT( 'THE FINAL COEFFICIENTS ARE', '/)
210 FORMAT(5X, 'A(', I1, ')=', IPE14.6)
220 FORMAT( 'WITH REDUCED CHI. SQUARE=', IPE16.6)
230 FORMAT( 'DO YOU WANT A DATA REVIEW? <1= YES, O= NO>', '/)
250 FORMAT( 'DO YOU WANT TO PLOT DATA? <1= YES, O= NO>', '/)
480 FORMAT( 'MEAN OF X ERROR=', F14.6)
470 FORMAT( 'INPUT THE NUMBER OF YOUR SELECTION?', '/)
490 FORMAT( 'DO YOU WANT SPECIAL SYMBOLS TO DENOTE DATA POINTS?', '/)
500 FORMAT( 'DO YOU WANT TO SET THE X RANGE', '/)
510 FORMAT( 'INPUT XMIN, XMAX?', '/)
520 FORMAT( 'DO YOU WANT TO SET THE Y RANGE', '/)
530 FORMAT( 'INPUT YMIN, YMAX?', '/)
END
C
C******************************************************************************
FUNCTION FUNCTN(X, I, A)******************************************************************************
C
PURPOSE
EVALUATE TERMS OF FUNCTION FOR NON-LINEAR LEAST-SQUARES SEARCH

USAGE
RESULT=FUNCTN(X, I, A)

DESCRIPTION OF PARAMETERS
LL - NO. OF COEFFICIENTS OF FITTING FUNCTION
X - ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
I - INDEX OF DATA POINT
A - ARRAY OF PARAMETERS

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

FUNCTION FUNCTN(X, I, A)

PARAMETER(LL=9)

DIMENSION X(1), A(LL)

XI=X(I)
FUNCTN=A(1)*TEXP(-A(2)*XI)*COS(A(3)*XI+A(4))+A(5)
FUNCTN=A(1)*TEXP(-0.5*((XI-A(2))/A(3))**2)+
- A(4)*TEXP(-0.5*((XI-A(5))/A(6))**2)+
- A(7)+A(8)*XI+A(9)*XI**2
RETURN
END

C
C******************************************************************************
FUNCTION TEXP(X)******************************************************************************
C
PURPOSE
TO ELIMINATE OVER/UNDER FLOW OF CPU IF EXP IS USED

USAGE
TEXP=EXP(X)

FUNCTION TEXP(X)

IF(X .LT. -100. ) X=-100.
IF(X .GT. 100. ) X=100.
TEXP=EXP(X)
END
SUBROUTINE PRETTY(YLABEL, XLABEL, IYLEN, IXLEN, IXLAB, IYLAB)

PURPOSE
TO SEARCH THROUGH X AND Y TITLE AND COUNT THE NUMBER OF CHARACTERS

USAGE
CALL PRETTY(YLABEL, XLABEL, IYLEN, IXLEN, IXLAB, IYLAB)

DESCRIPTION OF PARAMETERS
YLABEL - ARRAY OF Y-TITLE
XLABEL - ARRAY OF X-TITLE
IYLEN - LENGTH OF ARRAY YLABEL
IXLEN - LENGTH OF ARRAY XLABEL
IYLAB - ARRAY CONTAINING INTEGER EQUIVALENT OF YLABEL
IXLAB - ARRAY CONTAINING INTEGER EQUIVALENT OF XLABEL

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
CALL KAI2AS(50, YLABEL, IYLAB) : KAI2AS IS A PLOT-10 CALL
SUBROUTINE PRETTY(YLABEL, XLABEL, IX, IXLAB, IYLAB)

DIMENSION YLABEL(1), XLABEL(1), IXLAB(1), IYLAB(1)

NCHAR = 32
CALL KAI2AS(50, YLABEL, IYLAB)
CALL KAI2AS(50, XLABEL, IXLAB)

