NISTIR 88-3803

Logistic Function Data Analysis Program LOGIT

William H. Kirchhoff

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(Formerly National Bureau of Standards)
Center for Chemical Technology
Chemical Thermodynamics Division
Gaithersburg, MD 20899

March 1989

Prepared for
U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

NISTIR 88-3803

Logistic Function Data Analysis Program LOGIT

William H. Kirchhoff

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(Formerly National Bureau of Standards)
Center for Chemical Technology
Chemical Thermodynamics Division
Gaithersburg, MD 20899

March 1989

National Bureau of Standards became the National Institute of Standards and Technology on August 23, 1988, when the Omnibus Trade and Competitiveness Act was signed. NIST retains all NBS functions. Its new programs will encourage improved use of technology by U.S. industry.

Prepared for
U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

LOGISTIC FUNCTION DATA ANALYSIS PROGRAM

LOGIT

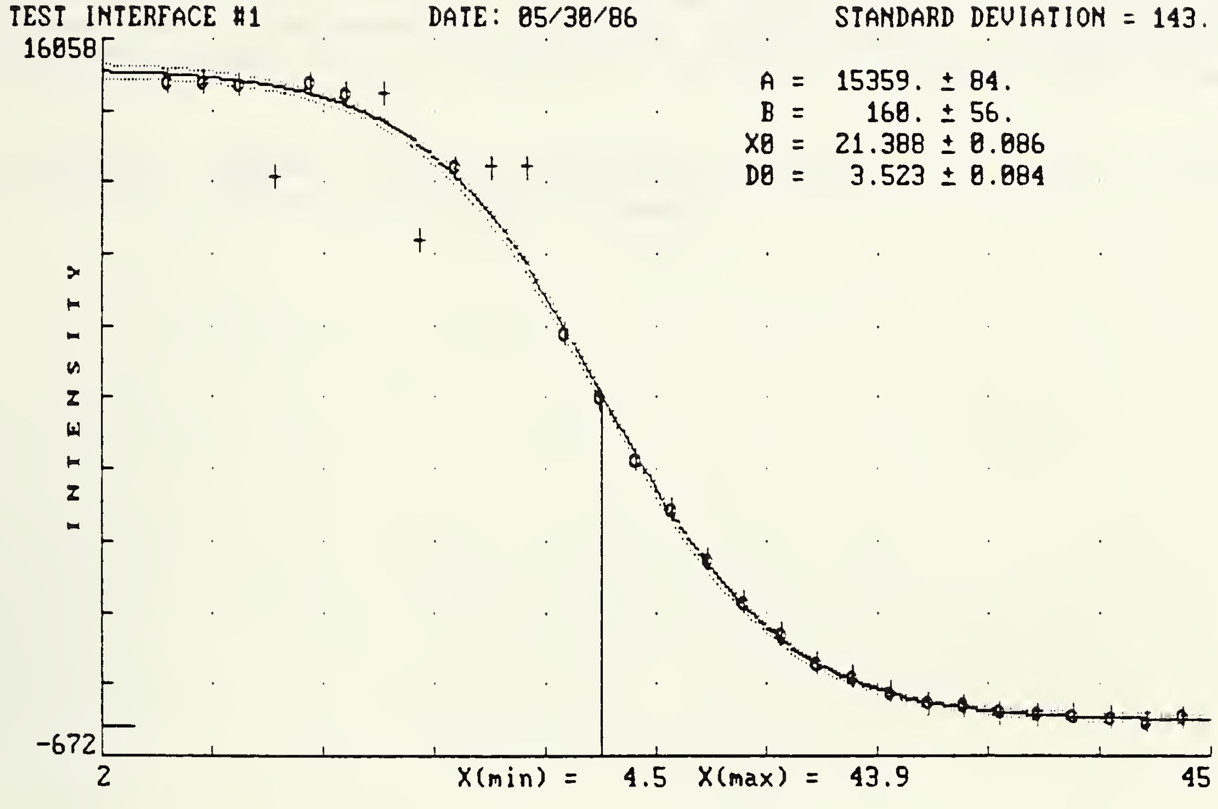
William H. Kirchhoff

Chemical Thermodynamics Division
Center for Chemical Technology
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899

Present address:

Chemical Sciences Division
Office of Basic Energy Sciences
U.S. Department of Energy
Washington, DC 20545

LOGIT



TEST INTERFACE #1
DATE: 05/30/86
STANDARD DEVIATION = 143.
16058
A = 15359. ± 84.
B = 160. ± 56.
X0 = 21.388 ± 0.086
D0 = 3.523 ± 0.084
INTENSITY
-672
2
X(min) = 4.5 X(max) = 43.9
45

ABSTRACT

A FORTRAN program has been written for the statistical analysis of experimental data in terms of an extended logistic function that includes non-horizontal asymptotes and asymmetry in the pre- and post-transition portions of growth and decay curves. The program is robust in that situations in which few or no data fall within the transition interval can be analyzed by the program. Individual weighting of the data is allowed for situations where the errors in experimental data are not uniform. The primary parameters describing the transition region include a location parameter, X_o, a width parameter, D_o, and an asymmetry parameter Q. Six more parameters describe the two quadratic asymptotic regions. The program does not require initial estimates for the parameters and provides statistical estimates of all parameters. A provision is included for the identification and exclusion of outliers. Sufficient information is given to allow the development of companion subroutines for the graphing of the function and its standard deviations as well as for displaying the original data with error bars. The purpose of the program is to provide a means for systematically parameterizing sigmoidal profiles for the comparison of measurements made with different instruments on different systems and for the comparison of measurements with simulation models. The program is extensively documented.

KEY WORDS: logistic function, hyperbolic function, autocatalytic function, statistical analysis, sigmoid

<u>LOGISTIC FUNCTION DATA ANALYSIS PROGRAM</u>

1.0 INTRODUCTION

In its simplest form, the logistic function may be written as:

$$Y = \frac{1}{1 + e^{-X}} \tag{1}$$

in which Y progresses from 0 to 1 as X varies from $-\infty$ to $+\infty$. The differential equation generating this function is:

$$dY/dX = Y(1-Y) \tag{2}$$

and in this form describes a situation where a measurable quantity Y grows in proportion to Y and in proportion to finite resources required by Y. The logistic function was first named and applied to population growth in the last century by Verhulst.[1] It has been applied to the analysis of phenomena in fields as diverse as demography,[2] economics,[3] bioassay,[4] and the physical and chemical sciences where it arises deterministically and/or heuristically in the description of autocatalysis, bimolecular reactions, oxidation-reduction potentials, enzymatic catalysis,[5] biomolecular phase transitions,[6] and depth profile analysis.[7] Numerous descriptions have been presented for the parametric fitting of logistic growth curves with an iterative approach using a truncated Taylor's series expansion providing the most generally satisfactory result.[3] The logistic function as a distribution function and growth curve has been extensively reviewed by Johnson and Kotz.[8]

The program described herein analyzes experimental data in terms of an extended logistic function:

$$Y = \frac{A + A_s(X-X_o) + A_q(X-X_o)^2}{1 + e^{z}} + \frac{B + B_s(X-X_o) + B_q(X-X_o)^2}{1 + e^{-z}} \tag{3a}$$

where $z = (X-X_o)/D$ (3b)

and $D = 2D_o/[1+e^{Q(X-X_o)}]$ (3c)

Y is typically some form of system response or a function of system response and X is the independent variable, such as time, temperature, or concentration. This equation can be fit to both growth and decay curves. Equations 3a - 3c are referred to collectively as an extended logistic function because they include non-linear asymptotes and asymmetry in the curvature of the pre- and post-transition regions.

The analysis program allows the user to determine which parameters are to be varied in the least squares fit of data to Eq. 3. The program can be directed to identify outliers and to exclude these outliers from the fit while, at the same time, printing statistical information for the outliers. The program is designed to calculate initial estimates of the parameters when the parameters are not being held fixed at values provided by the user.

The program consists of a main program and 14 subroutines, written in Fortran 77 and heavily documented, that can be tailored to a variety of computational environments. However, emphasis has been placed on a version that will run on IBM XT, AT, and compatible personal computers possessing a mathematics coprocessor (Intel 8087 and 80287). Portability has thus been sacrificed to a certain extent in favor of user convenience by use of program - compiler - machine dependent graphics subroutines. However, since many graphics packages contain similar commands, the inconvenience of program modification may not be too serious and is offset by the convenience of graphical editing of data. Memory usage depends upon the particular operating environment (such as the Fortran compiler, the size of the Fortran function libraries, the length of representations of floating point numbers, etc.). Typically, 250K of memory should be available for the program and its data requirements.

The purpose of this document is to guide the user in adapting the programs to the user's computing environment and interpreting the results. A rudimentary knowledge of error analysis and the significance of least squares fits are necessary in order to use the program.

1.1 DESCRIPTION OF TERMS

Throughout this documentation, the regions of the sigmoidal profile will be referred to as the pre-transition, transition, and post-transition regions. These terms are not dependent on whether a particular transition profile is a growth or a decay curve though the analysis program must, from time to time, distinguish between the two. The terms pre- and post- are taken in the sense of increasing values of the independent variable X.

The nine parameters necessary to characterize the transition profile are related to the three distinct regions of the transition profile. Three parameters, a slope A_s, an intercept A, and a quadratic term A_q are necessary to define the pre-transition asymptote, three more, B, B_s and B_q, to define the post-transition asymptote, and two more, D_o and X_o, to define the slope and position of the transition region. In addition, an asymmetry parameter Q which causes the width parameter to vary logistically from 0 to $2D_o$, is introduced as a measure of the difference in curvature in the pre- and post-transition ends of the transition region. If $Q < 0$, the pre-transition region has the greatest (sharpest) curvature. If $Q > 0$, the post-transition region has the greatest curvature. If $Q = 0$, $D = D_o$ and the transition profile is symmetric. The parameter Q has the dimensions of 1/X whereas D_o has the dimensions of X. The product QD_o is dimensionless and is a measure of the asymmetry of the profile independent of its width. If the absolute magnitude of QD_o is less than 0.1, the asymmetry in the transition profile should be barely discernible.

From time to time, it will be necessary to distinguish between the two asymptotic limits of Y in terms of their relative values. In these situations, they will be referred to accordingly as the high and low asymptotes. The high asymptote may be either the pre- or post-transition asymptote depending on whether a decay or a growth profile is being analyzed.

Mention is made of poorly structured data. Examples of poorly structured data include, very narrow transition intervals or transition intervals containing few or no data, data with noise comparable in magnitude to the slope of the transition interval, and asymptotes with slopes large compared with the change in Y through the transition interval.

2.0 DESCRIPTION OF THE ANALYSIS

Data in the form of X,Y pairs and saved in a text file are fit by the method of least squares to Equation (3). Because these equations are non-linear functions of the three transition region parameters, X_o, D_o, and Q, the least squares fit requires an iterative solution. Consequently, Y, as expressed by Eq. (3), is expanded in a Taylor series about the current values of the parameters and the Taylor series is terminated after the first (that is, linear) term for each parameter. Y(obs) - Y(calc) is fit to this linear expression and the least squares routine returns the corrections to the parameters. The parameters are updated and the procedure is repeated until the corrections to the parameters are deemed to be insignificant compared to their standard deviations.

2.1 Initial Estimates

The efficiency of the iterative procedure is sensitive to the initial estimates of the values of the parameters. Various means of calculating initial estimates have been tested, and the approaches selected, though they did not always give the best set of initial parameters, were the least prone to false starts in situations of poorly structured data. Three methods are used. The first method calculates the initial estimate of A as the average of the first three values of Y and the initial estimate of B as the average of the last three values of Y. The average value of Y, $\langle Y \rangle = (A+B)/2$, is calculated from the initial estimates of A and B and is taken to be the midpoint of the transition interval. The four data with values closest to $\langle Y \rangle$ are used to calculate a straight line. This line is assumed to be a line tangent to the logistic curve at its midpoint. From the tangent line and from the estimates of A and B, the transition parameters X_o and D_o are calculated. The remaining parameters are given initial values of zero.

In the second method, the initial estimates of A and B are obtained in the same way as the first method, but the largest change in Y is taken as spanning the midpoint of the transition interval. The tangent line is drawn through this interval from which X_o and D_o are calculated.

In the third method, the three regions are marked in a graphical presentation of the data. Straight lines are drawn through the data in each of the regions giving A, A_s, X_o, D_o, B, and B_s.

The third method is used only when the pretransition, transition, and posttransition regions are marked by the user. The second method is used only when the first method fails to converge in the requisite number of iterations.

2.2 The Least Squares Analysis.

A cycle of up to p iterations is executed in which, at the end of each iteration, the parameters are updated before the next iteration is performed. The number of iterations p is chosen on the basis of experience with particular classes of data. If p is selected to be a prime number, oscillations between two or three local minima can be identified by performing repeated multiples of p iterations. Generally, if convergence takes longer than 11 iterations, the solution is unstable in the sense that all of the parameters cannot be determined from the data. In most cases, instability of the fit can be interpreted by the program and the source of the instability removed by varying one fewer parameter in the least squares fit. Messages keep the user informed of these situations.

All measurement errors are assumed to be normally distributed in the Y's. The X's are assumed to be error-free. More precisely, the errors in X are assumed to be sufficiently small that the spread in Y corresponding to the error in X is small compared to the error in Y. If this is not the case, uncertainties in X_o and D_o returned by the least squares program will not be reliable.

The least squares fit has been written to allow individual weighting of the data to account for distributions of measurement errors that vary with the values of X or Y. The weights should be proportional to the inverse square of the uncertainty in the value of Y. If all data have the same expected error, unit weights may be used and the standard deviation of the fit will be a measure of the random error in Y. If the values of Y used in the least squares fit are not the primary measurements, but are functions of the primary measurements, then weights must be used and the program must be modified to calculate the weights based on the functional dependence of Y on the measurements. In particular, if z is a measured quantity with uncertainty s(z), the weight of z would be $w(z) = 1/s(z)^2$. If Y is a function of the measurements z, the weight of Y in the least squares fit, which should be proportional to the inverse square of the uncertainty in Y, would be given by:

$$W(Y) = w(z)/[dY/dz]^{-2} \tag{4}$$

In graphical displays of the results of this analysis, error bars should be presented for each of the experimental points and confidence limits should be drawn bracketing the curve calculated from the least squares parameters. Confidence limits can be calculated using the parameters obtained from the least squares fit as will be explained below. Confidence limits for experimental points are simply $tS/\sqrt{W_i}$ where W_i is the weight given to the ith datum in the fit, S is the standard deviation of the fit, and t is obtained from the desired percentile values of the student's t distribution.[9] (For example, for an infinite number of degrees of freedom and for a 95% confidence interval symmetrically bracketing the data, the value of t is 1.96.)

The confidence limits for the calculated values of Y are determined from the variance-covariance matrix, V, returned by the linear least squares subroutine. More generally, uncertainties in all functions of the parameters, including the calculated values of Y, are obtained from the variance-covariance matrix. The variance-covariance matrix contains all the information about the quality of the fit and the correlation of errors among the parameters. In particular, if F is any function of the parameters C_j (=A, B, ...), the variance (square of the standard deviation) of F is given by

$$s^2(F) = \sum_{j,k} (dF/dC_j)V_{jk}(dF/dC_k). \tag{5}$$

The square root of the variance-covariance matrix diagonals are the standard deviations of each of the determined parameters (as can be seen by setting F = C_j in eq. 5). Equation (5) can be derived directly from the least squares equations and from propagation of error formula:

$$s^2(F) = \sum_{i=1}^{n} (dF/dy_i)^2 s^2(y_i) \tag{6}$$

where yi are all of the observations included in the fit. The least squares equations in matrix form are

$$\mathbf{X}^T\mathbf{X}\mathbf{C} = \mathbf{X}^T\mathbf{Y} \tag{7}$$

where **C** is the column vector of the coefficients, **Y** is the column vector of the dependent variables and **X** is the m x n matrix of the m independent variables. The matrix $\mathbf{X}^T\mathbf{X}$ is the least squares matrix. The variance-covariance matrix is defined, after some manipulation of equations (5)-(7) as

$$\mathbf{V} = s^2(\mathbf{X}^T\mathbf{X})^{-1}. \tag{8}$$

If we define **A** as the unitary transformation that transforms the set of vectors **X** into an orthonormal set so that

$$\mathbf{A}T(\mathbf{X}^T\mathbf{X})\mathbf{A} = 1, \tag{9}$$

which can be rearranged to give

$$(\mathbf{X}^T\mathbf{X})^{-1} = \mathbf{A}\mathbf{A}^T, \tag{10}$$

then

$$\mathbf{V} = s^2\mathbf{A}\mathbf{A}^T \tag{11}$$

This is the procedure followed in Subroutine ORTHO which performs the least squares calculation.

The confidence limits for the logistic curve calculated from the parameters of the least squares fit can be calculated directly from Eqs. (3) and (5).

2.3 SITUATIONS WITH FEW DATA IN THE TRANSITION REGION

The analysis becomes complicated in situations where only a few data fall within the transition region. The parameters A, A_s, A_q, B, B_s, and B_q are determined primarily from the asymptotes of Equation (1), namely:

$$Y = A + A_s(X-X_o) + A_q(X-X_o)^2 \tag{12a}$$

and

$$Y = B + B_s(X-X_o) + B_q(X-X_o)^2 \tag{12b}$$

while X_o, D_o, and Q are determined primarily by data in the transition region.

We define the transition interval as the interval in which Equation (3) differs from the asymptotic limits of equations (12a and 12b) by more than n standard deviations. The number of data falling within the transition interval will determine whether any of X_o, D_o, and Q can be determined. The interval is given approximately by the limits X_m and X_p:

$$X_m = X_o - D\ln[|A-B|/U - 1] \quad (13a)$$

and

$$X_p = X_o + D\ln[|A-B|/U - 1] \quad (13b)$$

The subscripts m and p refer to the pre-transition (minus) and post-transition (plus) sides of the transition interval. D is a function of X_m or X_p (= X in Eq. (3c)). U is the uncertainty in the value of Y at the midpoint of the transition interval and is given by:

$$U = nS/\sqrt{W} \quad (14)$$

where S is the standard deviation of the fit, W is the average weight (see below) of Y in the interval between X_m and X_p and n is usually 2. When Q=0, the interval is symmetrically placed about X_o and D is independent of X. (These expressions are exact only when A_s, A_q, B_s, B_q, and Q are zero. In the analysis program, the value of the interval, $X_p - X_m$ is used only to estimate the upper limit of D_o when D_o cannot be determined by the least squares fit.)

The number of data falling within the transition interval, that is, the number of data for which the measured value Y_i differs from each asymptote (eqs. 12a and 12b) by more than $nS/\sqrt{W_i}$, is counted. If no datum falls within the interval (very sharp transitions), Q is set equal to 0 and X_o and D_o are estimated from the separation between the two measured values of X spanning the transition interval:

$$X_o = (X_{i+1} + X_i)/2 \quad (15a)$$

and

$$D_o = (X_{i+1} - X_i)/(2F_x) \quad (15b)$$

where F_x = the larger of $\ln[|A-B|/U - 1]$ or 2 so that D_o is always $\leq (X_{i+1} - X_i)/4$. All three parameters are held fixed at these values while the other parameters are varied.

If one datum falls within the interval, Q is held fixed at 0 and the maximum value of D_o is estimated from the average data interval among the three points spanning (one just below, one in, and one just above) the transition interval:

$$D_o = [(X_{i+1} - X_{i-1})/2]/F_x \quad (16)$$

The value of X_o and the remaining parameters are determined from the least squares fit, but the value of X_o is influenced very strongly by the single datum falling within the interval.

If two data fall within the transition interval, Q is held fixed at 0, and X_o and D_o are allowed to vary.

If confidence limits are to be assigned to X_o, D_o, or Q based on the standard deviations for these parameters returned by the least squares fit, then care should be taken in interpreting the number of degrees of freedom used in connection with a Student's t distribution. The appropriate number is NIN-2 or NIN-3 depending on whether Q is varied and where NIN is the number of data falling within X_m and X_p. In particular, if only two points fall within the interval, then the uncertainties in X_o and D_o must be estimated from the data separation and not from the standard deviations returned by the fit.

2.4 POST-FITTING TESTS.

Following the cycle of p iterations, four tests are performed to judge the quality of the fit, to test the assumptions of the determinability of X_o, D_o, and Q, and to test the determinability of the asymptotic parameters A_s, A_q, B_s, and B_q. If a test is failed, the analysis is repeated holding certain parameters constant. The user has control, however, over whether the analysis is repeated or not.

The philosophy underlying the performance of the post-fitting tests is that the parameters A_s, A_q, B_s, B_q, and Q are less critical in the analysis of the logistic profile than the parameters D_o and X_o. In general (but not always), the former are of a heuristic nature and have little basis in the theory underlying the use of the logistic function in a particular application.

2.4.1 The Convergence Test.

The most recent correction to the ith parameter, δ_i, is compared to the standard deviation of the ith parameter, σ_i. If $-\mathrm{LOG}_{10}(|\delta_i|/\sigma_i) \sim 1$, the corrections to the parameters are of the same order of magnitude as the standard deviations of the parameters and the iteration cycle is considered to have not converged. If this occurs, the analysis is repeated using the largest jump in the value of Y as an estimate of the midpoint of the transition interval. If the convergence test fails a second time, the parameters A_s, A_q, B_s, and B_q are held fixed at 0 and the data are refit to the remaining parameters. If A_s, A_q, B_s, and B_q had already been held fixed at 0, the asymmetry parameter Q is held fixed at 0. If all five had been held fixed at 0, a lack of convergence message is printed and the calculation continues.

2.4.2 The Interval Test.

Under circumstances of small vertical separation between the two asymptotes compared to the uncertainties in the values of Y, no datum may be found that lies more than two standard deviations away from each asymptote. The program will assume that the transition region is narrow and will estimate a small value for D_o. The assumption of no datum in the transition region is tested by noting how many data overlap the value of Y at the center of the transition interval to within one standard deviation. If one or more such data are found, the assumption of no datum in the transition interval is judged to be false. In general a broad transition profile can be fit reasonably well with a profile consisting of a very narrow interval and asymptotes with pronounced slopes and/or curvatures. The slopes and/or curvatures allow accommodation of Eq. (3) to the broad interval.

Consequently, when the interval test is failed, the data are refit holding A_s, A_q, B_s, and B_q constant at 0. If A_s, A_q, B_s, and B_q had already been held fixed at 0, the data are refit holding Q fixed at 0. If all five had been held fixed at 0, a warning message is printed and the calculation continues.

2.4.3 The Uncertainty Test.

This test, in a sense, is a reverse of the interval test in that it tests the assumption that the interval parameters, X_o, D_o, and Q, determined by the least squares fit, can indeed be determined. The count of the number of data in the transition interval is only an approximation to the number of degrees of freedom available for the determination of the interval parameters. Although n parameters can be obtained from n data, the uncertainties in the resulting parameters cannot be determined. One result of fitting n parameters to n data is large excursions in the uncertainty of the calculated value of Y in the vicinity of the transition interval.

The 2S curves of Y (that is, the values of Y calculated from Eq. (3) ± twice the standard deviation in Y calculated from Eq. (5)) are compared with the 2S curves for the two asymptotes. When the calculated value of Y plus twice its standard deviation is greater than the value of the high asymptote plus twice its standard deviation or when the calculated value of Y minus twice its standard deviation is less than the value of the low asymptote minus twice its standard deviation, the uncertainties in Y in the vicinity of the transition interval are judged to be excessive. When this situation is encountered, the data are refit, beginning with new initial estimates, and holding one more interval parameter constant than in the previous calculation. This test is conducted only when the number of interval parameters varied is equal to NIN, the number of data falling within the transition interval.

2.4.4 The Asymptote Test.

The purpose of the asymptote test is to assure that the data have reached well enough into the asymptotic region that the higher order asymptotic parameters, A_s and A_q or B_s and B_q, can be distinguished from the effects of the interval parameters and can be distinguished from 0. The separation between each value of Y and the values of the asymptotes at the corresponding value of X is compared with the standard deviation of the measured value of Y. The number of data falling within two standard deviations of one asymptote or the other is counted. If more than four data fall within two standard deviations of one asymptote or the other, the parameters defining that asymptote are considered to be determinable. If four or fewer data fall within two standard deviations of one of the asymptotes, that asymptote's parameters, A_s and A_q or B_s and B_q, are compared with their standard deviations. If their values are less than three times their standard deviations, the corresponding parameter is considered to indeterminable from 0.

2.5 OUTLIER IDENTIFICATION AND REJECTION

If directed to do so, LOGIT will, following completion of analysis, identify outliers using criteria provided by the user. If any are identified and if LOGIT is directed to do so, the data will be refit with the outliers dropped from the data being fit. The standardized residuals are used for the identification of the outliers. A standardized residual is the number of standard deviations by which $Y_{obs} - Y_{calc}$ differs from its expected value of zero, that is, the value of $Y_{obs} - Y_{calc}$ divided by the standard deviation of $Y_{obs} - Y_{calc}$. The standard deviation of $Y_{obs} - Y_{calc}$ is given by:

$$s(Y_{obs}-Y_{calc}) = \sqrt{(S_o^2 - S_c^2)} \tag{17}$$

where $S_o^2 = S^2/W$ is the variance (square of the standard deviation) of an observed value of Y, S is the standard deviation returned by the least squares fit, W is the weight of the observation used in the fit, and S_c^2 is the variance of the calculated value of Y and is calculated from variance-covariance matrix using equation (5). The - sign appearing in square root in equation (17) arises from the correlation between Y_{obs} and Y_{calc} (All of the Y_{obs} were used to calculate the parameters which in turn were used to calculate Y_{calc}.) The standardized residuals should follow a student's t distribution and therefore, for extensive data sets, a value of a standardized residual greater in absolute magnitude than 3.0 is generally an outlier.

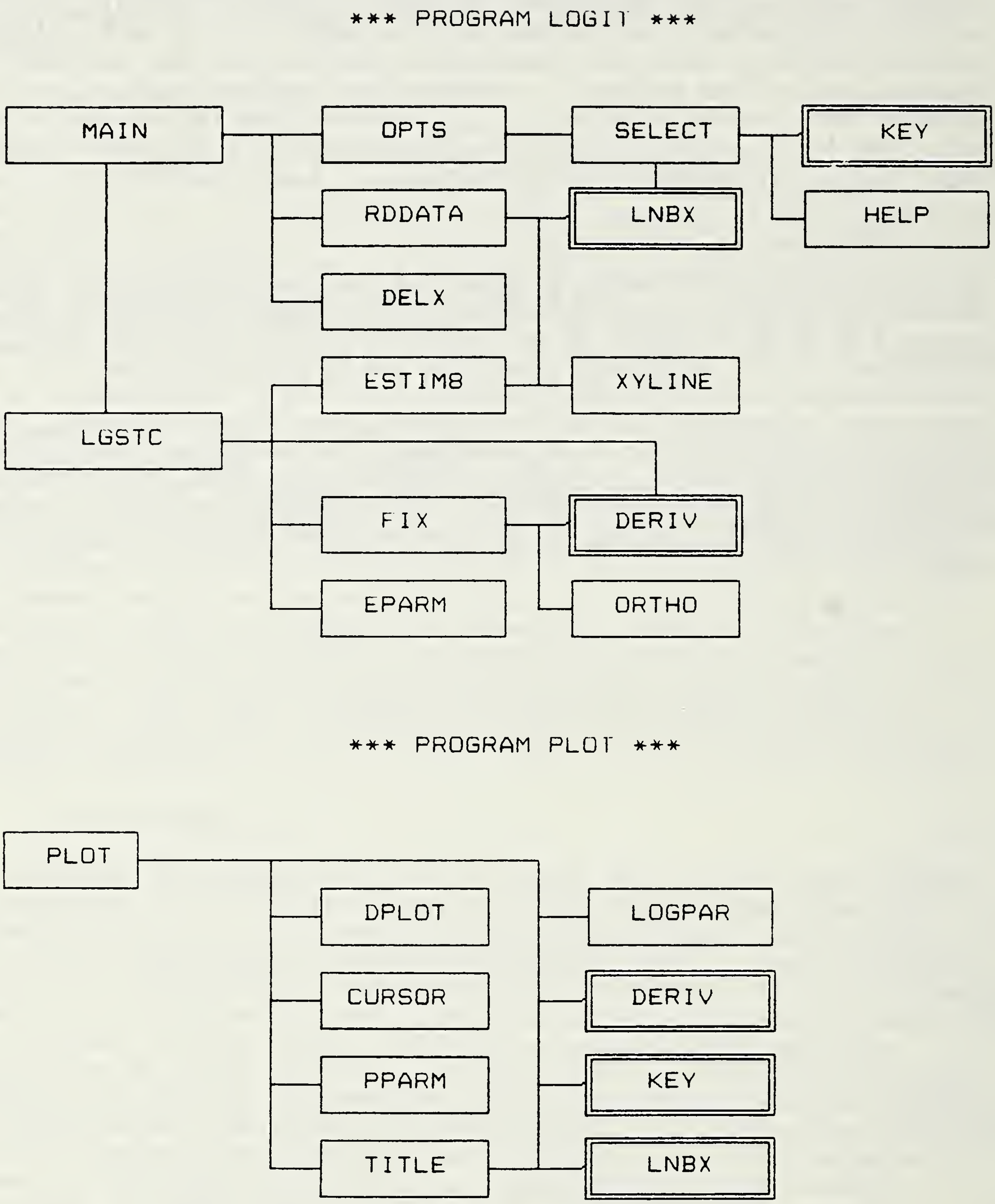


Figure 1 Subroutine structure of LOGIT and PLOT. Subroutines enclosed in double lines are shared by LOGIT and PLOT.

3.0 PROGRAM STRUCTURE AND FLOW

The analysis program, LOGIT, consists of a main program and 14 subroutines, OPTS, SELECT, KEY, HELP, RDDATA, LNBX, DELX, LGSTC, ESTIM8, XYLINE, DERIV, FIX, ORTHO, and EPARM. The plotting program, PLOT, consists of a main program and eight subroutines, three of which are identical to those used by LOGIT. The structures of these programs are represented schematically in Figure 1. Listings of each of these subroutines, which are heavily commented, are included in Appendix A of this documentation. A brief description of each program element follows.

3.1 MAIN

MAIN initiates the program and interactively directs the calculations and output of the results. It saves the data, the parameters and the variance-covariance matrix in an unformatted data file to be used by the plotting program PLOT.

3.2 OPTS

OPTS directs a question and answer session in which the user can select which parameters are to be determined, where the output of the results are to be directed, which post-fitting tests are to be performed, the conditions for outlier identification and rejection, and which data are to be included in the least squares analysis. OPTS is called by MAIN.

3.3 SELECT

SELECT allows the user to select graphically which data are to be included in the fit. It is called by OPTS in response to the user. SELECT is also used to identify the pre-transition, transition, and post-transition regions for those situations when the program is unable to do so. Once identified, these three regions are used by the program to estimate the initial values of the parameters.

3.4 KEY

KEY reads the keyboard to detect if any identified keys have been pressed. It is called by SELECT during graphical editing of the data. (KEY is also a subroutine of PLOT.)

3.5 HELP

HELP prints help messages on the top line of the display during graphical editing of data.

3.6 RDDATA

RDDATA reads data files formatted according to the options offered by OPTS. It performs some preliminary manipulation of the data. For example, it forms the ratio of two columns of data when the ratios rather than the primary data are being analyzed.

3.7 XYLINE

XYLINE calculates the slope and intercept of a line segment and is used for forming initial estimates of the parameters.

3.8 LGSTC

LGSTC is called by the main program and directs the analysis of the data. It is the longest and most complex of the subroutines and is consequently divided into segments. Each segment is identified by the leading statement number in the segment.

3.8.1 SEGMENT 400

Executes a cycle of iterative least squares fits. In each iteration the standard deviation is calculated from the current values of the parameters, the number of data falling within the transition interval is counted and, based on the count, Q, D_o, or X_o may be held fixed. Following the fit, the parameters are updated and a test for convergence is performed. If the most recent corrections to the parameters are less than three orders of magnitude smaller than the corresponding standard deviations and if the interval parameters directed to be varied have not been held fixed, the procedure is deemed to have converged. Otherwise another iteration is performed until the maximum number is reached.

3.8.2 SEGMENT 500

Performs the post-fitting tests:

Convergence test. If the most recent corrections to the parameters are of the same order of magnitude as the standard deviations of the parameters, the test fails.

Test of the number of data in the transition interval. If the number was deduced to be 0 in SEGMENT 400, the number of data whose uncertainties overlap the midpoint of the transition is noted. If not also 0, the test fails.

Test for large uncertainties in Y_{calc} in the vicinity of the transition interval when the number of data falling within the transition interval is equal to the number of interval parameters being varied.

Test for determinability of the asymptotic parameters.

3.8.3 SEGMENT 700

Prints the data and their associated statistics. Checks for outliers and processes them if instructed to do so by the calling program. If outliers are found, the data are refit beginning with the determination of their estimates by Subroutine ESTIM8. All parameters that had been held fixed as a result of a failure of one of the post-fitting tests will again be allowed to vary.

3.9 ESTIM8

ESTIM8 determines the initial values of the parameters by various means as described in section 2.1. ESTIM8 is called by LGSTC.

3.10 DERIV

DERIV is called by FIX and LGSTC and calculates the value of Y from the current values of the parameters for a given value of X. DERIV also calculates the derivative of Y with respect to each of the parameters. These derivatives are used as the independent vectors in the interative least squares fit and are also used for calculating uncertainties in functions of the parameters according to Eq.(5). (DERIV is also called by PLOT.)

3.11 FIX

FIX is called by LGSTC and forms a set of vectors to be sent to ORTHO which performs the least squares fit. The dependent variable is $Y_{obs} - Y_{calc}$ and the independent variables are the derivatives of Y_{calc} with respect to each of the seven parameters. FIX keeps track of which parameters are not being varied in the fit by keeping track of two sets of parameter indices, one corresponding to the parameters in LGSTC and one corresponding to the parameters being fit by ORTHO. FIX also excludes the data that are being excluded from the fit. Following the fit, it readjusts the data returned by ORTHO (the coefficients of the least squares fit, their standard deviations and the variance-covariance matrix) to conform to the indexing of the parameters in LGSTC.

3.12 ORTHO

ORTHO is called by FIX and performs the least squares fit using a Gram-Schmidt orthonormalization procedure on the dependent and independent variables. It is derived from a program originally written by Walsh[10] and has been extensively tested by Wampler[11]. It returns to FIX the corrections to the varied parameters, their standard deviations, and the variance-covariance matrix.

3.13 EPARM

EPARM is called by LGSTC and prints the values of the parameters, their standard deviations, their initial values, and the number of significant figures in the standard deviations of the parameters that remained unchanged in the last iteration performed. Thus, if the standard deviation of a parameter was .123 and the convergence index is given as 3, the most recent correction to the corresponding parameter would be less than .000123. EPARM also prints the standard deviation of the fit, the number of data fit, the number of data falling within the transition interval, the relative deviation (relative to the value of Y at X_o), and the Gram Determinant, the determinant of the least squares matrix X^TX.

3.14 DELX

DELX is called by the main program following the completion of the analysis. DELX calculates the range in X in which the logistic function proceeds from 10% completion of the transition region (transition interval) to 90% completion. DELX also calculates and prints two measures of the asymmetry of the sigmoidal profile. The first of these is the product QD_o. The second is a parameter η given by:

$$\eta = - \frac{(X_{.9} - X_o) + (X_{.1} - X_o)}{(X_{.9} - X_o) - (X_{.1} - X_o)} \tag{18}$$

where $X_{.1}$ is the value of X where the transition is 10% complete and $X_{.9}$ is 90% complete.

3.15 LNBX

LNBX returns the last non-blank character in a string and is used for display purposes in SELECT and RDDATA. (LNBX is also called by the program PLOT).

3.16 PLOT

PLOT is the main program for plotting the results of the analysis in LOGIT. It reads the data generated in an unformatted file by LOGIT and calculates a number of graphs from it including: 1) the original data with vertical 95% confidence bars, the calculated curve, and 95% confidence curves for the calculated curves; 2) the first derivative of the data (taken as ratios of differences between adjacent values of X and Y) and the calculated first derivative; 3) the calculated curve with vertical, 95% confidence bars; 4) a graph normalized to the lower or upper asymptote or to the lowest or highest value of y; and 5) a graph of the standardized residuals. PLOT controls which of these graphs appear on the screen through keystroke commands entered by the user. PLOT also allows "zooming" of plots for examination of narrow regions of data under higher resolution. A cursor routine allows reading of calculated and observed values from the various graphs.

3.17 LOGPAR

LOGPAR forms strings of parameter names and values for presentation on the various graphs.

3.18 DPLOT

DPLOT clears the graphics screen, draws the axes and experimental data with vertical 95% confidence bars for one of the five graphs as directed by PLOT. If the minimum value of X included in the fit is not the first value, DPLOT marks the minimum value on the plot. If the maximum value of X included in the fit is not the last value, DPLOT marks the maximum value on the plot. DPLOT calculates the format for bottom line information displayed on the graph consistent with the magnitude of the y vector being plotted. DPLOT also determines which quadrant of the screen is the emptiest for printing of the parameter values.

3.19 CURSOR

CURSOR places a cursor on the screen following a change in the selection of the graph being displayed. Its main function is to match as closely as possible the cursor position when the principal curve being plotted changes from calculated values to observed values and vice-versa.

3.20 PPARM

PPARM writes the parameter name and value strings into the emptiest quadrant on the screen.

3.21 TITLE

TITLE prints a help message on the top line of the graph and/or a title line consisting of the title appearing in the first line of the original data file, the date on which the analysis of the data file was performed, and the standard deviation of the fit.

4.0 DATA REQUIREMENTS

LOGIT operates on ASCII text files created by the user. The data is in the form of X,Y pairs where X is the independent variable and Y is the dependent variable. The data must be in order of increasing X in order for the initial estimates to be made correctly. The data should also contain at least five points falling within each of the asymptotic limits if the asymptotic parameters are to be determined. The first line of the data file may contain a title up to 30 characters in length which allows for clearer identification than the 8 character file name can convey. If a title line is not given, its absence is recognized by the subroutine RDDATA through absence of any letters in the first line of the file. Alternatively, *if the title line contains no letters, it will be read as data!*

Currently three types of data files can be interpreted. The first consists of X,Y pairs, one value of X and one value of Y per line. The second consists of one value of X per line, but several values of Y. Each value of Y is associated with the same value of X. LOGIT will fit any column of the Y's or any ratio of two columns. For the ratios, appropriate weighting is used assuming unit weights for each of the individual values of Y. In order to distinguish which ratio is being analyzed, a second title line is <u>required</u>. This second line contains a short (up to 8 characters) title for each value of Y. LOGIT determines the number of Y values by counting the number of titles in the second line of the data file. The titles are separated by spaces (any number of spaces). The third type of data file contains for each value of X a value for Y and a weight, W.

5.0 MISCELLANEOUS CONSIDERATIONS

The current versions of LOGIT and PLOT have been written for IBM XT, AT, and compatible computers possessing the mathematics coprocessors. It may be possible to compile the programs and their subroutines with a compiler that does not require the mathematics coprocessors. However, the graphics routines will not be realizable. The only departure from Fortran 77 is in the printing of the iteration number during the least squares fit. The iteration numbers are preceded by an & which causes the line being printed to be appended to the last line printed.

If the program is unable to fit poorly structured data, numerical overflow or underflow may result. The error messages will not give much of a clue on how to respond. Generally, one should try again holding more parameters constant or allowing more parameters to vary. Occasionally, the error message "no file handles left" may appear. In this case, the number of files that can be simultaneously open must be increased. This is done in the CONFIG.SYS file as described in the DOS manuals.

The graphics programs should work with most displays but there may be variations in performance depending on the display adapter. The graphics displays are not in color, but will work with color displays. With some graphics printers, the graphics displays may be printed by pressing the F10 or Shift-F10 keys. If the printer is incompatible with the graphics subroutines, a message to this effect is printed at the top of the screen.