DO 10 I=1,50
   IF((IYLAB(I).EQ.NCHAR).AND.(IYLAB(I+1).EQ.NCHAR).AND. 
     -(IYLAB(I+2).EQ.NCHAR)) IY=I
   IF((IYLAB(I).EQ.NCHAR).AND.(IYLAB(I+1).EQ.NCHAR).AND. 
     -(IYLAB(I+2).EQ.NCHAR)) GO TO 20
10 CONTINUE

20 DO 30 I=1,50
     -(IXLAB(I+2).EQ.NCHAR)) IX=I
     -(IXLAB(I+2).EQ.NCHAR)) GO TO 40
30 CONTINUE

40 RETURN
END
SUBROUTINE CHAR(LOCX, LOCY, ISTRNG, NCHAR, ANG, SIG)

PURPOSE

REQUIRED PLOT-10 SUBROUTINE (CHARACTER GENERATION PACKAGE)

USAGE

CALL CHAR(LOCX, LOCY, ISTRNG, NCHAR, ANG, SIG)

DESCRIPTION OF PARAMETERS

LOCX - INTEGER VALUE OF LOCATION OF XDOT (0-->1024)
LOCY - INTEGER VALUE OF LOCATION OF YDOT (0-->780)
ISTRNG - ARRAY CONTAINING TITLE STRING
NCHAR - NO. OF CHARACTER IN ISTRNG
ANG - ROTATION ANGLE FOR PLOTTING CHARACTER
SIG - SIZE OF PLOTTED CHARACTER

DIMENSION ISTRNG(NCHAR)

REAL COMST(60)

CALL SVSTAT(COMST)
CALL RESET
CALL BINITT
CALL MOVABS(LOCX,LOCY)
CALL RROTAT(ANG)
CALL ZRSCALE(SIG)

DO 10 I=1,NCHAR
  CALL LCHAR(ISTRNG(I))
10 CONTINUE

CALL RESTAT(COMST)
RETURN
END
C
C**************************** SUBROUTINE ERRBAR(XDATA, YBU, YBD) ****************************
C
C     PURPOSE
C     TO DRAW ERROR BAR IF MODE=1
C
C     USAGE
C     CALL ERRBAR(XDATA, YBU, YBD)
C
C     DESCRIPTION OF PARAMETERS
C     XDATA - DUMMY ARRAY TO STORE INDEPENDENT DATA POINTS
C     YBU - DUMMY ARRAY TO STORE SIGMA
C     YBD - DUMMY ARRAY TO STORE SIGMA
C     SUBROUTINE ERRBAR(XDATA, YBU, YBD)
C
C     DIMENSION XDATA(1), YBU(1), YBD(1)
C
C
C     NDATA=XDATA(1)
C     XMIN=1. E+99
C     XMAX=-XMIN
C
C     DO 5 I=2, NDATA+1
C     IF(XDATA(I), LT, XMIN) XMIN=XDATA(I)
C     IF(XDATA(I). GT. XMAX) XMAX=XDATA(I)
C     CONTINUE
C
C     WEERB=(XMAX-XMIN)/(2. 0*FLOAT(NDATA))
C
C     DO 10 I=2, NDATA+1
C     XL=XDATA(I)-WEERB
C     XR=XDATA(I)+WEERB
C     CALL MOVEA(XL, YBU(I))
C     CALL DRAWA(XR, YBU(I))
C     CONTINUE
C
C     DO 15 I=2, NDATA+1
C     CALL MOVEA(XDATA(I), YBU(I))
C     CALL DRAWA(XDATA(I), YBD(I))
C     CONTINUE
C
C     DO 20 I=2, NDATA+1
C     XL=XDATA(I)-WEERB
C     XR=XDATA(I)+WEERB
C     CALL MOVEA(XL, YBD(I))
C     CALL DRAWA(XR, YBD(I))
C     CONTINUE
C
C     RETURN
C     END
SUBROUTINE CHIFIT(X, Y, SIGMAY, NPTS, NTERMS, MODE, A, DELTAA, SIGMAA, YFIT, CHISQR)