6.0 DESCRIPTION OF AN ANALYSIS SESSION

In the following sections, an analysis session is presented as a tutorial. Section 6.1 deals with the configuration session in which options for the analysis are selected by the user in response to a series of questions. Section 6.2 describes the final results provided by the program. Section 6.3 describes the intermediate results. Section 6.4 describes the graphical editing in LOGIT and section 6.5 the graphical display in PLOT. In sections 6.1-6.3 the response from the program is presented in the left hand column and comments on the interpretation of the response are given to the right. LOGIT and PLOT are run by entering LOGIT and PLOT from the keyboard. The user is prompted for additional information as the program progresses.

6.1 THE SELECTION OF OPTIONS

After typing LOGIT, the user enters into an interactive configuration session which allows the selection of:

- which data files are to be analyzed
- where the results of the analysis are to be stored or displayed
- which parameters are to be varied
- which data are to be included in the analysis
- whether or not outliers are to be excluded from the fit and if so, using what criteria
- which post-fitting tests are to be applied
- whether or not data for a plotting program are to be saved

During the question and answer session, the current state of each option is presented in square brackets such as [Y], [N], [3.00], etc. If a carriage return (ENTER) is entered in response to the question, the option will remain in the state indicated in the square brackets. Thus, only changes need be entered.

In general, it is not necessary to engage in an extensive question and answer session each time LOGIT is run. Any configuration (set of options selected) can be saved either in a default file (DFLT.OPT) or in a special configuration file of the user's choosing (e.g. MYFILE.OPT). At the beginning of a session, the user is given the option of changing any of the options in the default configuration file or of loading a special configuration file.

```
        NATIONAL BUREAU OF STANDARDS
LOGISTIC CURVE FITTING PROGRAM, VERSION 6/10/86
```

Title. Updates can be distinguished by the version date.

```
RECONFIGURE                    - Y/N? [N] Y

                          or

RECONFIGURE                    - Y/N? [N] MYFILE
DO YOU WISH TO ESTABLISH OPTIONS FILE:   MYFILE
```

Several options are available at this point. Any word of one or two letters beginning with a "y" is interpreted as yes. Any word with 3 or more letters (e.g. MYFILE) is interpreted as a special configuration file (bearing the file name MYFILE.OPT). That file is read and options contained in the file become the default options for the current session. If no file of that name exists, the user is prompted with a message asking if an options file of that name is to be formed.

If the reconfiguration is selected by entering either yes or the name of a <u>new</u> options file, the following question and answer session is entered. The first five options of the configuration session are offered <u>only</u> during the configuration session. The remainder of the options may be changed at various times during the course of the analysis.

```
FINAL RESULTS OF ANALYSIS DIRECTED TO:

    [1] CONSOLE
    [2] PRINTER
    [3] RESULTS FILE
    [4] SESSION.LOG FILE
    [5] USER FILE

WHICH? ENTER NUMBER 1 -> 5          [3]
```

The results of the analysis can be directed to the console, the printer, the default files "RESULTS" or "SESSION.LOG" or a file designated by the user. The final results contain the data with associated statistics and the determined parameters with associated statistics.

```
INTERMEDIATE RESULTS OF ANALYSIS DIRECTED TO:

    [1] CONSOLE
    [2] PRINTER
    [3] RESULTS FILE
    [4] SESSION.LOG FILE
    [5] USER FILE

WHICH? ENTER NUMBER 1 -> 5          [1]
```

The intermediate results contain the parameters with associated statistics for each step of the calculation including the identification and dropping of outliers and the failure of post-fitting tests.

Most users will elect to display the intermediate results on the console and print the final results or save them in a results file.

IDENTIFY DATA TYPE:		Three classes of data can be analyzed.
[1] X AND Y [2] SEVERAL Y'S, TAKING RATIOS [3] X, Y, AND WEIGHTS		In the first, values of x and y are analyzed assuming uniform weights.
WHICH? ENTER NUMBER 1 -> 3	[1]	In the second, several values of y accompany each value of x. Each set of y values can be analyzed separately, or ratios of the y's can be analyzed. The second line of the data file for this case contains titles for each of the y values. The number of titles determines the number of values of y included in the data set. In the third, values of the weight for each x,y pair are included in the data file. The first line of each data file should contain a title up to 30 characters in length. This title will appear on all output displays. If a title line is missing (determined by the absence of letters in the first line) the title is given the name of the file. All files are assumed to be ASCII text files. Each line contains one value of x and as many values of y or w as are appropriate for each class described above.
LOCK OPTIONS	- Y/N? [N]	If this is answered "y" then "LOCK" is turned on and options will not be offered during the course of the session. The options entered during the reconfiguration or existing in the selected configuration file will be in effect for the remainder of the session. This is useful if the main program is intended to cycle through a list of files.
AUTOSAVE DATA FOR PLOTTING	- Y/N? [Y]	If yes, all of the data necessary for plotting will automatically be saved following the analysis of each data file. If no, the user will be prompted (unless "LOCK" is on) to save or not save plotting data. Note, that LOGIT does not contain plotting routines. Rather it saves a data file (called PLOT.DAT) for a separate plotting program to use.

* * * * *

The above options can be selected only during the reconfiguration session. The following options may be selected at appropriate times during the remainder of the session if "LOCK" is off.

* * * * *

```
SELECT PARAMETERS TO BE VARIED - Y/N? [N]

     ALLOW  A TO VARY          - Y/N? [Y]
     ALLOW  B TO VARY          - Y/N? [N]
          ENTER VALUE FOR  B       [0.0]
     ALLOW X0 TO VARY          - Y/N? [Y]
     ALLOW D0 TO VARY          - Y/N? [Y]
     ALLOW AS TO VARY          - Y/N? [N]
     ALLOW BS TO VARY          - Y/N? [N]
     ALLOW AQ TO VARY          - Y/N? [N]
     ALLOW BQ TO VARY          - Y/N? [N]
     ALLOW  Q TO VARY          - Y/N? [Y]
```

If yes, the user is prompted to select which of the parameters are to be varied in the analysis.

The pretransition asymptote obeys an expression of the form:

$$A + AS(X-X0) + AQ(X-X0)^2$$

The posttransition asymptote obeys an expression of the form:

$$B + BS(X-X0) + BQ(X-X0)^2$$

X0 is the midpoint of the transition where Y=(A+B)/2

D0 is the width parameter for the interval. The slope at X = X0 is (AS+BS)/2 + (B-A)/4D0

Q is a measure of the asymmetry of the interval.

If the answer to the first four parameter questions (A, B, X0, and D0) is no, the user will be prompted to enter the value of the parameter (note default value in square brackets). The parameter will keep this value during the analysis. If the answer to the last five parameter questions (AS, BS, AQ, BQ, and Q) is no, the parameter will be held at the value of 0.0 during the analysis.

Note that the selection of parameters to be varied and their initial values are part of the configuration file.

```
SELECT POST-FITTING TESTS?     - Y/N? [N]

     INCLUDE CONVERGENCE TEST  - Y/N? [Y]
     INCLUDE INTERVAL TEST     - Y/N? [Y]
     INCLUDE UNCERTAINTY TEST  - Y/N? [Y]
     INCLUDE ASYMPTOTE TEST    - Y/N? [Y]
```

If yes, the user is allowed to select which of the four post-fitting tests are to be performed and which are not. Until the user is familiar with the purposes of the post-fitting tests, it is advisable to allow all post-fitting tests to be performed.

IDENTIFY AND REJECT OUTLIERS - Y/N? [N] Y

OUTLIER REJECTION VALUE: [4.00] 3

MAX. NO. OF RETRIES: [0] 6

If yes is entered, the user is prompted for additional information. If a carriage return (or any character other than "n") is entered the current settings for the outlier rejection criteria are neither changed nor printed to the screen.

The outlier rejection value is the number of standard deviations by which Y(obs)-Y(calc) must differ from its expected value of 0 before a particular value of X,Y is considered to be an outlier. In performing this analysis, the uncertainties in both Y(OBS) and Y(CALC) are considered. A number of retries may be necessary to identify all of the outliers if several happen to be grouped together.

If the outlier rejection option is not selected, the value for the identification (but not rejection) of an outlier is 4.00.

ENTER NUMBER OF ITERATIONS/CYCLE: [11]

For nearly all cases encountered so far, 11 has proven to be an adequate number of iterations for the convergence of the least squares procedure. If convergence has not been reached by 11 iterations, the data are generally not sufficient to support the determination of all of the parameters selected for variation. For highly precise data, fewer iterations may be necessary. The parameters are tested for convergence at the end of each iteration. If the most recent correction to each of the parameters is less than 1/1000 the value of its standard deviation, the process is deemed to have converged and no more iterations are performed.

*** FINISHED WITH CONFIGURATION OPTIONS ***

SAVE THE OPTIONS AS DEFAULTS - Y/N? [N]

The options selected may be saved in the default options file (DFLT.OPT) or in the special options file (e.g. MYFILE.OPT) selected by the user at the beginning of the session. If any character other than a "y" is entered, the options are not saved and apply to the current analysis session alone.

ENTER DATA FILE NAME: FILENAME.EXT

Enter the name of the file containing the data to be analyzed. If the file cannot be found or the data cannot be interpreted, the program will terminate with an error message of the following form:

```
UNABLE TO OPEN OR INTERPRET DATA FILE
     DATA FILE NAME: FILENAME.EXT
     DATA TITLE:
     LINE NUMBER:        0
     CHECK DATA FILE FOR PROPER FORMAT
99
```

SELECT OPTIONS - Y/N? [N]	The options offered are a subset of the options offered during reconfiguration (except for the selection of data to be excluded from the analysis which requires a data file to have been read). The options are not offered if "LOCK" is on.
EDIT DATA - Y/N? [N]	If yes, the data will be graphically displayed and, through use of the cursor keys, the delete and insert keys, and several of the function keys, selected data can be tagged to be ignored in the least squares fit. Deleted data will be displayed in the analysis of the data but will not contribute to the determination of the parameters. The display can also be zoomed. In section 6.4, the graphical editing of data is described in more detail.
SELECT PARAMETERS TO BE VARIED - Y/N? [N] SELECT POST-FITTING TESTS? - Y/N? [N] IDENTIFY AND REJECT OUTLIERS - Y/N? [Y] ENTER NUMBER OF ITERATIONS/CYCLE: [11]	These options are identical to those offered in the reconfiguration session. They are offered again in the event the user wishes to depart from the usual default configuration or if the data warrant special treatment.
BEGIN ANALYSIS AT 10:01:29.06 *** PLEASE WAIT *** ITERATION NUMBER: 1 2 3 4 5 END ANALYSIS AT 10:01:47.96	Depending of where the intermediate and final results are directed, the results of the analysis may appear on the screen. If both are directed to disk files, only the begin and end messages appear on the screen. Depending on the data, the wait can vary from 5 seconds to several minutes. Printing of the intermediate results to the screen can serve as an assurance that the program is still working productively. The count of iterations is also displayed to assure the user that the program is still functioning.
REFIT - Y/N? [N] REINITIALIZE PARAMETERS - Y/N? [N]	This option is presented if "LOCK" is off and allows the user to refit the data selecting different analysis options. If yes, the user is also asked whether the parameters are to be reinitialized. If not, the next round of iterations begins with the current values of the parameters.
SELECT OPTIONS - Y/N? [N]	If the refit option is elected, the user is given the opportunity to reset some of the options.
BEGIN ANALYSIS AT 10:01:59.82 *** PLEASE WAIT *** ITERATION NUMBER: 1 2 3 4 5 END ANALYSIS AT 10:02:18.66 REFIT - Y/N? [N]	

PLOT THE RESULTS	- Y/N? [N]	If the option of automatically saving plotting data was not selected in the configuration file, the option is offered for the current data file. Note that the data are not plotted by LOGIT. Rather, the data necessary for the plotting program are saved. If more than one data file is analyzed, the plotting data from the analysis of each data file are saved in succession.
END OF ANALYSIS FOR DATA FILE: TESTDATA.002 ENTER ANOTHER FILE	 - Y/N? [N]	At this point, the user is given the option of analyzing the data in another file. If yes, the user will be given all of the options available for earlier data files. Also, if yes and no entry is made when the user is prompted to enter a file name, the previous file is assumed. Thus, for data for which ratios are being analyzed, different ratios may be analyzed by requesting that another file be analyzed without entering the name of that file.
FINAL RESULTS CAN BE FOUND IN THE FILE: RESULTS INTERMEDIATE RESULTS CAN BE FOUND IN THE FILE: SESSION.LOG		When all the desired data files have been analyzed, the program ends with a series of messages telling the user where the results can be found if they were written to disk files. The final "1986" indicates that the program exited properly.

```
        NATIONAL BUREAU OF STANDARDS
LOGISTIC CURVE FITTING PROGRAM, VERSION 6/10/86

1986
```

6.2 THE FINAL RESULTS

The final results include the original data; the calculated values of Y and associated statistics; the values of the determined parameters and their uncertainties; and statistics related to the overall quality of the least squares fit. Below, the final results are presented for the analysis of the data in the file TESTDATA.001. The data in this file contain five outliers grouped near the beginning of the data and do not extend sufficiently into the pre- and post-transition regions to allow the determination of the slopes of the asymptotes. During the analysis, while the outliers are being identified, three of the four post fitting tests are failed at least once. At the start of the analysis, all parameters are directed to be varied, though, as can be seen from the results, the slopes of the pre- and post-transition asymptotes cannot be determined from the data. In all, 14 cycles through the least squares program are required to arrive at a final analysis.

The final results can be printed to the screen, the printer, the "RESULTS" or "SESSION.LOG" file or to a file designated by the user.

NATIONAL BUREAU OF STANDARDS

LOGISTIC CURVE FITTING PROGRAM, VERSION 6/10/86

ANALYSIS OF DATA IN TESTDATA.001

DATE 6/10/86

TIME 18:09:44.51

TEST INTERFACE #1

Program name and version date, title of the file, date and time of the analysis and the title of the data contained in the first record of the data file.

X(OBS)	Y(OBS)	Y(CALC)	DIFF	STANDARD RESIDUALS	
4.52	15011.	15094.	-82.	-1.0	
5.92	15011.	15074.	-62.	-0.7	
7.32	14959.	15035.	-77.	-0.9	
8.72	12800.	14966.	-2165.	-20.1	*
10.12	14990.	14843.	147.	1.7	
11.52	14761.	14637.	124.	1.5	3
12.92	14761.	14302.	458.	4.0	*
14.32	11331.	13788.	-2456.	-20.3	*
15.73	13037.	13034.	3.	0.0	21
17.13	13037.	12016.	1021.	8.3	*
18.53	13037.	10739.	2298.	19.4	*
19.93	9157.	9267.	-110.	-1.4	60
21.33	7642.	7715.	-73.	-0.9	75
22.73	6192.	6214.	-22.	-0.3	63
24.13	5024.	4869.	156.	1.8	51
25.52	3878.	3742.	135.	1.5	39
26.93	2865.	2824.	40.	0.5	29
28.32	2150.	2121.	29.	0.3	22
29.72	1449.	1583.	-133.	-1.5	15
31.13	1123.	1176.	-53.	-0.6	11
32.62	770.	859.	-89.	-1.0	8
34.02	520.	639.	-119.	-1.3	5
35.42	479.	475.	4.	0.0	5
36.82	311.	353.	-41.	-0.4	3
38.30	267.	256.	11.	0.1	3
39.70	232.	187.	45.	0.5	
41.10	167.	135.	32.	0.4	
42.50	89.	95.	-6.	-0.1	
43.90	206.	64.	142.	1.6	

* NOT INCLUDED IN THE FIT

A summary of the data. The column labeled DIFF contains the difference Y(OBS) - Y(CALC). The calculated values of Y were calculated using the parameters in the table that follows the data. The standardized residuals are the number of standard deviations by which the differences, Y(OBS) - Y(CALC), differ from their expected value of 0. The standard deviation of Y(OBS) - Y(CALC) includes the contributions from both the observed and calculated values of Y and include the effects of correlation when a value of Y is included in the fit. The standardized residuals should follow a student's t distribution and values greater than 3.00 are suspect. The numbers appearing to the right of the standardized residuals are the number of standard deviations by which the calculated values of Y differ from their nearest asymptote and, taken together, indicate how well the interval region can be characterized. Those data not included in the fit are flagged with an asterisk. (The values of their standardized residuals - ranging from 4.0 to 20.3 - show clearly that these data are indeed outliers.) Had additional outliers been found, but not excluded from the fit, they would have been flagged with a double asterisk and identified as possible outliers.

The number of significant figures reported for X is determined by the average separation between adjacent X values. The number of significant figures reported for Y is determined by the standard deviation of the fit. A value of Y equal to the standard deviation is reported with two significant figures.

CYCLE 14 ITERATION NO. 5

A =	15113.	+/-	62.	(14994.	3)
B =	-53.	+/-	69.	(154.	3)
X0 =	21.498	+/-	0.059	(21.454	3)
D0 =	3.448	+/-	0.057	(3.751	3)
AS =	0.0			(0.0	*)
BS =	0.0			(0.0	*)
Q =	-0.0315	+/-	0.0066	(0.0	3)

* NOT VARIED IN THE ANALYSIS

As indicated, 14 cycles of iterations were performed. On the last cycle, 5 iterations were required to reach convergence. This large number of cycles was required for the identification of outliers and the response to the failing of various post fitting tests. These are detailed in the discussion of the intermediate results in Section 6.3. Following the values for each parameter are the values of their standard deviations. Following these, in parentheses, are the starting values for the parameters. These starting values, if not provided by the user, are estimated by SUBROUTINE ESTIM8 and the manner of their estimation can be found in the documentation of that subroutine. The second number in each of the parentheses is roughly the number of digits beyond the standard deviation that remained unchanged in the last iteration. Values of 3 or greater indicate adequate convergence. Values of 1 or 2 are suspect. Values of 0 cause the convergence test to fail with subsequent refitting of the data. The parameters AQ and BQ were not directed to be varied and they do not appear in the parameter list. LOGIT determined that the parameters AS and BS could not be evaluated from the data though they were directed by the user to be varied. Hence they appear in the parameter list.

24 DATA POINTS, 16 IN THE INTERVAL

The transition interval is defined in terms of the standard deviation of the fit (the precision in the values of Y) and the magnitude of the separation between the upper and lower asymptote (A-B). It is the interval in which an observed value of y can be expected to fall more than two standard deviations away from either asymptote. The number of data falling within the transition interval is a measure of the number of degrees of freedom for the determination of T0, D0, and Q. If less than three fall within the interval then not all three of T0, D0, and Q can be determined.

STANDARD DEVIATION = 98.5 (98.5)

The number in parentheses represents the standard deviation calculated with the values of the parameters just before the last iteration. Convergence is indicated when these two numbers agree.

RELATIVE DEVIATION = 1.31%

100 x the standard deviation divided by the magnitude of a hypothetical value of Y at the midpoint of the transition interval.

GRAM DETERMINANT = 0.345497E-01	The determinant of the least squares matrix. Although the least squares matrix is never formed in SUBROUTINE ORTHO, the value of the Gram Determinant is a measure of the degree of independence between the various parameters. For data typical of the logistic function, values in the range of 0.1 to 0.0001 are to be expected. Values less than 10^{-n} where n is the number of parameters varied indicate possible problems with broad minima and high correlation among the parameters.
AT X = 14.73 TRANSITION IS 10.0% COMPLETE AT X = 30.09 TRANSITION IS 90.0% COMPLETE RANGE = 15.37 +/- 0.27 ETA = -0.1187 +/- 0.0245 Q*D0 = -0.1086 +/- 0.0226	These values are calculated by SUBROUTINE DELX and are additional measures of the width and asymmetry of the transition interval. If X(f) represents the value of X at which the transition has reached the fraction f of completion and X(1-f) the value at which X has reached a fraction 1-f of completion, the parameter ETA is given by ETA = -[(X(1-f)-X0)+(X(f)-X0)]/[(X(1-f)-X0)-(X(f)-X0)]. For f=0.1 and for low values of Q, the product QD0 is approximately equal to ETA. Both ETA and QD0 are dimensionless measures of the asymmetry of the interval. If Q is held fixed at 0, the values of QD0 and ETA are both zero and their values are not printed.