PURPOSE
-make a least-squares fit to a non-linear function
with a parabolic expansion of chi-square

SOURCE
-data reduction and error analysis for the physical sciences
-p. r. bevington

USAGE
-call CHIFIT(X, Y, SIGMAY, NPTS, NTERMS, MODE, A, DELTAA, SIGMAA, YFIT, CHISQR)

DESCRIPTION OF PARAMETERS
LL - no. of coefficients of fitting function
X - array of data points for independent variable
Y - array of data points for dependent variable
SIGMAY - array of standard deviations for Y data points
NPTS - number of pairs of data points
NTERMS - number of parameters
MODE - determines method of weighting least-squares fit
   +1 (instrumental) weight(I) = 1./SIGMAY(I)**2
   0 (no weighting) weight(I) = 1.0
   -1 (statistical) weight(I) = 1./Y(I)
A - array of parameters
DELTAA - array of increments for parameters A
SIGMAA - array of standard deviations for parameters A
YFIT - array of calculated values of Y
CHISQR - reduced chi-square for fit

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
FUNCTN(X, I, A)
evaluates the fitting function for the I-th term
FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)
evaluates reduced chi-square for fit to data
MATINV(ARRAY, NTERMS, DET)
inverts a symmetric two-dimensional matrix of degree NTERMS and calculates its determinants

COMMENTS
-dimension statement valid for NTERMS is changed by parameter statement

dimension X(1), Y(1), SIGMAY(1), A(1), DELTAA(1), SIGMAA(1), YFIT(1)
dimension ALPHA(LL, LL), BETA(LL, LL), DA(LL)
C
11 NFREE=NPTS- NTERMS
FREE=NFREE
IF(NFREE) 14, 14, 16
14 CHISQR=0.
GO TO 120
C
16 DO 17 I=1, NPTS
17 YFIT(I)=FUNCTN(X, I, A)
C
18 CHISQ1=FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)
C
EVALUATE ALPHA AND BETA MATRICES
C
20 DO 60 J=1, NTERMS
C
A(J)+DELTAA(J)
C
21 AJ=A(J)
A(J)=AJ+DELTAA(J)
C
DO 24 I=1, NPTS
24 YFIT(I)=FUNCTN(X, I, A)
C
CHISQ2=FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)
ALPHA(J, J)=CHISQ2-2. *CHISQ1
BETA(J)=-CHISQ2
C
31 DO 50 K=1, NTERMS
IF(K-J) 33, 50, 36
33 ALPHA(K, J)=(ALPHA(K, J)-CHISQ2)/2.
ALPHA(J, K)=ALPHA(K, J)
GO TO 50
36 ALPHA(J, K)=CHISQ1-CHISQ2
C
A(J)+DELTAA(J) AND A(K)+DELTAA(K)
C
41 AK=A(K)
A(K)=AK+DELTAA(K)
C
DO 44 I=1, NPTS
44 YFIT(I)=FUNCTN(X, I, A)
C
CHISQ3=FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)
ALPHA(J, K)=ALPHA(J, K)+CHISQ3
A(K)=AK
50 CONTINUE
C
A(J)-DELTAA(J)
C
51 A(J)=AJ-DELTAA(J)
DO 53 I=1,NPTS
53 YFIT(I)=FUNCTN(X, I, A)
C
CHISQ3=FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)
A(J)=AJ
ALPHA(J, J)=(ALPHA(J, J)+CHISQ3)/2.
BETA(J)=(BETA(J)+CHISQ3)/4.
60 CONTINUE
C
ELIMINATE NEGATIVE CURVATURE
C
61 DO 70 J=1,NTERMS
62 IF(ALPHA(J, J)<0.01) 63,65,70
63 ALPHA(J, J)=-ALPHA(J, J)
GO TO 66
65 ALPHA(J, J)=0.01
66 DO 72 K=1,NTERMS
67 IF(K-J) 68,72,68
68 ALPHA(J, K)=0.0
ALPHA(K, J)=0.0
72 CONTINUE
70 CONTINUE
C
INVERT MATRIX AND EVALUATE PARAMETER INCREMENTS
C
71 CALL MATINV(ALPHA, NTERMS, DET)
DO 76 J=1,NTERMS
74 DO 75 K=1,NTERMS
75 DA(J)=DA(J)+BETA(K)*ALPHA(J, K)
76 DA(J)=0.2*DA(J)*DELTAA(J)
C
MAKE SURE CHI. SQUARE DECREASES
C
81 DO 82 J=1,NTERMS
82 A(J)=A(J)+DA(J)
83 DO 84 I=1,NPTS
84 YFIT(I)=FUNCTN(X, I, A)
C
CHISQ2=FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)
IF(CHISQ1-CHISQ2)<0.91,91
87 DO 89 J=1,NTERMS
88 DA(J)=DA(J)/2.
89 A(J)=A(J)-DA(J)
GO TO 83

INCREMENT PARAMETERS UNTIL CHI. SQUARE STARTS TO INCREASE

91 DO 92 J=1, NTERMS
92 A(J)=A(J)+DA(J)

DO 94 I=1, NPTS
94 YFIT(I)=FUNCTN(X, I, A)

CHISQ3=FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)
IF(CHISQ3-CHISQ2) 97, 101, 101
97 CHISQ1=CHISQ2
98 CHISQ2=CHISQ3
99 GO TO 91

FIND MINIMUM OF PARABOLA DEFINED BY LAST THREE POINTS

101 DELTA=1./((1.+(CHISQ1-CHISQ2)/(CHISQ3-CHISQ2))+.5

DO 104 J=1, NTERMS
104 SIGMAA(J)=DELTA*DA(J)
104 SIGMAA(J)=DELTA*DA(J)*SQRT(FREE*DABS(ALPHA(J, J)))

DO 106 I=1, NPTS
106 YFIT(I)=FUNCTN(X, I, A)

CHISQ2=FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)
111 IF(CHISQ2-CHISQ3) 112, 120, 120

DO 113 J=1, NTERMS
113 A(J)=A(J)+(DELTA-1.)*DA(J)

DO 115 I=1, NPTS
115 YFIT(I)=FUNCTN(X, I, A)

CHISQ3=CHISQ2
120 RETURN
END
FUNCTION FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)

PURPOSE
EVALUATE REDUCED CHI-SQUARE FOR FIT TO DATA
FCHISQ = SUM((Y - YFIT)**2/SIGMA**2)/NFREE

SOURCE
DATA REDUCTION AND ERROR ANALYSIS FOR THE PHYSICAL SCIENCES
P. R. BEVINGTON

USAGE
RESULT = FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)

DESCRIPTION OF PARAMETERS
Y - ARRAY OF DATA POINTS
SIGMAY - ARRAY OF STANDARD DEVIATIONS FOR DATA POINTS
NPTS - NUMBER OF DATA POINTS
NFREE - NUMBER OF DEGREES OF FREEDOM
MODE - DETERMINES METHOD OF WEIGHTING LEAST-SQUARES FIT
   +1 (INSTRUMENTAL) WEIGHT(I) = 1./SIGMAY(I)**2
   0 (NO WEIGHTING) WEIGHT(I) = 1.0
   -1 (STATISTICAL) WEIGHT(I) = 1./Y(I)
YFIT - ARRAY OF CALCULATED VALUES OF Y

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

FUNCTION FCHISQ(Y, SIGMAY, NPTS, NFREE, MODE, YFIT)

DOUBLE PRECISION CHISQ, WEIGHT
DIMENSION Y(1), SIGMAY(1), YFIT(1)