6.3 THE INTERMEDIATE RESULTS

During the course of the analysis, any of the four post-fitting tests may fail and/or outliers may be identified and dropped from the fit. When these events occur, the values of the parameters are printed and the iterative least squares fit is redone holding certain parameters fixed or dropping outliers from the fit. The intermediate results contain the parameters for each of the intermediate stages as well as the final least squares fit in the same format as they are presented in the final results. In addition, the intermediate results contain messages about post-fitting tests failed and outliers discovered.

The intermediate results can be printed to the screen, to a printer, to the results file, to the session.log file, or to the file designated by the user for the results file. The intermediate and final results can both be printed to the same file or to two different files.

Reproduced below are the intermediate results of the analysis of TESTDATA.001 leading to the final results appearing in the preceding section.

```
        NATIONAL BUREAU OF STANDARDS
LOGISTIC CURVE FITTING PROGRAM, VERSION 6/10/86


START OF SESSION LOG FOR UNIT  2
DATA FILE NAME: TESTDATA.001
DATE 6/10/86
TIME 19:12:33.33

CALL LGSTC AT 19:12:44.10
```

Program title, version number, file name, date of analysis are printed for the session log as they are for the results file. Timing for the analysis is indicated by the times at which LGSTC is called and when LGSTC returns control to the main program.

```
CYCLE  1 ITERATION NO.  11

 A =  12204. +/-   1490. (  14994.  1)
 B =   1858. +/-  15944. (    154.  0)
X0 =    21.7 +/-     7.6 (    21.5  0)
D0 =     1.6 +/-     8.6 (     3.8  0)
AS =   -167. +/-     64. (     0.0  1)
BS =    -86. +/-    665. (     0.0  0)
 Q =   -0.23 +/-    0.13 (     0.0  0)

29 DATA POINTS, 9 IN THE INTERVAL
STANDARD DEVIATION = 764. ( 791.)
RELATIVE DEVIATION =  10.87%
GRAM DETERMINANT   =    0.253846E-05

CONVERGENCE TEST FAILED
```

When the convergence test fails, the program redoes the calculation using a different approach for calculating the initial estimates. In the first approach, the average value of Y is used as the initial estimate of the midpoint of the transition. In the second approach, the largest jump in Y is used as the initial estimate of the midpoint of the transition. Note the last column of numbers in the parameter list. The 0's and 1's indicate that the last corrections to the parameters were comparable to the standard deviations of the parameters.

Note also the lack of agreement of the standard deviation before and after the last iteration, again indicating lack of convergence.

```
CYCLE  2 ITERATION NO.  11

 A =   9436. +/-   3511. (  14994.  0)
 B =   -770. +/-   2156. (    154.  0)
X0 =    24.2 +/-     1.9 (    21.4  0)
D0 =    2.14 +/-    0.67 (    2.59  0)
AS =   -314. +/-    213. (     0.0  0)
BS =     34. +/-    108. (     0.0  0)
 Q =   -0.22 +/-    0.17 (     0.0  0)

29 DATA POINTS, 6 IN THE INTERVAL
STANDARD DEVIATION = 783. ( 810.)
RELATIVE DEVIATION =  18.07%
GRAM DETERMINANT   =    0.255142E-04

CONVERGENCE TEST FAILED
REFIT HOLDING AS AND BS = 0
```

Again, the analysis fails to converge. The next step is to conduct the analysis holding AS and BS = 0. If this too fails, then the analysis will be repeated holding Q=0. If this fails, the analysis ends.

```
CYCLE  3 ITERATION NO.   6

 A =  14558. +/-    392. ( 14994.  3)
 B =    120. +/-    437. (   154.  3)
X0 =   22.04 +/-   0.37 (  21.45  4)
D0 =    3.03 +/-   0.34 (   3.75  3)
AS =     0.0            (    0.0  *)
BS =     0.0            (    0.0  *)
 Q =  -0.027 +/-  0.050 (    0.0  3)

* NOT VARIED IN THE ANALYSIS

29 DATA POINTS, 8 IN THE INTERVAL
STANDARD DEVIATION = 788. ( 788.)
RELATIVE DEVIATION =  10.74%
GRAM DETERMINANT   =   0.665502E-01

THE FOLLOWING HAVE BEEN IDENTIFIED AS OUTLIERS

 N  X(OBS)  Y(OBS)   Y(CALC)    DIFF   STANDARD
                                       RESIDUALS

  8   14.32   11331.   13762.   -2431.  -3.4

REPEAT, EXCLUDING ABOVE DATA FROM ANALYSIS
```

This time the analysis converges in 6 iterations. One outlier is found so the analysis is repeated with the outlier dropped from the least squares fit and with all parameters varied. If no outlier had been found, the analysis would have been complete. Note the agreement between the two estimates of the standard deviation.

```
CYCLE  4 ITERATION NO.  11

 A =   3401. +/-    740. ( 14994.  0)
 B =   1002. +/-    961. (   154.  0)
X0 =   29.02 +/-   0.70 (  21.45  *)
D0 <    0.70            (   3.75  *)
AS =   -571. +/-     51. (    0.0  0)
BS =    -68. +/-    105. (    0.0  0)
 Q =     0.0            (    0.0  *)

* NOT VARIED IN THE ANALYSIS

28 DATA POINTS, 0 IN THE INTERVAL
STANDARD DEVIATION = 1528. ( 1555.)
RELATIVE DEVIATION =  69.43%
GRAM DETERMINANT   =   0.599094E-01

CONVERGENCE TEST FAILED
```

Again the procedure does not converge. The analysis is restarted using the largest jump in Y as the measure of the midpoint of the transition interval. Note that the fit appears to be worse than following the first iteration.

Note that LGSTC concluded that no data fell within the interval and consequently held the values of X0, D0, and Q fixed at predetermined values depending in part on the structure of the data. Also note the large difference between initial and final values of the parameters.

Repeat the fit using the largest jump in Y as the initial estimate of X0.

```
CYCLE  5 ITERATION NO.  11

 A =   3401. +/-    740. (  14994.  0)
 B =   1002. +/-    961. (    154.  0)
XO =   29.02 +/-    0.70 (   21.45  *)
DO <    0.70             (    3.75  *)
AS =   -571. +/-     51. (     0.0  0)
BS =    -68. +/-    105. (     0.0  0)
 Q =     0.0             (     0.0  *)

* NOT VARIED IN THE ANALYSIS

28 DATA POINTS, 0 IN THE INTERVAL
STANDARD DEVIATION = 1528. ( 1555.)
RELATIVE DEVIATION =  69.43%
GRAM DETERMINANT   =    0.599094E-01

CONVERGENCE TEST FAILED
REFIT HOLDING AS AND BS = 0
```

As in the second cycle, the procedure fails to converge and so again AS and BS are held constant.

Note that both procedures for calculating the initial estimates of the parameters gave the same starting values.

```
CYCLE  6 ITERATION NO.   7

 A =  14579. +/-    222. (  14994.  4)
 B =   -176. +/-    379. (    154.  4)
XO =   21.95 +/-    0.27 (   21.45  4)
DO =    2.88 +/-    0.27 (    3.75  3)
AS =     0.0             (     0.0  *)
BS =     0.0             (     0.0  *)
 Q =  -0.093 +/-   0.042 (     0.0  3)

* NOT VARIED IN THE ANALYSIS

28 DATA POINTS, 12 IN THE INTERVAL
STANDARD DEVIATION = 572. ( 572.)
RELATIVE DEVIATION =   7.94%
GRAM DETERMINANT   =    0.581992E-01

THE FOLLOWING HAVE BEEN IDENTIFIED AS OUTLIERS

 N  X(OBS)  Y(OBS)   Y(CALC)    DIFF   STANDARD
                                       RESIDUALS

  4    8.72   12800.   14578.   -1778.  -3.4
  8   14.32   11331.   14321.   -2989.  -4.9

REPEAT, EXCLUDING ABOVE DATA FROM ANALYSIS
```

The procedure converges and a second outlier is discovered. Note that the standardized residual of the first outlier which had been excluded from this fit has increased in magnitude from 3.4 to 4.9. The analysis is now repeated dropping both data from the least squares fit.

```
CYCLE  7 ITERATION NO.   11

 A =   5890. +/-    712. (  14994.  6)
 B =   1845. +/-    786. (    154.  7)
XO =   26.23 +/-    0.70 (   21.45  *)
DO <    0.70             (    3.75  *)
AS =   -526. +/-     56. (     0.0  6)
BS =   -118. +/-     73. (     0.0  7)
 Q =     0.0             (     0.0  *)

* NOT VARIED IN THE ANALYSIS

27 DATA POINTS, 0 IN THE INTERVAL
STANDARD DEVIATION = 1372. ( 1372.)
RELATIVE DEVIATION =  35.46%
GRAM DETERMINANT   =   0.644768E-01

NMP =  3.  INTERVAL TEST FAILED
REFIT HOLDING AS AND BS = 0
```

The fit is repeated allowing all parameters to vary. This time the procedure converges but the analysis has concluded that the transition interval is extremely narrow because no data fall within the transition interval. However a count of the number of data whose standard deviations overlap the midpoint of the transition (NMP = 3) indicate that the interval may not be narrow and that the sloping baselines may be accommodating a broad transition region. Hence, the fit is repeated holding AS and BS = 0.

(Note that when the analysis concludes that no data fall within the transition interval the maximum value for D0 is estimated and the uncertainty in X0 is taken from this value of D0.)

```
CYCLE  8 ITERATION NO.   6

 A =  14928. +/-    186. (  14994.  3)
 B =    -98. +/-    275. (    154.  3)
XO =   21.83 +/-    0.19 (   21.45  4)
DO =    2.97 +/-    0.19 (    3.75  4)
AS =     0.0             (     0.0  *)
BS =     0.0             (     0.0  *)
 Q =  -0.069 +/-   0.029 (     0.0  3)

* NOT VARIED IN THE ANALYSIS

27 DATA POINTS, 13 IN THE INTERVAL
STANDARD DEVIATION = 413. ( 413.)
RELATIVE DEVIATION =   5.57%
GRAM DETERMINANT   =   0.527857E-01

THE FOLLOWING HAVE BEEN IDENTIFIED AS OUTLIERS

 N  X(OBS)  Y(OBS)   Y(CALC)    DIFF   STANDARD
                                       RESIDUALS

  4    8.72   12800.   14921.   -2121.  -4.7
  8   14.32   11331.   14440.   -3109.  -6.8
 11   18.53   13037.   11594.    1443.   4.1

REPEAT, EXCLUDING ABOVE DATA FROM ANALYSIS
```

The procedure converges with a greatly improved value for the standard deviation and relative deviation. Moreover, the interval is no longer deemed to be narrow.

A third outlier is discovered as the magnitudes of the first two outliers identified continue to increase.

```
CYCLE  9 ITERATION NO.    8

 A =  14808. +/-    708. (  14994.  3)
 B =  -3595. +/-   5235. (    154.  4)
XO =    23.3 +/-     2.9 (    21.5  4)
DO =     4.5 +/-     2.3 (     3.8  4)
AS =    -11. +/-     42. (     0.0  3)
BS =    136. +/-    186. (     0.0  4)
 Q =  -0.093 +/-   0.039 (     0.0  4)

26 DATA POINTS, 16 IN THE INTERVAL
STANDARD DEVIATION = 202. ( 202.)
RELATIVE DEVIATION =   3.61%
GRAM DETERMINANT   =   0.201439E-06

NU = 0 & NS6 = 0.  HIGH ASYMPTOTE TEST FAILED
REFIT HOLDING SLOPE(S) = 0
```

The fit is repeated dropping the three outliers, and the parameters AS and BS are again allowed to vary. This time, the asymptote test fails. The asymptote test requires that the value of the slope must be greater than three times its standard deviation if less than five points fall within two standard deviations of the asymptote. If fewer than five data fall within two standard deviations of an asymptote, the magnitude of the slope is compared with its standard deviation. The magnitude of the slope must be at least three times as large as its standard deviation if it is to be judged to be significantly different from zero. When the asymptote test fails, the fit is repeated holding the offending slope at 0.

```
CYCLE 10 ITERATION NO.    4

 A =  15467. +/-    877. (  14994.  3)
 B =    -96. +/-    165. (    154.  3)
XO =   21.43 +/-    0.41 (   21.45  3)
DO =    3.28 +/-    0.13 (    3.75  3)
AS =     29. +/-     59. (     0.0  3)
BS =     0.0             (     0.0  *)
 Q =  -0.045 +/-   0.022 (     0.0  3)

* NOT VARIED IN THE ANALYSIS

26 DATA POINTS, 15 IN THE INTERVAL
STANDARD DEVIATION = 210. ( 210.)
RELATIVE DEVIATION =   2.73%
GRAM DETERMINANT   =   0.507217E-03

THE FOLLOWING HAVE BEEN IDENTIFIED AS OUTLIERS
```

N	X(OBS)	Y(OBS)	Y(CALC)	DIFF	STANDARD RESIDUALS
4	8.72	12800.	15023.	-2223.	-9.4
8	14.32	11331.	14170.	-2839.	-11.9
10	17.13	13037.	12406.	631.	3.9
11	18.53	13037.	11049.	1988.	8.1

```
REPEAT, EXCLUDING ABOVE DATA FROM ANALYSIS
```

A fourth outlier is discovered as the magnitudes of the standard residuals of the previously identified outliers continue to increase. From this point on, the asymptote tests continue to fail and one more outlier is discovered. The analysis is finally complete after 14 cycles. The large number of cycles required for convergence resulted from a combination of exacerbating conditions: the clustering of the outliers, the very large errors in most of these outliers and the inability of the data to support the determination of the slopes of the asymptotes.

```
CYCLE 11 ITERATION NO.   7

 A =  15910. +/-    676. (  14994.  3)
 B =  -2390. +/-   1330. (    154.  4)
XO =   22.18 +/-    0.88 (  21.45  4)
DO =    4.15 +/-    0.41 (   3.75  5)
AS =     54. +/-     43. (    0.0  3)
BS =     94. +/-     50. (    0.0  4)
 Q =  -0.056 +/-   0.020 (    0.0  4)

25 DATA POINTS, 19 IN THE INTERVAL
STANDARD DEVIATION = 93.7 ( 93.7)
RELATIVE DEVIATION =    1.39%
GRAM DETERMINANT   =    0.183880E-06

NL = 4 & NS5 = 1.  LOW ASYMPTOTE TEST FAILED
NU = 0 & NS6 = 1.  HIGH ASYMPTOTE TEST FAILED
REFIT HOLDING SLOPE(S) = 0

CYCLE 12 ITERATION NO.   6

 A =  15107. +/-     75. (  14994.  3)
 B =   -111. +/-     94. (    154.  4)
XO =  21.514 +/-   0.075 ( 21.454  3)
DO =   3.404 +/-   0.074 (  3.751  3)
AS =     0.0             (    0.0  *)
BS =     0.0             (    0.0  *)
 Q = -0.0411 +/-  0.0082 (    0.0  3)

* NOT VARIED IN THE ANALYSIS

25 DATA POINTS, 16 IN THE INTERVAL
STANDARD DEVIATION = 130. ( 130.)
RELATIVE DEVIATION =    1.73%
GRAM DETERMINANT   =    0.354953E-01

THE FOLLOWING HAVE BEEN IDENTIFIED AS OUTLIERS

 N  X(OBS)  Y(OBS)   Y(CALC)    DIFF   STANDARD
                                       RESIDUALS

  4   8.72   12800.   15011.  -2210. -15.6
  7  12.92   14761.   14425.    336.   3.0
  8  14.32   11331.   13927.  -2596. -16.9
 10  17.13   13037.   12132.    905.   5.7
 11  18.53   13037.   10813.   2224.  14.4

REPEAT, EXCLUDING ABOVE DATA FROM ANALYSIS
```

```
CYCLE 13 ITERATION NO.   6

 A =  16049. +/-    620. (  14994.  3)
 B =  -1471. +/-    733. (    154.  3)
XO =   21.70 +/-    0.58 (  21.45  3)
DO =    3.93 +/-    0.17 (   3.75  3)
AS =     62. +/-     38. (    0.0  3)
BS =     61. +/-     28. (    0.0  3)
 Q =  -0.039 +/-   0.016 (    0.0  3)

24 DATA POINTS, 20 IN THE INTERVAL
STANDARD DEVIATION = 64.2 ( 64.2)
RELATIVE DEVIATION =    0.88%
GRAM DETERMINANT   =    0.200482E-06

NL = 3 & NS5 = 1.  LOW ASYMPTOTE TEST FAILED
NU = 0 & NS6 = 2.  HIGH ASYMPTOTE TEST FAILED
REFIT HOLDING SLOPE(S) = 0

CYCLE 14 ITERATION NO.   5

 A =  15113. +/-     62. (  14994.  3)
 B =    -53. +/-     69. (    154.  3)
XO =  21.498 +/-   0.059 ( 21.454  3)
DO =   3.448 +/-   0.057 (  3.751  3)
AS =     0.0             (    0.0  *)
BS =     0.0             (    0.0  *)
 Q = -0.0315 +/-  0.0066 (    0.0  3)

* NOT VARIED IN THE ANALYSIS

24 DATA POINTS, 16 IN THE INTERVAL
STANDARD DEVIATION = 98.5 ( 98.5)
RELATIVE DEVIATION =    1.31%
GRAM DETERMINANT   =    0.345497E-01

THE FOLLOWING HAVE BEEN IDENTIFIED AS OUTLIERS

 N  X(OBS)  Y(OBS)   Y(CALC)    DIFF   STANDARD
                                       RESIDUALS

  4   8.72   12800.   14966.  -2165. -20.1
  7  12.92   14761.   14302.    458.   4.0
  8  14.32   11331.   13788.  -2456. -20.3
 10  17.13   13037.   12016.   1021.   8.3
 11  18.53   13037.   10739.   2298.  19.4

RETURN FROM LGSTC AT 19:15:21.02
```

* * * * *

After all outliers have been identified and the determinable parameters evaluated, the final values of the parameters and the list of outliers are printed. This is the same table as that appearing in RESULTS.

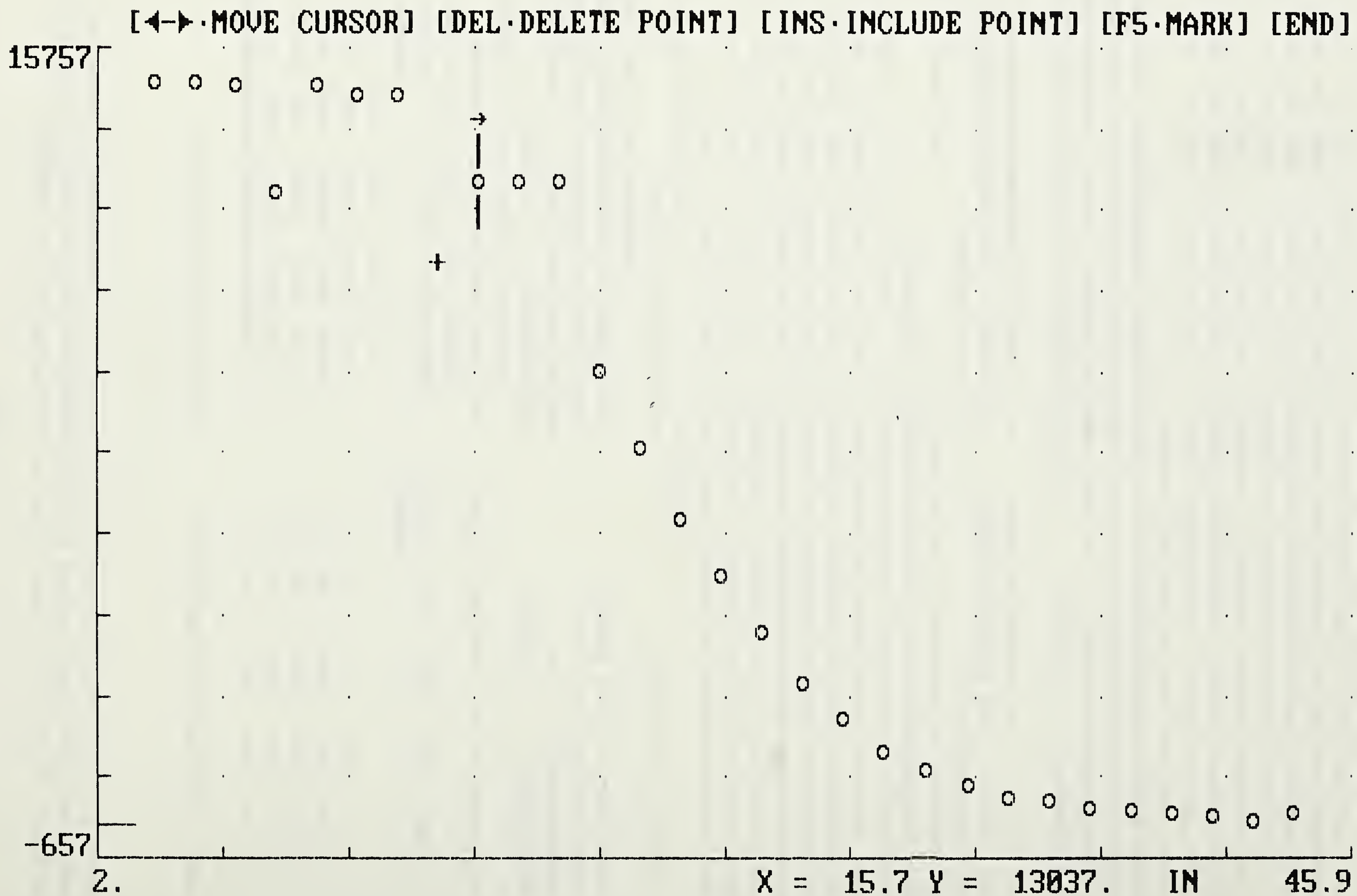


Figure 2. Data Editing Session

6.4 GRAPHICAL EDITING OF DATA

Before analysis, the data may be inspected; obvious outliers may be rejected; and/or blocks of data may be selected for analysis. In addition, the pretransition, transition, and posttransition regions may be marked to aid in the initial estimation of the parameters. If, during the options session, the user enters yes to the prompt: EDIT DATA - Y/N?, the screen will be cleared and the data will be displayed in a manner similar to that shown in Figure 2.