11 CHISQ = 0.
12 IF(NFREE) 13, 13, 20
13 FCHISQ = 0.0
GO TO 40

14 ACCUMULATE CHI-SQUARES

20 DO 30 I = 1, NPTS
21 IF(MODE) 22, 27, 29
22 IF(Y(I)) 25, 27, 23
23 WEIGHT = 1./Y(I)
GO TO 30
25 WEIGHT = 1./(-Y(I))
GO TO 30
27 WEIGHT = 1.
GO TO 30
29 WEIGHT = 1./SIGMAY(I)**2
30 CHISQ = CHISQ + WEIGHT*(Y(I) - YFIT(I))**2

31 FREE = NFREE
32 FCHISQ = CHISQ/FREE
40 RETURN
END
C

C*************** SUBROUTINE MATINV(ARRAY,NORDER,DET) ***************
C
C PURPOSE
INVERT A SYMMETRIC MATRIX AND CALCULATE ITS DETERMINANT
C
C SOURCE
DATA REDUCTION AND ERROR ANALYSIS FOR THE PHYSICAL SCIENCES
P. R. BEVINGTON
C
C USAGE
CALL MATINV(ARRAY,NORDER,DET)
C
C DESCRIPTION OF PARAMETERS
LL - NO. OF COEFFICIENTS OF FITTING FUNCTION
ARRAY - INPUT MATRIX WHICH IS REPLACED BY ITS INVERSE
NORDER - DEGREE OF MATRIX (ORDER OF DETERMINANT)
DET - DETERMINANT OF INPUT MATRIX
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE
C
C COMMENTS
DIMENSION STATEMENT VALID FOR NORDER IS CHANGED BY PARAMETER STATEMENT
C
C SUBROUTINE MATINV(ARRAY,NORDER,DET)
C
PARAMETER(LL=9)
C
DOUBLE PRECISION ARRAY, AMAX, SAVE
C
DIMENSION ARRAY(LL,LL), IK(LL), JK(LL)

10 DET=1.0

11 DO 100 K=1,NORDER
C
C FIND LARGEST ELEMENT ARRAY(I,J) IN REST OF MATRIX
C
AMAX=0.

21 DO 30 I=K,NORDER
      DO 30 J=K,NORDER
23 IF(DABS(AMAX)-DABS(ARRAY(I,J))) 24,24,30
24 AMAX=ARRAY(I,J)
      IK(K)=I
      JK(K)=J
30 CONTINUE
C
C INTERCHANGE ROWS AND COLUMNS TO PUT AMAX IN ARRAY(K,K)
C
31 IF(AMAX) 41,32,41
32 DET=0.0
      GO TO 140
41 I=IK(K)
      IF(I-K) 21,51,43
C
DO 50 J=1,NORDER
SAVE=ARRAY(K,J)
ARRAY(K,J)=ARRAY(I,J)
50 ARRAY(I,J)=-SAVE

C

J=JK(K)
IF(J-K) 21, 61, 53
C

DO 60 I=1,NORDER
SAVE=ARRAY(I,K)
ARRAY(I,K)=ARRAY(I,J)
60 ARRAY(I,J)=-SAVE

C ACCUMULATE ELEMENTS OF INVERSE MATRIX

DO 70 I=1,NORDER
IF(I-K) 63, 70, 63
70 CONTINUE

DO 80 I=1,NORDER
DO 80 J=1,NORDER
IF(I-K) 74, 80, 74
70 IF(J-K) 75, 80, 75
75 ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
80 CONTINUE

DO 90 J=1,NORDER
IF(J-K) 83, 90, 83
90 CONTINUE

ARRAY(K,K)=1./AMAX
100 DET=DET*AMAX

C RESTORE ORDERING OF MATRIX

DO 110 L=1,NORDER
K=NORDER-L+1
J=IK(K)
IF(J-K) 111, 111, 105
105 DO 110 I=1,NORDER
SAVE=ARRAY(I,K)
ARRAY(I,K)=-ARRAY(I,J)
110 ARRAY(I,J)=SAVE

C

I=JK(K)
IF(I-K) 130, 130, 113
C

DO 120 J=1,NORDER
SAVE=ARRAY(K,J)
ARRAY(K,J)=-ARRAY(I,J)
120 ARRAY(I,J)=SAVE

130 CONTINUE

140 RETURN
END
Appendix B

Sample Case 1: Application in Classical Mechanics

Appendix B is the complete listing of an interactive session for a fitting function of five parameters.
NONLINEAR CURVE-FITTING CODE

NUMBER OF DATA PAIRS = 166

CHOOSEN FITTING FUNCTION IS:

\[ Y = A \exp(-Bx) \cos(Cx+D) + E \]

ENTER INITIAL GUESSES FOR THE A1-->A5 PARAMETERS

FOR A(1) ENTER GUESS?
? 68.0
FOR A(2) ENTER GUESS?
? 0.002
FOR A(3) ENTER GUESS?
? 0.1
FOR A(4) ENTER GUESS?
? -2.0
FOR A(5) ENTER GUESS?
? 100.0

STARTING VALUES

\[
\begin{align*}
A(1) &= 6.800000 \times 10^1 \\
A(2) &= 2.000000 \times 10^{-3} \\
A(3) &= 1.000000 \times 10^{-1} \\
A(4) &= -2.000000 \times 10^0 \\
A(5) &= 1.000000 \times 10^2 \\
\end{align*}
\]

FINISHED ITERATION # 1 WITH REDUCED CHI.SQ. = 2.3147E+01
FINISHED ITERATION # 2 WITH REDUCED CHI.SQ. = 1.9383E+01
FINISHED ITERATION # 3 WITH REDUCED CHI.SQ. = 4.8630E+00
FINISHED ITERATION # 4 WITH REDUCED CHI.SQ. = 2.0090E-01
FINISHED ITERATION # 5 WITH REDUCED CHI.SQ. = 1.2585E-01
FINISHED ITERATION # 6 WITH REDUCED CHI.SQ. = 1.2322E-01

ITERATION STOPPED BECAUSE |XCHI| < 0.01

THERE WERE 6 ITERATIONS

USING

\[ Y = A \exp(-Bx) \cos(Cx+D) + E \]

THE FINAL COEFFICIENTS ARE

\[
\begin{align*}
A(1) &= 7.877684 \times 10^1 \\
A(2) &= 1.140880 \times 10^{-3} \\
A(3) &= 9.244445 \times 10^{-2} \\
A(4) &= -3.744320 \times 10^0 \\
A(5) &= 1.343607 \times 10^2 \\
\end{align*}
\]

WITH REDUCED CHI. SQUARE = 1.232234E-01
DO YOU WANT A DATA REVIEW? 1=YES, 0=NO
? 0

DO YOU WANT TO PLOT DATA? 1=YES, 0=NO
? 1

INPUT TITLE OF Y - AXIS
? amplitude (cm.)

INPUT TITLE OF X - AXIS
? time (sec.)

WHICH TYPE OF GRAPH DO YOU WANT?

1 - LINEAR
2 - SEMI-LOG
3 - LOG-LOG

INPUT THE NUMBER OF YOUR SELECTION?
? 1

DO YOU WANT SPECIAL SYMBOLS TO DENOTE DATA POINTS?
1=YES, 0=NO
? 1

SYMBOLS ARE:

1 - CIRCLE
2 - CROSS
3 - TRINAGLE
4 - SQUARE
5 - STAR
6 - DIAMOND
7 - VERTICAL BAR
8 - + SYMBOL
9 - UP ARROW BELOW POINT
10 - DOWN ARROW BELOW POINT
11 - REVERSE TRINAGLE