The cursor position is marked with a pair of vertical bars and the direction of cursor movement during deletion or insertion of data is indicated by an arrow above or below the cursor. On the bottom line is given the x and y values of the current cursor position and an indication of whether the datum at the cursor is to be included ("IN") in or excluded ("OUT") from the fit. If the delete key is pressed, the datum is marked as excluded and the cursor moves in the direction indicated. If the insert key is pressed, the datum is marked as included and the cursor moves in the direction indicated. Points included in the fit appear on the graph marked with "o" while points excluded are marked with "+". On the top line is an abbreviated menu to aid the user in key use:

[<->:MOVE CURSOR] [DEL:DELETE POINT] [INS:INCLUDE POINT] [F5:MARK] [END]

The function key F5 is used to mark two data values. Moving the cursor to a particular point on the graph and pressing F5 causes a double vertical bar to be placed at that point and the help message at the top of the screen to be changed to:

[<->:MOVE CURSOR] [DEL] [INS] [F5:MARK] [SHIFT-F5:UNMARK] [END]

with the Shift-F5 key now identified as the key to erase markers. When a second marker is placed on the screen, the help message changes to:

[F5:ZOOM] [SHIFT-F5:UNMARK] [F6:DELETE BLOCK] [SHIFT-F6:INCLUDE BLOCK] [END].

The delete and insert keys are still operative, but in addition the function key F6 will mark all points at and between the markers as excluded from the fit while Shift-F6 will mark all points at and between the markers as included in the fit. If F5 is pressed again, the display will be "zoomed" to the region including and between the markers. If the second and first marker are coincident on the display, the display will be zoomed to half the width of the current display with the marked datum as close to the center of the display as possible. If either the END or ENTER keys is pressed while two markers appear on the screen, the markers are used to delimit the pretransition, the transition and the posttransition regions. Subroutine ESTIM8 then uses these markers to draw straight lines through each of the three regions as a basis for the initial estimates of the parameters.

If the display is zoomed, the help message is changed to:

[<->:MOVE CURSOR] [DEL] [INS] [F5:MARK] [SHIFT-F5:UNZOOM] [END],

and the markers are erased. Thus, the editing session can be exited without worry that unseen markers are marking the transition regions. The "zoomed" display can be returned to the original display by pressing Shift-F5. As with the original display, the cursor movement keys, insert and delete keys and the function key F5 have their defined function. That is, while in a zoomed display, one can mark regions and delete or insert the marked blocks of data, delimit the transition regions, or zoom further.

6.5 PLOT

Once an analysis is completed, the results of the analysis can be graphically examined with the program PLOT. PLOT works with data files generated by LOGIT. On running PLOT, the user is prompted only to enter the limits of X values for the display:

```
ENTER VALUES FOR LOWER AND UPPER LIMITS OF X
TO BE PLOTTED.  PRESSING [ENTER] ALONE ACCEPTS DEFAULT VALUE

ENTER LOWER LIMIT FOR X: [  2.0]
ENTER UPPER LIMIT FOR X: [ 45.0]
CALCULATING . . .PLEASE WAIT
```

As with LOGIT, default values, based on the range in X values, are displayed in brackets. After the limits are entered, PLOT requires a few seconds to calculate all of the calculated curves to be plotted. When complete, the screen is cleared and a display similar to Figure 3 appears.

As in the editing of data, the options available for inspection of the data are summarized in a help line at the top of the screen. Appearing in the display are the original data. Data included in the fit are displayed with the symbol "o" and data excluded with the symbol "-". At each datum is a vertical bar corresponding to the 95% confidence intervals for that measurement. For uniform weighting, the vertical bar is just the standard deviation multiplied by the one-sided 97.5 percentile values for the t distribution. (For more than 30 degrees of freedom, the 97.5 percentile values for the t distribution are taken to be ± 2.) The solid curve is the calculated curve and it is flanked by two dotted curves representing the 95% confidence interval centered vertically about the calculated curve. A vertical line is drawn at the midpoint of the transition. The values of X(min) and X(max) printed at the bottom of the display are the minimum and maximum values of X included in the fit. If these are other than the first and last values of X in the data set, they are marked with arrows in the display. If the ENTER key is pressed, the help line at the top is replaced with a title line containing the title from the title line of the data, the date on which the numerical analysis was performed, and the standard deviation of the fit.

Printed on the graph are the values of the varied parameters along with their standard deviations.

Several options are available for additional inspection of the data.

If F2 is pressed, the derivative of Y with respect to X is displayed. The experimental points in the derivative plot are the first differences of the data and the solid curve is the derivative calculated from the values of the parameters.

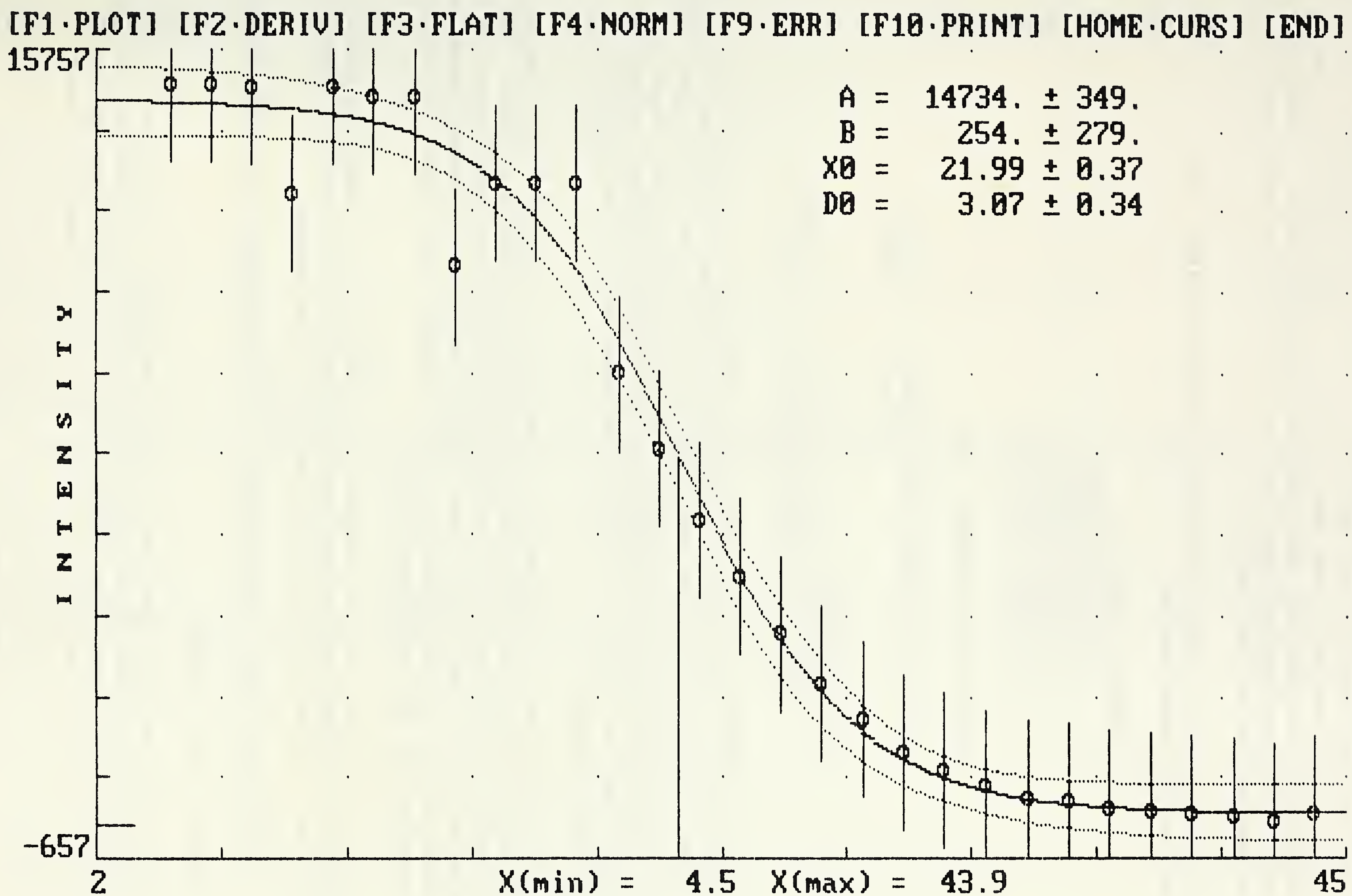


Figure 3. Data Analysis Plot

If F3 is pressed, the calculated curve is plotted as the primary curve with error bars representing 95% confidence intervals. The density of calculated values on the screen is such that the plot presents a grey band across the screen. The observed points are also placed on the screen but without reference to whether they were included in the fit or not. Using the cursor, extrapolated and interpolated values are obtained from the graph and printed on the bottom line of the display.

If F4 is pressed, the data are normalized and plotted. The user is given four choices for normalization:

```
NORMALIZE PLOT ON:
    [1] UPPER ASYMPTOTE
    [2] LOWER ASYMPTOTE
    [3] MAXIMUM VALUE OF Y
    [4] MINIMUM VALUE OF Y

WHICH? ENTER NUMBER: [1]

FOR THE NORMALIZED DATA, Y(min) =   0.0178
                         Y(max) =   0.9985

FOR THE GRAPH:
    ENTER THE MINIMUM VALUE OF Y [  0.0178] 0
    ENTER THE MAXIMUM VALUE OF Y [  0.9985] 1
```

After the datum or parameter that will establish the normalization constant (normalized to 1) has been selected and the maximum and minimum normalized values of Y to be displayed have been entered, the data are graphed. The normalization constant, X range and normalized Y range then remain constant for all additional data files analyzed in the analysis session. In this way, measurements on different quantities of materials can be compared in a systematic manner.

If F9 is pressed, the standardized residuals are plotted. The standardized residuals for the data depicted in Figure 3 are shown in Figure 4. The systematic variation of the standardized residuals throughout most of the range along with the clustering of most of the standardized residuals near zero are characteristic of the effects of a few outliers. The horizontal, dotted lines represent the 95% confidence limits.

If F10 is pressed and the computer is connected to a compatible graphics printer, the help line is replaced with the title line and the screen is printed. If the printer is not compatible, the message:

```
PRINTER NOT READY, NOT AVAILABLE, OR NOT COMPATIBLE - ANY KEY TO CONTINUE
```

appears on the top line.

A few cursor-related functions are also available. In contrast to the data editing routine, the cursor must first be turned on by pressing the HOME key. (HOME also turns the cursor off when it is on.) The cursor appears at the leftmost point on the display and the help line is changed to read:

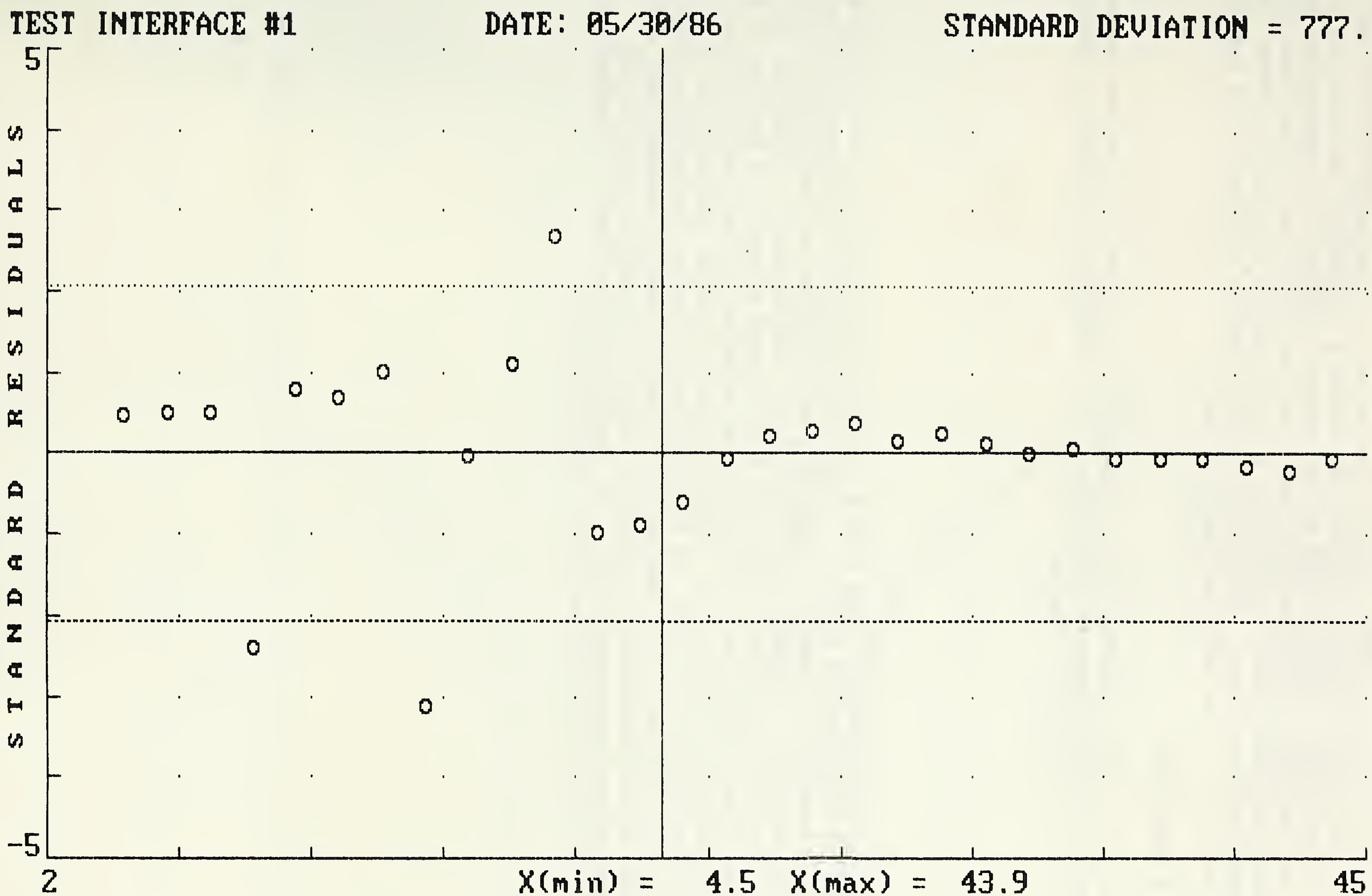


Figure 4. Standardized Residuals

[F1-F4:PLOT] [F9:ERR] [HOME:CURSOR OFF] [<-->:CURSOR MOVE] [F5:MARK] [END]

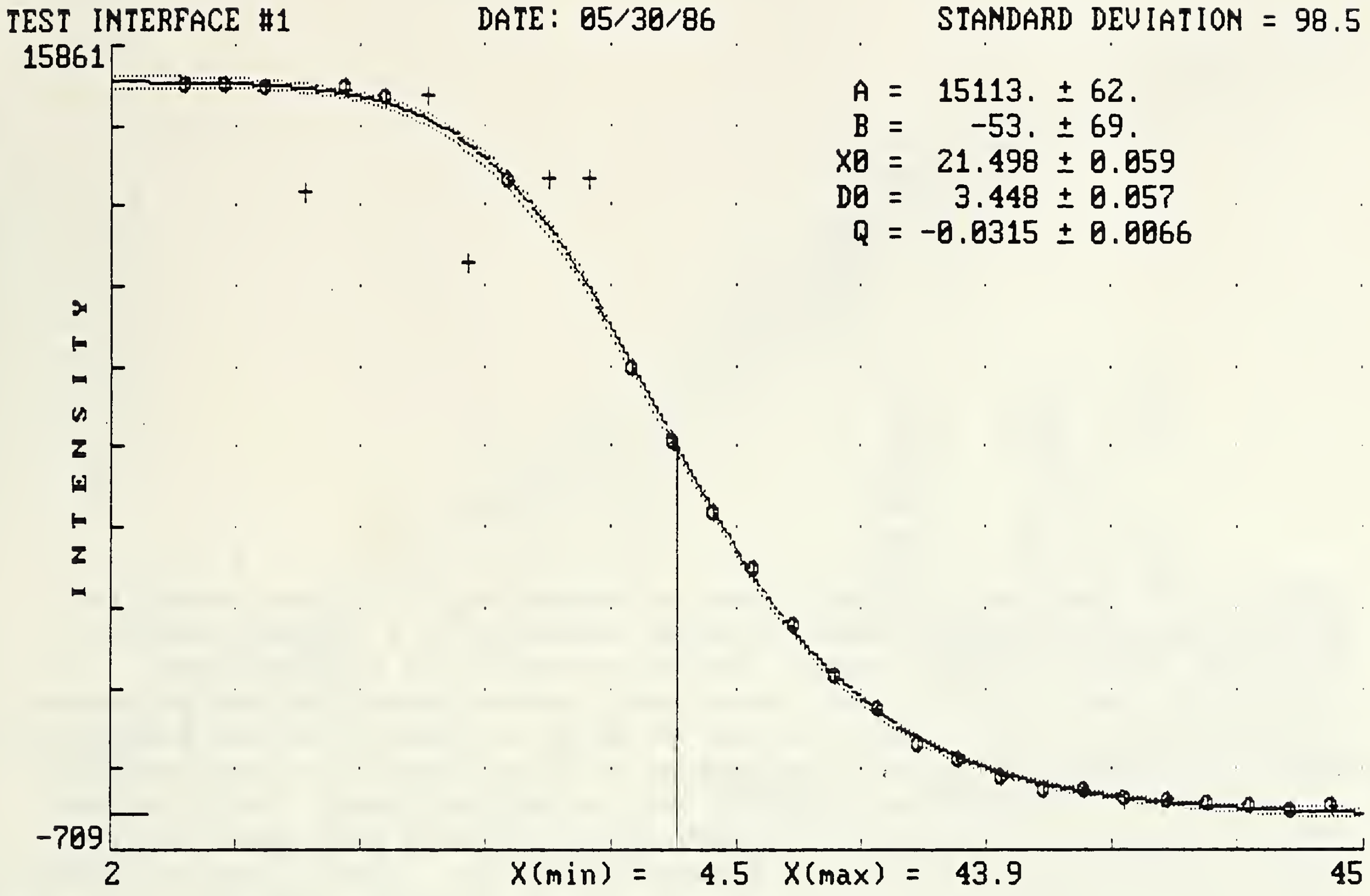
As with the data editing routine, the cursor is moved with the left and right arrow keys and the F5 key is used for marking regions of the data for zooming. Only one marker can be placed on the screen. To zoom to a particular region, place the marker at one end of the region and the cursor at the other end. Pressing F5 will cause the display to be zoomed to the region between the marker and the cursor. Shift-F5 is used to unmark a marker or to restore a zoomed plot to its original display range.