INPUT THE NUMBER MATCHING YOUR SELECTION?
? 1

DO YOU WANT TO SET THE X RANGE?
1=YES, 0=NO
? 0

DO YOU WANT TO SET THE Y RANGE?
1=YES, 0=NO
? 1

INPUT YMIN, YMAX?
? 0.0, 250.0
Appendix C
Sample Case 2: Application in Fluid Mechanics

Appendix C is the complete listing of an interactive session for a fitting function of nine parameters.
NONLINEAR CURVE-FITTING CODE

NUMBER OF DATA PAIRS = 22

CHOOSEN FITTING FUNCTION IS:

\[ Y = A \times \exp(-0.5((X-B)/C)^2) + D \times \exp(-0.5((X-E)/F)^2) + G + H \times X + I \times X^2 \]

ENTER INITIAL GUESSES FOR THE A1-->A9 PARAMETERS

FOR A(1) ENTER GUESS?
? -1.0
FOR A(2) ENTER GUESS?
? 0.0
FOR A(3) ENTER GUESS?
? 3.5
FOR A(4) ENTER GUESS?
? 0.5
FOR A(5) ENTER GUESS?
? 6.0
FOR A(6) ENTER GUESS?
? 3.0
FOR A(7) ENTER GUESS?
? -1.0
FOR A(8) ENTER GUESS?
? -0.5
FOR A(9) ENTER GUESS?
? -0.1

STARTING VALUES

\[
\begin{align*}
A(1) &= -1.000000E+00 \\
A(2) &= 0.000000E+00 \\
A(3) &= 3.500000E+00 \\
A(4) &= 5.000000E-01 \\
A(5) &= 6.000000E+00 \\
A(6) &= 3.000000E+00 \\
A(7) &= -1.000000E+00 \\
A(8) &= -5.000000E-01 \\
A(9) &= -1.000000E-01 \\
\end{align*}
\]

FINISHED ITERATION # 1 WITH REDUCED CHI.SQ. = 9.9716E+03
FINISHED ITERATION # 2 WITH REDUCED CHI.SQ. = 1.6085E+00
FINISHED ITERATION # 3 WITH REDUCED CHI.SQ. = 7.0243E-01
FINISHED ITERATION # 4 WITH REDUCED CHI.SQ. = 6.9253E-01

ITERATION STOPPED BECAUSE ABS(XCHI).LT.0.01
THERE WERE 4 ITERATIONS

USING

\[ Y = A \times \exp(-0.5 \times ((X-B)/C)^2) + D \times \exp(-0.5 \times ((X-E)/F)^2) + G + H \times X + I \times X^2 \]

THE FINAL COEFFICIENTS ARE

\[
\begin{align*}
A(1) &= -6.583788E-02 \\
A(2) &= 0.000000E+00 \\
A(3) &= 3.431341E+00 \\
A(4) &= 2.952222E-02 \\
A(5) &= 5.160242E+00 \\
A(6) &= 2.575265E+00 \\
A(7) &= -2.212542E-02 \\
A(8) &= -4.082130E-03 \\
A(9) &= -2.473477E-05
\end{align*}
\]

WITH REDUCED CHI. SQUARE = 6.925316E-01

DO YOU WANT A DATA REVIEW? (1=YES, 0=NO).