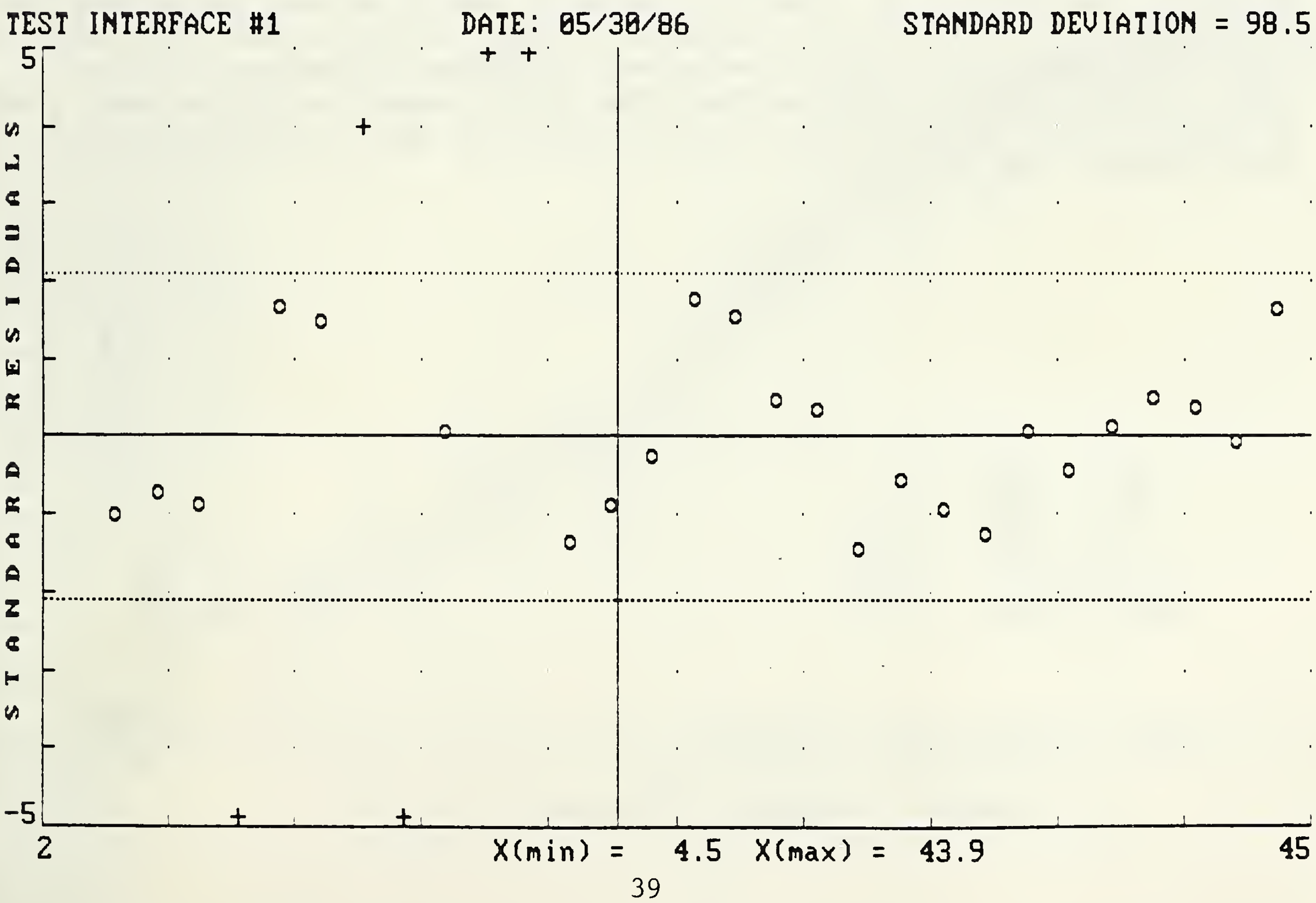
7.0 TEST DATA

Six data files, TESTDATA.001 - TESTDATA.006 accompany the program and may be used to evaluate program performance. The results of analyses of these test data are displayed in Figures 5-10.

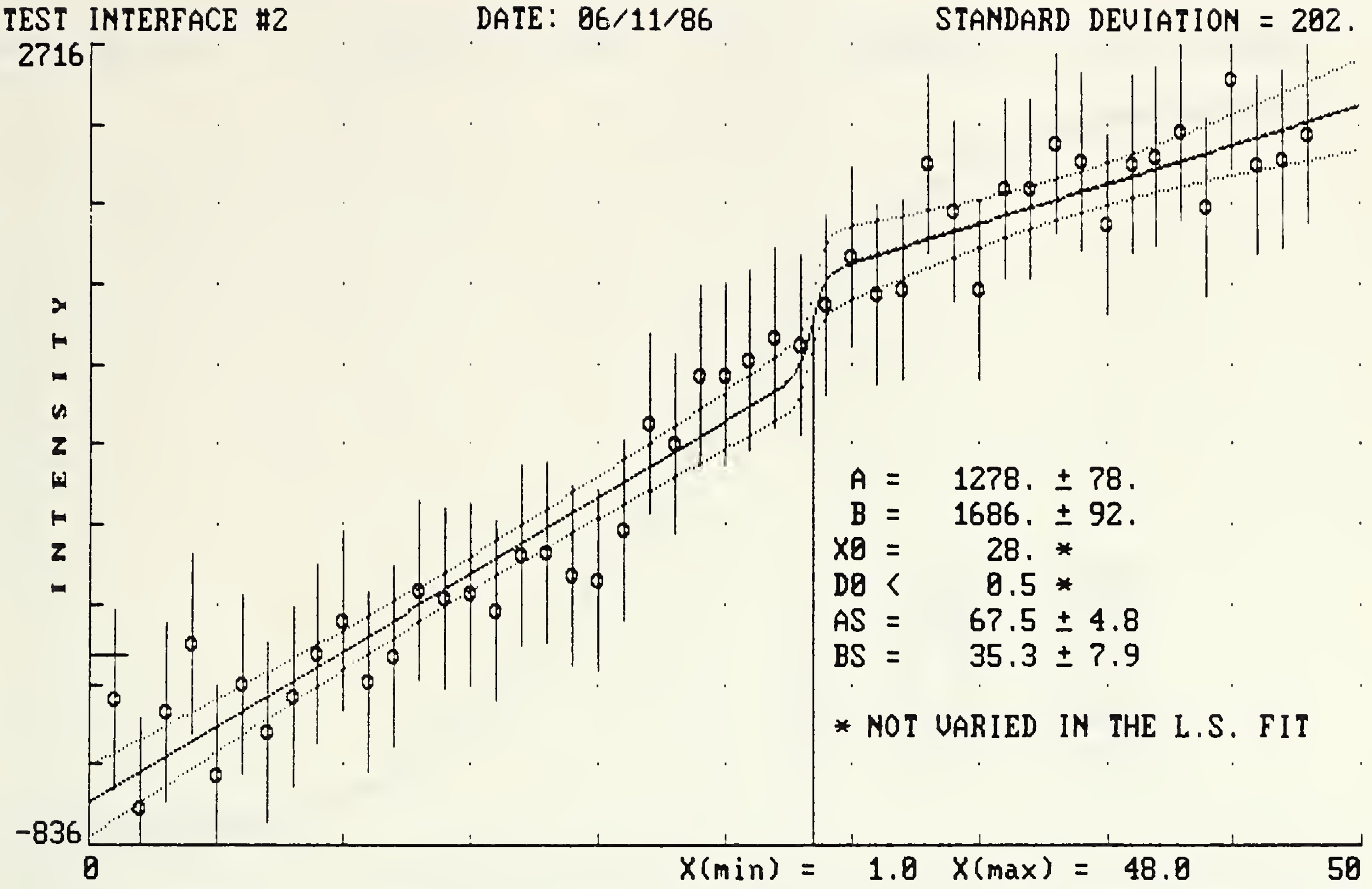
TESTDATA.001 contains data from a depth profile analysis of an interface between chromium and nickel. It has been discussed in the preceding text and is included as an example of outlier identification and asymptotic slope indeterminacy. In analyzing TESTDATA.001, the asymptotic slopes AS and BS and the asymmetry parameter Q should all be allowed to vary. A maximum of 6 retries for outlier rejection should be selected and the outlier rejection value should be set at 3.0. LOGIT should be able to identify the five outliers and should determine that the slopes AS and BS cannot be evaluated from the data. In Figure 5. the results of the analysis are presented along with a graph of the standardized residuals. Note the random scatter in the standardized residuals. Also, the standardized residuals of the outliers appear on the top and bottom line of the graph.

Figure 5. Results of Analysis of TESTDATA.001





TESTDATA.002 represents synthetic data corresponding to a broad transition interval to which random, normal deviates amounting to a relative standard deviation of 20% have been added. The broad profile is difficult to distinguish from a profile with an extremely narrow transition and asymptotes with pronounced slopes. Thus, in the top graph of Figure 5 is the result of an analysis in which the seven parameters A, B, X_o, D_o, A_s, B_s, and Q are all varied and the post-fitting, interval test is not performed. LOGIT deduced that the transition interval was narrow because no data fell more than two standard deviations away from its nearest asymptote. Hence, Q was set equal to 0 (its value is missing from the parameter list), T_o at the midpoint of the two data values spanning the intervals, and D_o at a value dependent on the standard deviation of the fit and the separation between adjacent data points. When the post-fitting interval test is performed, the "erroneous" assumption that no point lies in the interval is uncovered and the fit is repeated holding A_s and B_s fixed at zero. The results of this second fit appear in the lower graph of Figure 6. When the fit is repeated allowing A_s and B_s to vary but holding Q=0 (not pictured), a fit of nearly identical quality to that in the lower graph of Figure 6 is obtained but with substantially different parameters. In this case, the fit cannot be distinguished from the fit appearing in the lower graph of Figure 6, and criteria other than the quality of the fit must be used to select between the two.



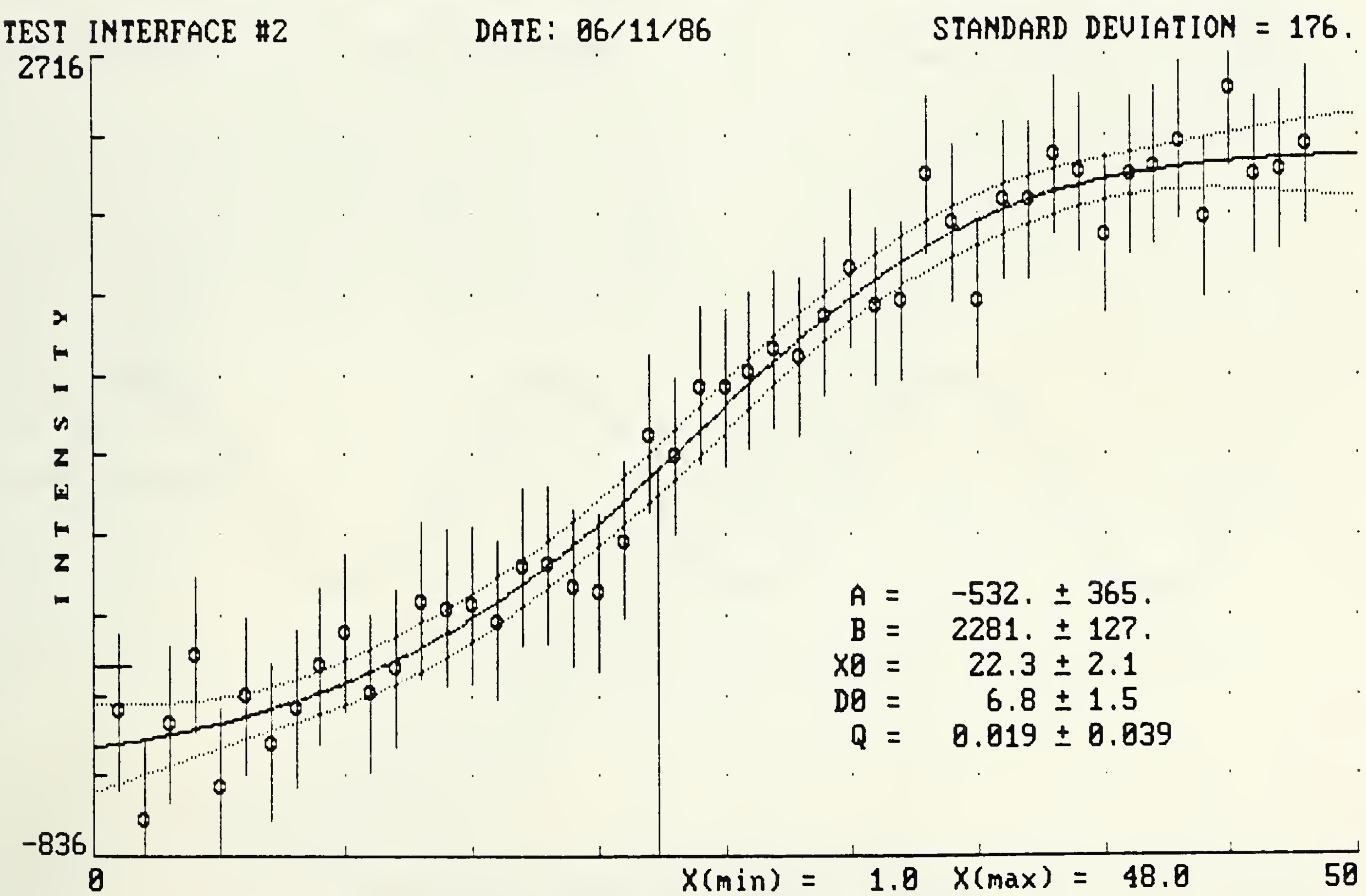
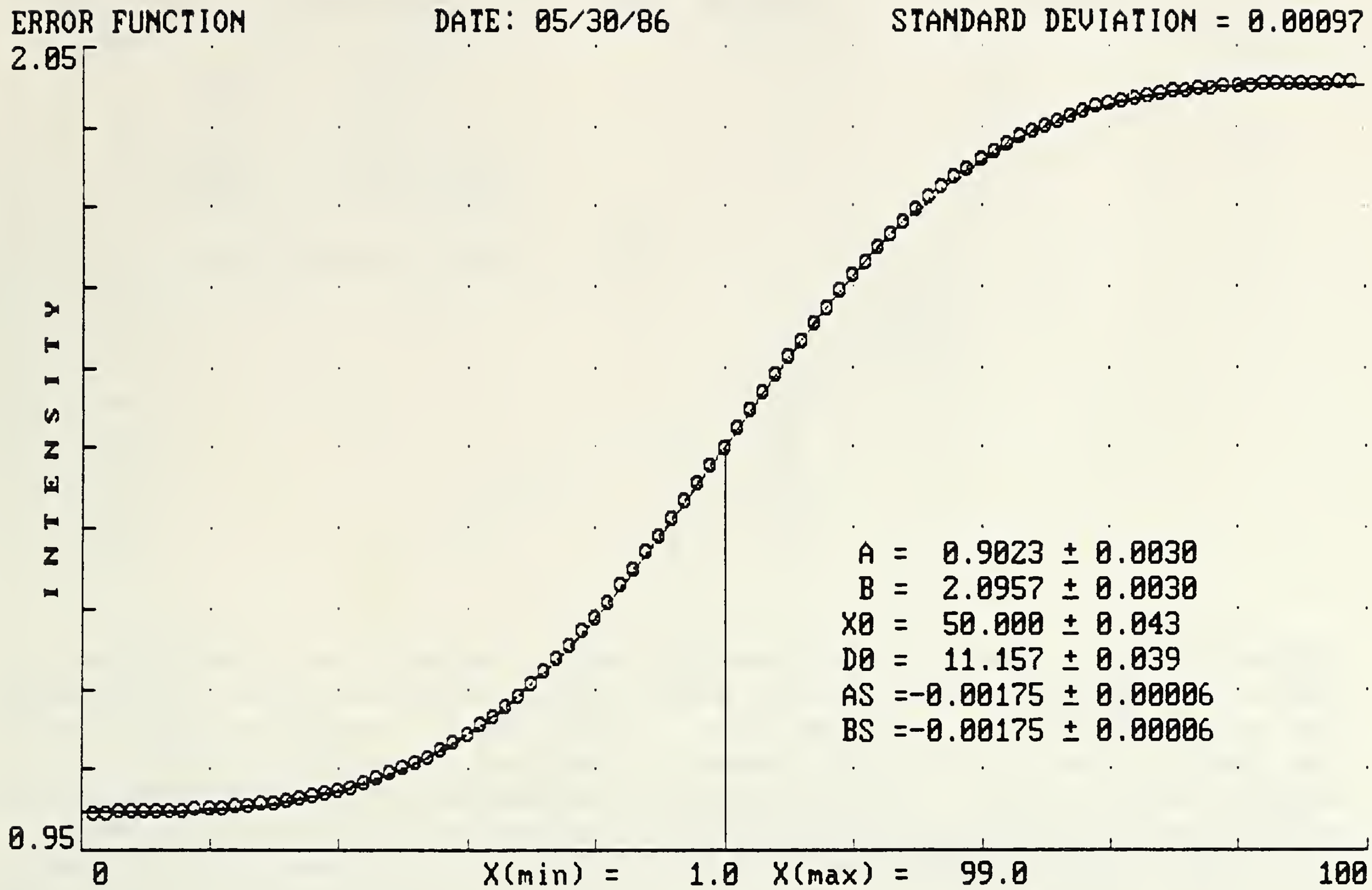


Figure 6. Results of Analysis of TESTDATA.002

TESTDATA.003, is the error function. It is included as a test data file in order to demonstrate the closeness of overlap between the logistic function and the error function and as a demonstration of the effects of systematic error on the standardized residuals. Although the fit of the data appears to be quite good, the systematic nature of the deviations becomes quite apparent in the standardized residuals.



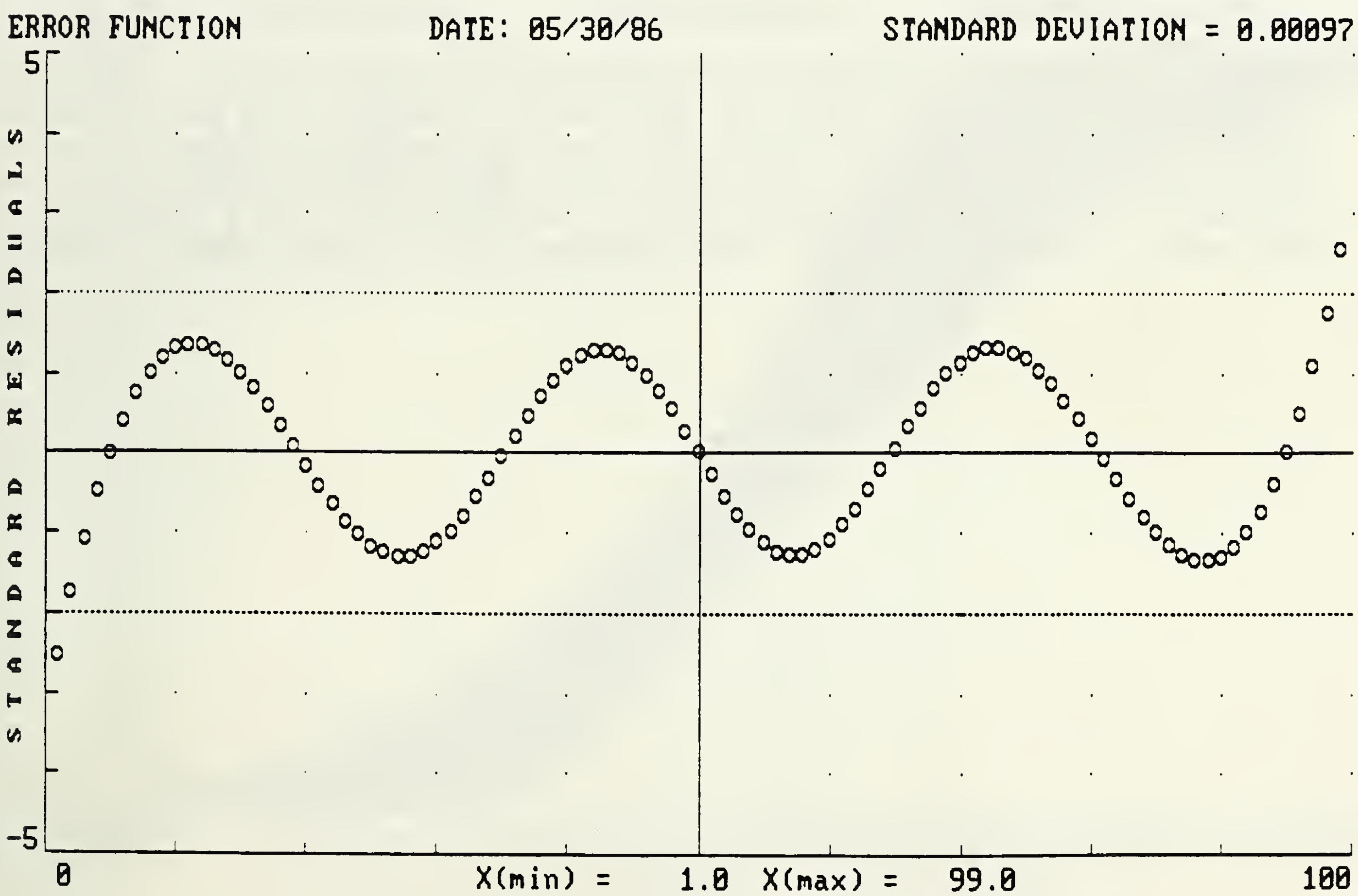
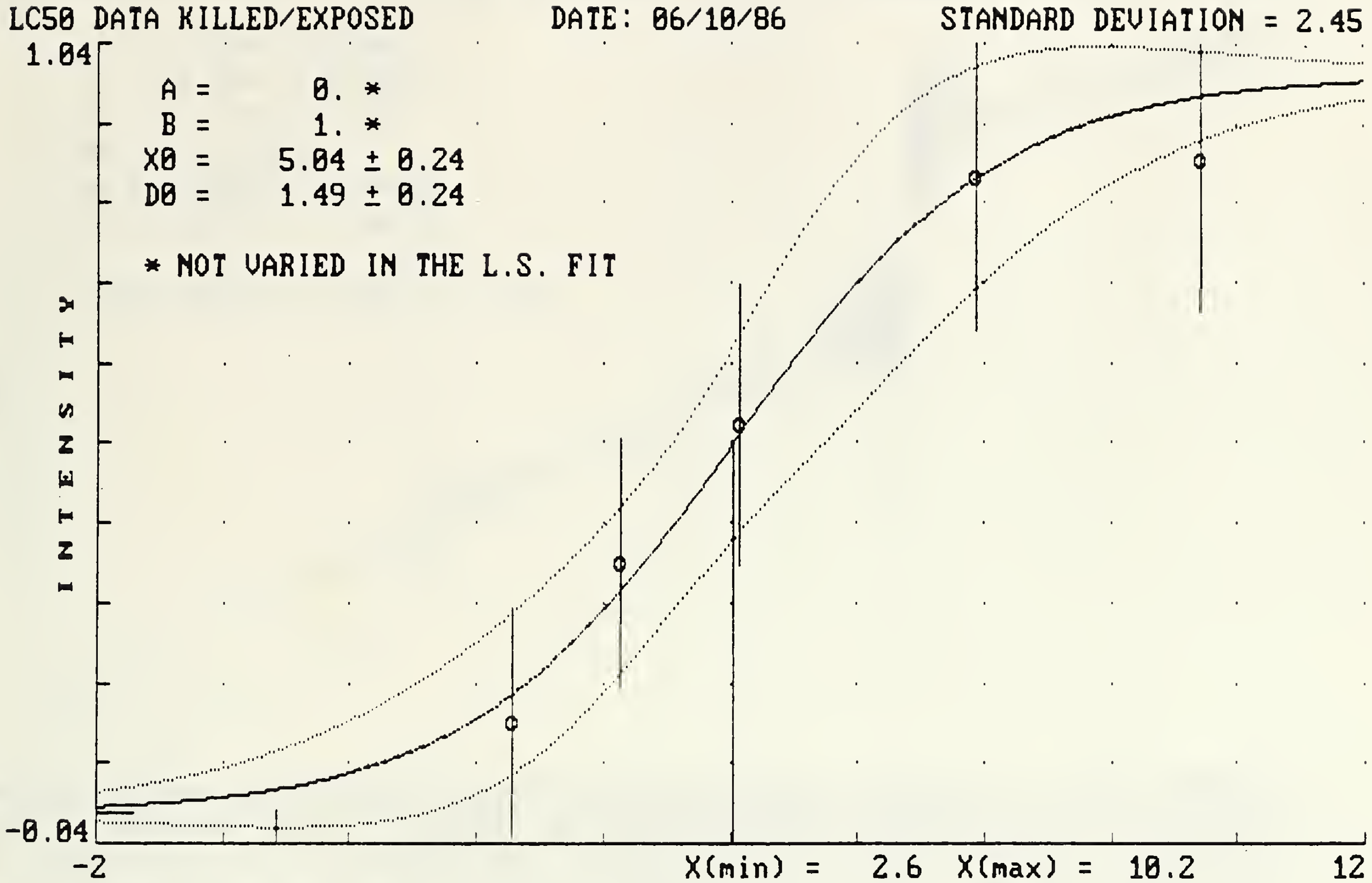


Figure 7. Results of Analysis of TESTDATA.003

TESTDATA.004, is a small set of toxicity data. The data set contains the number of organisms tested and the number of organisms killed as a function of the concentration of a known toxicant. In the fit of these data, the ratio of the number killed to the number tested is analyzed (Data Type 2 in the configuration session). The pretransition asymptote should be held fixed at 0 and the posttransition asymptote at 1.0. This set of configurations is contained on the configuration file LC50.OPT which is included with the data. Therefore, typing LC50 in response to the question "RECONFIGURE Y/N? [N]" with LC50 will invoke this configuration. Alternatively, the appropriate options may be entered from the keyboard. Note that in this data set and the next, TESTDATA.005, when ratios are being tested, the least squares fit is a weighted fit with weights:

$$W_i = Y_{di}^2/(1+r_i^2)$$

where Y_{di} is the i^{th} value of Y in the denominator of the ratio r_i. The 95% confidence intervals are seen to be quite large in keeping with the 3 degrees of freedom. Figure 8 consists of 1) the standard graphical display of the data and their error bars and the calculated curve and its 95% confidence limit, and 2) a plot of the calculated curve with 95% confidence error bars associated with each calculated point, the observed values, and a cursor placed at X=6.52.



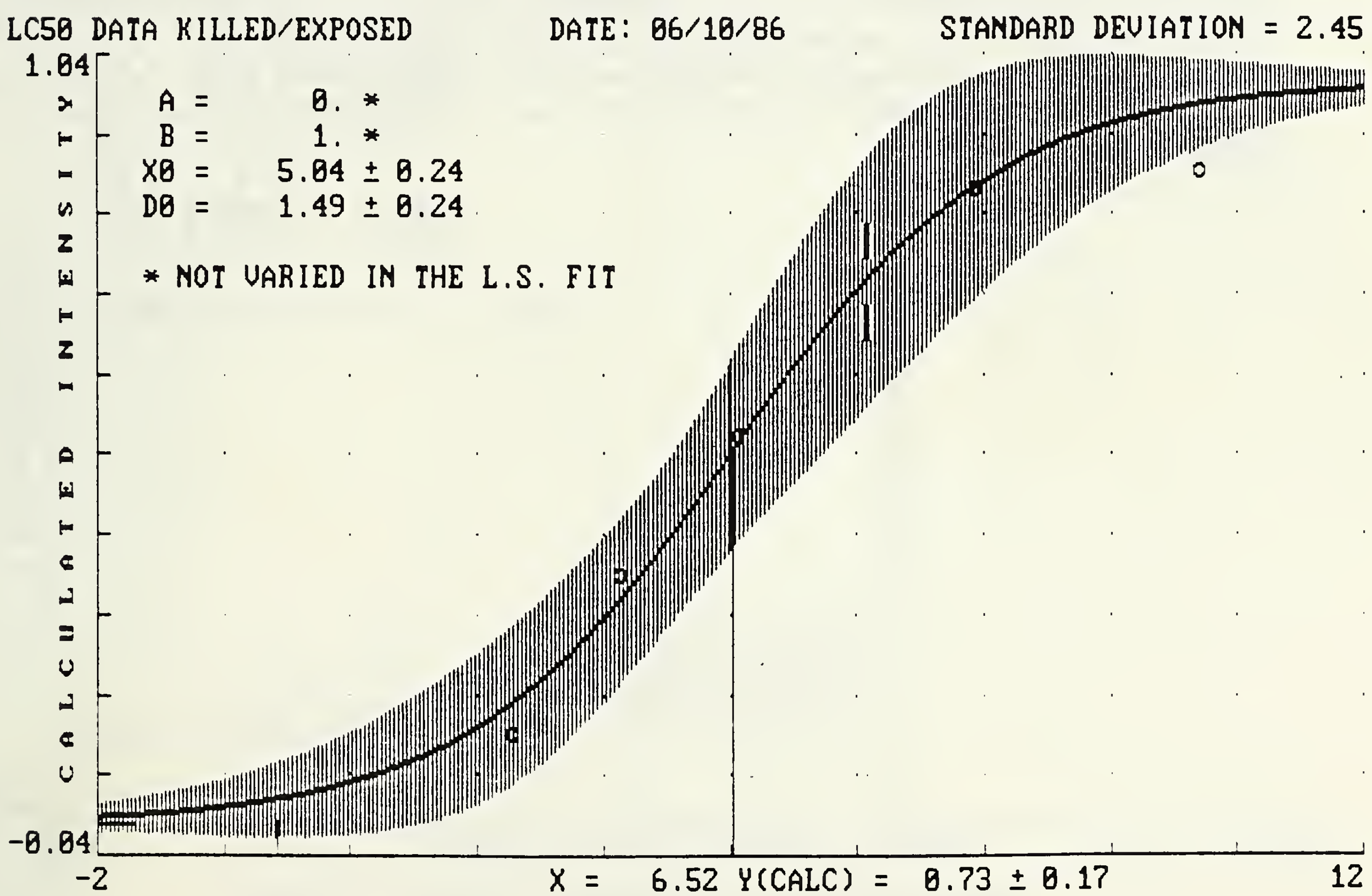
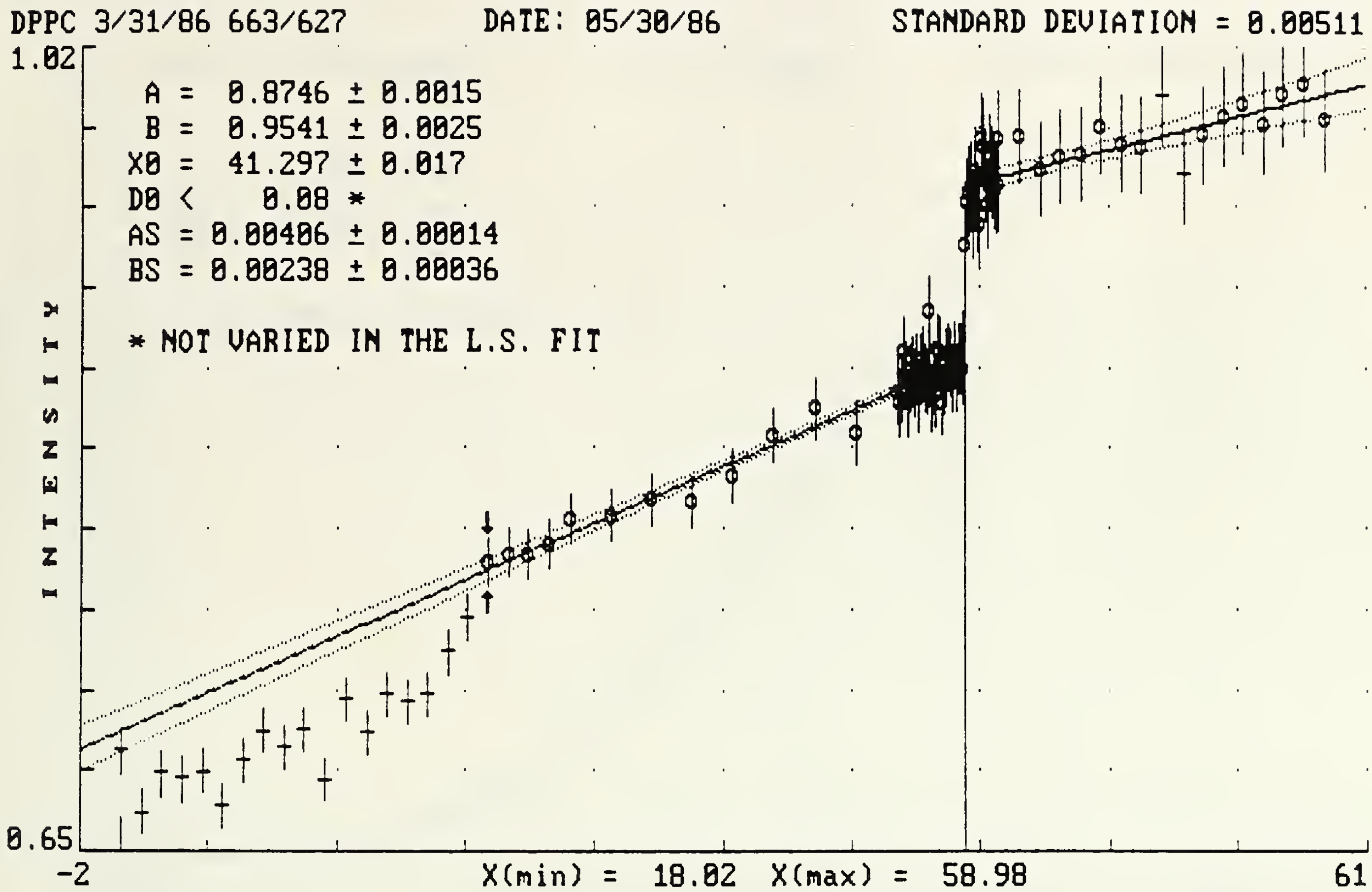


Figure 8. Results of Analysis of TESTDATA.004

TESTDATA.005, is a phase transition curve for an ensemble of membrane bilayers composed of dipalmitoylphosphatidylcholine (DPPC). In this analysis, the ratios of Y values labeled as 663 and 627 were analyzed. The transition is very sharp and the width of the interval cannot be determined (though it can be determined for the 575/627 and 575/663 ratios.) Figure 10 depicts the zoomed region around the transition for TESTDATA.005. It can be seen that the 95% confidence intervals for the posttransition data are larger than those for the pretransition data. This is because the errors in the ratios are not normally distributed, which the errors in the primary data are assumed to be. Hence the fit was a weighted fit with each ratio being weighted in proportion to the inverse square of its uncertainty as described for TESTDATA.004. Figure 9 includes a standard graphical display of the original data with the calculated curves and also a "zoomed" graph of the cluster of points near the transition region.



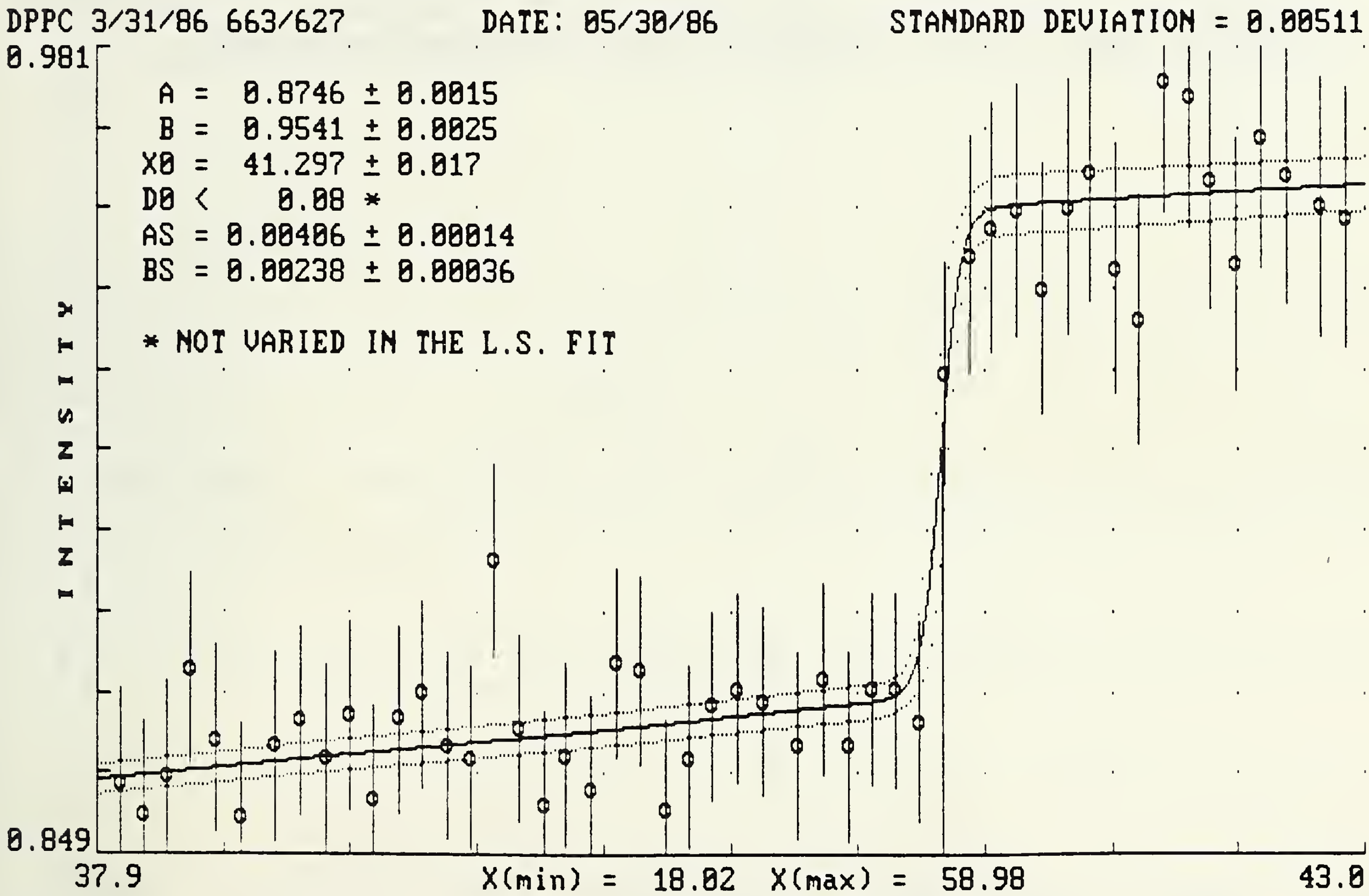
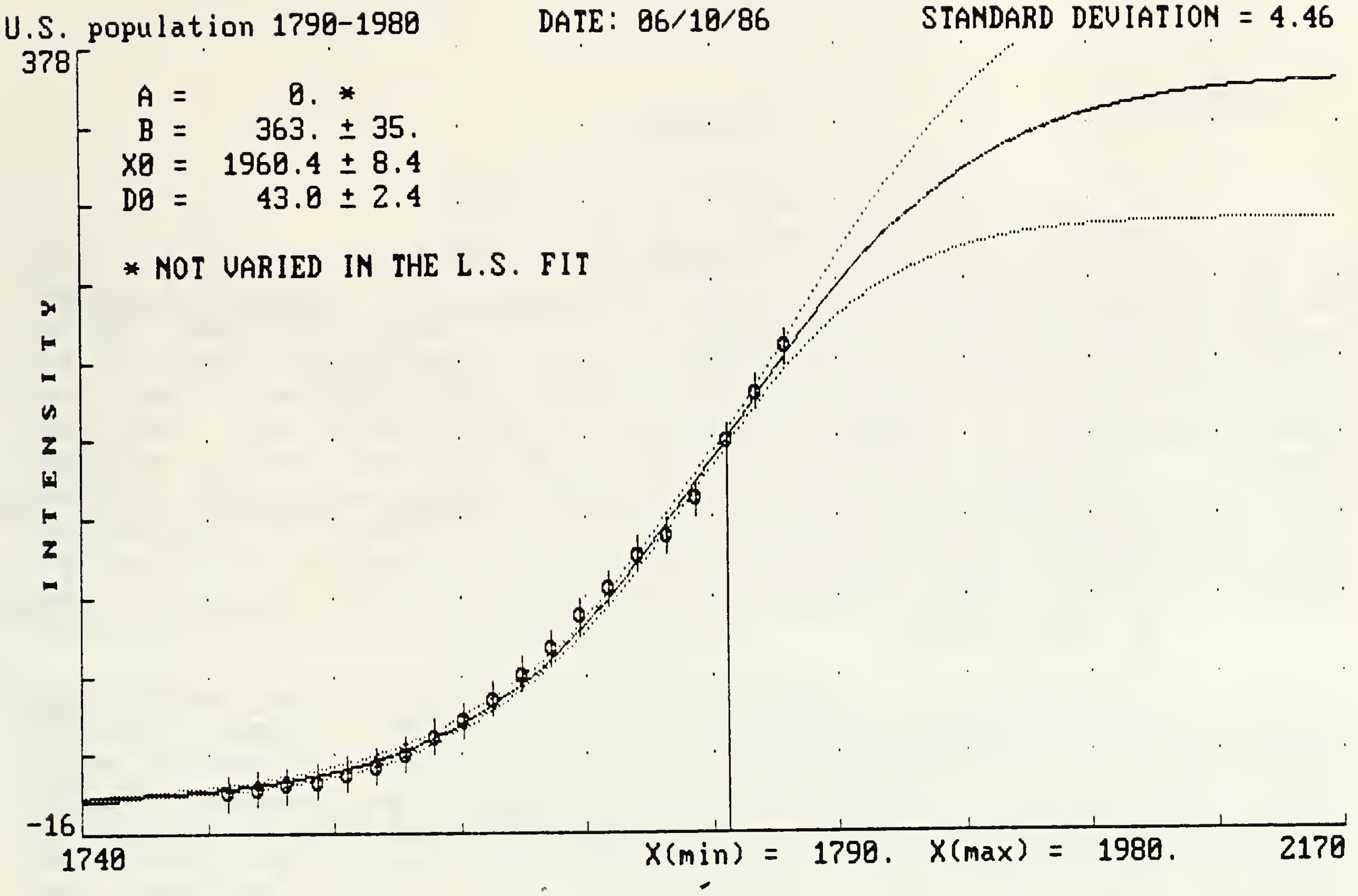