\[
\begin{array}{cccc}
\text{X-DATA} & \text{Y-DATA} & \text{YFIT} & \% \text{DIFFR.} \\
-3.000000E+01 & 6.766000E-02 & 7.807718E-02 & -1.539637E+01 \\
-2.201900E+01 & 5.808000E-02 & 5.576668E-02 & 3.982988E+00 \\
-1.901710E+01 & 5.212000E-02 & 4.655950E-02 & 1.066864E+01 \\
-1.601690E+01 & 4.021000E-02 & 3.691094E-02 & 8.204583E+00 \\
-1.301570E+01 & 2.678000E-02 & 2.676663E-02 & 6.995715E+00 \\
-1.001340E+01 & 1.427000E-02 & 1.533882E-02 & -7.489994E+00 \\
-8.015800E+00 & 5.510000E-03 & 4.706874E-03 & 1.457580E+01 \\
-6.005700E+00 & -1.124000E-02 & -1.273132E-02 & -1.326794E+01 \\
-5.006200E+00 & -2.505000E-02 & -2.500952E-02 & 1.616062E-01 \\
-4.005900E+00 & -4.011000E-02 & -3.942317E-02 & 1.712360E+00 \\
-3.006100E+00 & -5.841000E-02 & -5.473962E-02 & 6.283815E+00 \\
-2.005800E+00 & -7.123000E-02 & -6.832002E-02 & 3.242984E+00 \\
-1.006000E+00 & -8.048000E-02 & -7.943255E-02 & 1.301504E+00 \\
-7.900000E-03 & -8.273000E-02 & -8.398955E-02 & -1.522850E+00 \\
9.941000E-01 & -7.394000E-02 & -8.136283E-02 & -2.356062E+00 \\
1.395400E+00 & -7.122000E-02 & -7.209165E-02 & -1.223888E+00 \\
2.994800E+00 & -6.222000E-02 & -5.882644E-02 & 5.454124E+00 \\
4.995200E+00 & -4.135000E-02 & -3.649070E-02 & 1.175163E+01 \\
6.995400E+00 & -3.289000E-02 & -3.723041E-02 & -1.319675E+01 \\
8.996700E+00 & -4.212000E-02 & -5.323740E-02 & -2.639458E+01 \\
1.098690E+01 & -7.369000E-02 & -6.806898E-02 & 7.627933E+00 \\
1.299570E+01 & -7.030000E-02 & -7.911515E-02 & -1.158695E+01
\end{array}
\]

MEAN OF % ERROR = -0.475987
DO YOU WANT TO PLOT DATA? (1=YES, 0=NO)

? 1

INPUT TITLE OF Y - AXIS
? pressure coeff.

INPUT TITLE OF X - AXIS
? distance (in.)

WHICH TYPE OF GRAPH DO YOU WANT?

1 - LINEAR
2 - SEMI-LOG
3 - LOG-LOG

INPUT THE NUMBER OF YOUR SELECTION?

? 1

DO YOU WANT SPECIAL SYMBOLS TO DENOTE DATA POINTS
(1=YES, 0=NO)?

? 1

SYMBOLS ARE:

1 - CIRCLE
2 - CROSS
3 - TRIANGLE
4 - SQUARE
5 - STAR
6 - DIAMOND
7 - VERTICAL BAR
8 - + SYMBOL
9 - UP ARROW BELOW POINT
10 - DOWN ARROW BELOW POINT
11 - REVERSE TRIANGLE

INPUT THE NUMBER MATCHING YOUR SELECTION?

? 1

DO YOU WANT TO SET THE X RANGE
(1=YES, 0=NO)?

? 0

DO YOU WANT TO SET THE Y RANGE
(1=YES, 0=NO)?

? 0
References


Interactive Application of Quadratic Expansion of Chi-Square Statistic to Nonlinear Curve Fitting

F. F. Badavi and Joel L. Everhart

NASA Langley Research Center
Hampton, VA 23665-5225

This report contains a detailed theoretical description of an all-purpose, interactive curve-fitting routine that is based on P. R. Bevington's description of the quadratic expansion of the $\chi^2$ statistic. The method is implemented in the associated interactive, graphics-based computer program. The Taylor's expansion of $\chi^2$ is first introduced, and justifications for retaining only the first term are presented. From the expansion, a set of $n$ simultaneous linear equations are derived, which are solved by matrix algebra. A brief description of the code is presented along with a limited number of changes that are required to customize the program for a particular task. To evaluate the performance of the method and the goodness of nonlinear curve fitting, two typical engineering problems are examined and the graphical and tabular output of each is discussed. A complete listing of the entire package is included as an appendix.