Figure 9. Results of Analysis of TESTDATA.005

TESTDATA.006 contains population figures for the U.S. in decade intervals since 1790. It is an extension of the data originally analyzed by Pearl and Reed[2] but to which data from 1920 through 1970 have been added. This data was analyzed allowing the asymptotes to vary but holding the slopes of the asymptotes fixed at zero. It serves as a demonstration of the extrapolation capability of the analysis. Also, the plot of the standardized residuals in Figure 10 shows a trend suggestive of the limitations of the model.



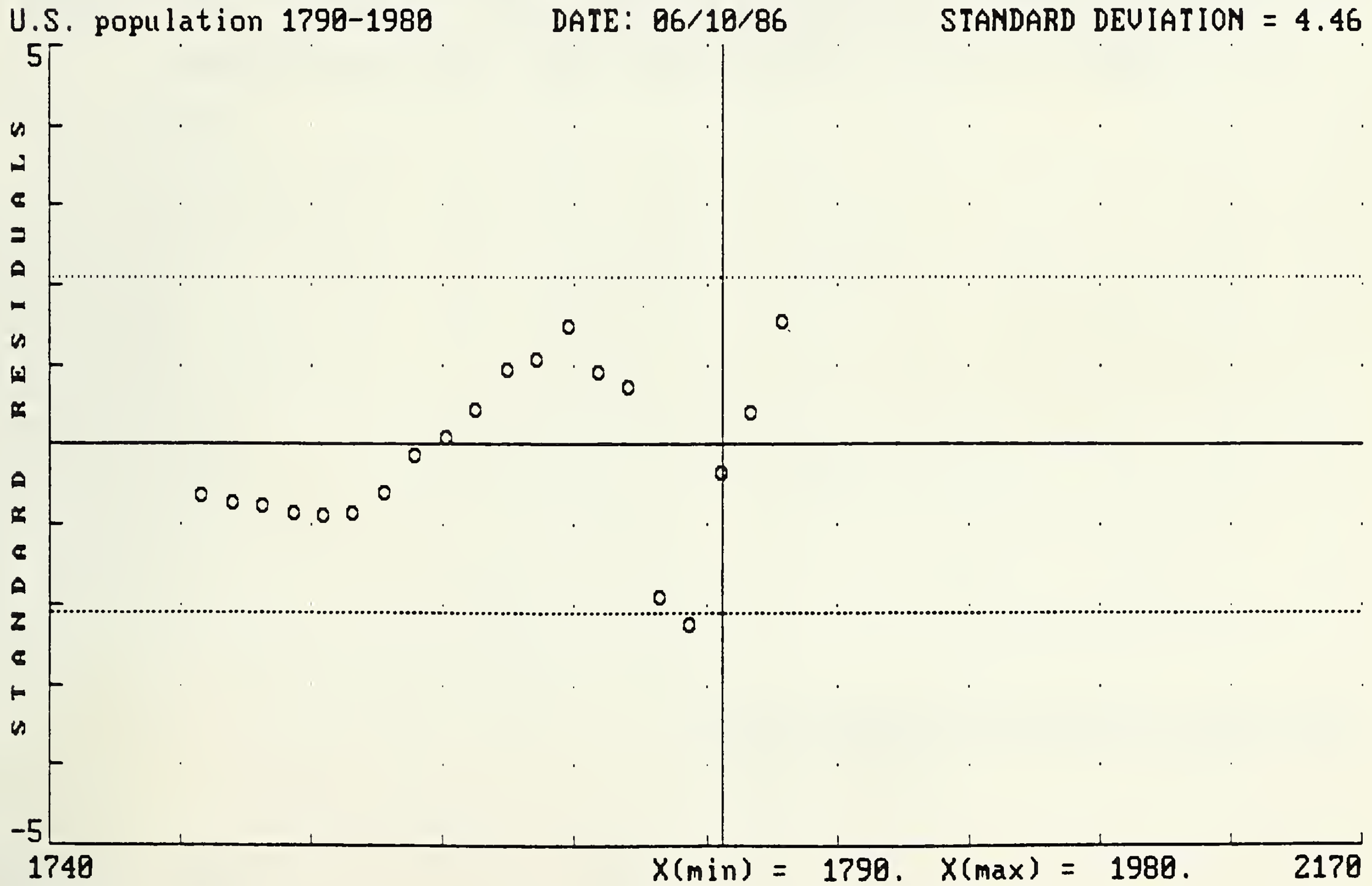


Figure 10. Results of Analysis of TESTDATA.006

REFERENCES

1. Verhulst, P.F. Acad. Brux., 18, 1-38 (1845).

2. Pearl, R. and Reed, L. J., Proc. Nat. Acad. Sci., 6, 275-88 (1920).

3. Oliver, F. R., Applied Statistics, 13, 57-66 (1964).

4. Berkson, J., Biometrics, 7, 327-339 (1951).

5. Reed, L. J. and Berkson, J., Journal of Physical Chemistry, 33 760-779 (1929).

6. Kirchhoff, W.H., Levin, I.W., and Hancock, A. J., Biophysical Journal, 45, 196a, (1984).

7. Kirchhoff, W. H., Chambers, G. P., and Fine, J., J. Vac. Sci. Technol. (Series A,in press,1986).

8. Johnson, N. L. and Kotz, S. *Distributions in Statistics: Continuous Univariate Distributions*, Houghton Mifflin Co., Boston, 2, Chap. 22 (1970).

9. Wilson, Jr., E. B., *An Introduction to Scientific Research*, McGraw-Hill Book Co., New York, 234-46, (1952).

10. Walsh, P. J., *Algorithm 127, ORTHO*, Communications of the ACM, 5, 511-516, (1962).

11. Wampler, R. H., J. Res. Nat. Bur. Stds., 73B, 59-90 (1969).

APPENDIX A

PROGRAM AND SUBROUTINE LISTINGS

The following pages contain the listings of the programs LOGIT and PLOT and their subroutines. The subroutines appear in the order in which they are called. SUBROUTINES DERIV, KEY, AND LNBX are called by both LOGIT and PLOT and are not duplicated following the listing of PLOT. Note that although LOGIT is the name of the executable program, the main program is stored in the file MAIN.FOR. The program PLOT and SUBROUTINE SELECT in LOGIT call several plotting library functions which are not generally available. If these programs are to be adapted to a new computing environment, new plotting routines will be required. Many plotting libraries are similar in their calls and functions. Therefore, the listing of PLOT has beem included as a guide since it may be possible to revise the program by little more than replacing the calls to the various plotting routines.

The programs and subroutines have been written in FORTRAN 77 and contain extensive comments. The main programs and subroutine OPTS contain input and output statements of the form READ (*,...) and WRITE (*,...) with the intention that the input is entered from the keyboard and the output appears on the screen. For some installations, the default input and output devices are not the keyboard and screen, but a primary disk. If the program does not accept input from the keyboard and/or print prompts to the screen (as outlined in APPENDIX A), the READ (*,...) statements should be replaced with READ (L0,...) and the WRITE (*,...) with WRITE (LP0,...) (Note, L"zero" and LP"zero"). The integer variable L0 has been reserved as the keyboard input unit and LP0 has been reserved as the screen output unit. Both L0 and LP0 are defined at the beginning of the main program. It may be necessary to redefine L0 and LP0 to coincide with a particular installation's standard input and output units.

```
C                             MAIN PROGRAM
C
C             *****    LOGISTIC FUNCTION CURVE FIT   *****
C
C             NINE PARAMETER FIT TO AN EQUATION OF THE FORM:
C
C  Y(X) = (A+(AS+AQ(X-X0))(X-X0))/(1+EXP(Z))
C             + (B+(BS+BQ(X-X0))(X-X0))/(1+EXP(-Z))
C
C         WHERE Z = (X-X0)/D AND D = 2D0/(1+EXP(Q(X-X0)))
C
C                         C(1) =   A
C                         C(2) =   B
C                         C(3) =   T0
C                         C(4) =   D0
C                         C(5) =   AS
C                         C(6) =   BS
C                         C(7) =   AQ
C                         C(8) =   BQ
C                         C(9) =   Q
C
C
C                       VERSION 6/10/86
C
      COMMON/CFGPARM/KTYPE,LOCK,IFPLOT,MODRES,MODLOG
      COMMON/IOPARM/L0,L1,L2,L3
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/CONTROL/IFINIT,INSEQ,NIT,NKOUT,YOUT,ITG(4),IFC(9),IFCIN(9)
      COMMON/STAT/COR(9),BS(9),VCV(9,9),NZ,MZ,STD,REL,GMDT,ESTSTD
      COMMON/MISC/W0,W1,KIT,NCYC
      COMMON/DATA/N,X(200),Y(200),W(200),IFIO(200),NMARK,MARK(2)
      COMMON/SCRATCH/Z(11,209)
      CHARACTER DATAF*24,Q*1,TITLEF*30,RDAT*8,RTIM*11,DATLIN*16
      CHARACTER OPF*8,OPFE*12
      CHARACTER VERS*8,FNAM(5)*12
      DATA VERS/' 6/10/86'/
      DATA FNAM/'CON','PRN','RESULTS','SESSION.LOG','CON'/
C
C *** ESTABLISH LOGICAL UNITS FOR INPUT AND OUTPUT
C *** L0  IS RESERVED FOR SCREEN OUTPUT AND KEYBOARD INPUT
C *** LPO IS RESERVED FOR THE LOCAL PRINTER
C *** L1  IS THE OUTPUT FOR THE RESULTS OF THE ANALYSIS AND CAN
C ***        BE THE SCREEN, A PRINTER, OR THE FILE 'RESULTS'
C *** L2  IS THE OUTPUT UNIT FOR INTERMEDIATE RESULTS WHICH CAN
C ***        BE THE SCREEN, A PRINTER, OR THE FILE 'SESSION.LOG'
C *** L3  IS A CURRENT DISK\DIRECTORY INPUT AND OUTPUT UNIT
C ***        IT IS USED FOR READING AND WRITING THE 'OPTIONS.LOG'
C ***        AND READING DATA FILES
C *** L4  IS A CURRENT DISK\DIRECTORY INPUT FOR SAVING THE DATA
C ***        FOR THE PLOTTING PROGRAM.  IF OPENED, IT REMAINS
C ***        OPEN UNTIL THE END OF THE SESSION
C
      L0 = 5
      LPO = 6
      L1 = 1
      L2 = 2
      L3 = 3
      L4 = 4
      WRITE (*,900) VERS
C
C *** READ THE OPTIONS FILE.  IF ONE DOESN'T EXIST, CREATE ONE.
C *** THE DEFAULT OPTIONS FILE IS DFLT.OPT.  THE FIRST QUESTION IS
C *** RECONFIGURE? -Y/N [N].  IF Y, THE DEFAULT OPTIONS FILE WILL
C *** BE READ AND THE USER IS GIVEN THE OPPORTUNITY TO CHANGE ALL
C *** OPTIONS.  IF ANY OTHER ENTRY OF LESS THAN THREE CHARACTERS
C *** IS GIVEN, THE DEFAULT OPTIONS FILE IS READ AND THE PROGRAM
C *** CONTINUES.  IF A NAME OF 3 TO 8 CHARACTERS IS ENTERED, IT
C *** IS APPENDED WITH THE EXTENSION .OPT AND IS TAKEN TO BE
C *** THE NAME OF THE OPTIONS FILE.  THE USER IS NOT GIVEN THE
C *** OPTION OF CHANGING THE OPTIONS IF AN OPTIONS FILE OTHER THAN
C *** THE DEFAULT IS READ.
C
    1 OPFE='DFLT.OPT'
      WRITE (*,910)
      READ (*,'(A)') OPF
      L=INDEX(OPF,' ')-1
      IF (L .LE. 2) GO TO 10
      OPFE=OPF(1:L)//'.OPT'
   10 OPEN (L3,FILE=OPFE,STATUS='OLD',ERR=11)
C
C *** READ THE VALUES OF THE CONTROL PARAMETERS ET AL.
C
      CALL OPTS(1)
      GO TO 12
   11 WRITE (*,'(''$ ^GDO YOU WISH TO ESTABLISH OPTIONS FILE: ''
     1,A,'' - Y/N [N]'')') OPFE
      READ (*,'(A1)') Q
      IF (Q .NE. 'Y' .AND. Q .NE. 'y') GO TO 1
C
C *** IF YES, CREATE THE NAMED OPTIONS FILE
C
      OPEN (L3,FILE=OPFE,STATUS='NEW',ERR=500)
      CALL OPTS(2)
C
C *** NOW OFFER THE OPPORTUNITY TO CHANGE ANY OF THE OPTIONS IN THE
C *** DEFAULT OR SPECIFIED OPTIONS FILE IF ANY ANSWER BEGINNING WITH
C *** Y WAS OFFERED IN RESPONSE THE THE QUESTION "RECONFIGURE?"
C
   12 IF (OPF(1:1) .EQ. 'Y' .OR. OPF(1:1) .EQ. 'y') CALL OPTS(3)
      CLOSE (L3)
C
C *** OPEN OUTPUT FILES IN ACCORDANCE WITH THE OPTIONS SELECTED.
C
   40 IF (MODRES .EQ. 5) THEN
        WRITE (*,912)
        READ (*,902) DATAF
        IF (DATAF .EQ. ' ') MODRES = 3
```

```
        FNAM(5) = DATAF
      END IF
      IF (MODRES .EQ. 1) L1=L0
      IF (MODRES .EQ. 2) L1=LP0
      OPEN (L1,FILE=FNAM(MODRES),STATUS='OLD',ERR=42)
      GO TO 43
   42 OPEN (L1,FILE=FNAM(MODRES),STATUS='NEW',ERR=501)
   43 IF (MODLOG .EQ. 1) L2=L0
      IF (MODLOG .EQ. 2) L2=LP0
      IF (MODLOG .EQ. MODRES) L2=L1
      OPEN (L2,FILE=FNAM(MODLOG),STATUS='OLD',ERR=44)
      GO TO 45
   44 OPEN (L2,FILE=FNAM(MODLOG),STATUS='NEW',ERR=502)
   45 IF (MODRES .NE. 1) WRITE (L1,900) VERS
      IF (MODLOG .NE. MODRES .AND. MODLOG .NE. 1) WRITE (L2,900) VERS
C
C *** RDDATA READS IN A DATA FILE AND, IF APPROPRIATE, CALCULATES
C *** THE WEIGHTS
C
   50 CALL TIME (RTIM)
      CALL DATE (RDAT)
      CALL RDDATA (DATAF,TITLEF)
      IF (MODLOG .GT. 2 ) WRITE (L2,996) L2,DATAF,RDAT,RTIM
C
C *** IF LOCK = 1, NO OPTIONS ARE ALLOWED
C
  100 IF (LOCK .EQ. 1) GO TO 110
      WRITE (*,911)
      READ (*,902) Q
C
C *** ASK IF THE USER WANTS TO CHANGE THE OPTIONS WHICH INCLUDES EDITING
C *** THE DATA.  IF YES, THE OPTIONS ARE OFFERED BY OPTS.
C
      IF (Q .EQ. 'Y' .OR. Q .EQ. 'y') CALL OPTS(4)
C
C *** WRITE TITLES TO EACH OF THE OUTPUT FILES AND BEGIN THE ANALYSIS
C
  110 WRITE (L1,992) DATAF,RDAT,RTIM
      WRITE (L1,993) TITLEF
      CALL TIME(RTIM)
      IF (MODRES .NE. MODLOG) WRITE (L2,997) RTIM
      IF (MODLOG .NE. 1) WRITE (*,994) RTIM
      CALL LGSTC
      CALL TIME(RTIM)
      WRITE (L2,998) RTIM
      IF (MODLOG .NE. 1) WRITE (*,995) RTIM
C
C *** DELX(F) CALCULATES THE RANGE IN X FOR WHICH THE TRANSITION
C *** PROCEEDS FROM A FRACTION F OF COMPLETION TO 1-F
C
      CALL DELX(0.1)
C
C *** IF LOCK IS NOT "ON" THE USER IS GIVEN THE OPTION OF REFITTING
C *** THE DATA.  IF YES, THE OPTIONS ARE AGAIN OFFERED INCLUDING
C *** THE OPTION OF RE-INITIALIZING PARAMETERS. IF THIS OPTION IS
C *** REJECTED, THE INITIAL VALUES OF THE PARAMETERS FOR THE REFIT ARE
C *** THE VALUES OF THE PARAMETERS FROM THE PREVIOUS FIT.
C
      IF (LOCK .EQ. 1) GO TO 120
      WRITE (*,940)
      READ (*,903) Q
      IF (Q .EQ. 'Y' .OR. Q .EQ. 'y') THEN
        IFINIT = 1
        WRITE (*,941)
        READ (*,903) Q
        IF (Q .NE. 'Y' .AND. Q .NE. 'y') THEN
          DO 115 I=1,9
  115     CIN(I)=C(I)
          IFINIT = 3
        END IF
        WRITE (L1,955) TITLEF
        IF (MODRES .NE. MODLOG) WRITE (L2,955) TITLEF
        GO TO 100
      END IF
C
C *** SAVE DATA FOR PLOTTING?  IF IFPLOT=1, THE DATA WILL AUTOMATICALLY
C *** BE SAVED.  IF NOT, AND IF LOCK IS OFF, OPTION TO SAVE WILL BE
C *** GIVEN.
C
  120 IF (IFPLOT .EQ. 1) GO TO 121
      IF (LOCK .EQ. 1) GO TO 200
      WRITE (*,953)
      READ (*,903) Q
      IF (Q .NE. 'Y'.AND. Q .NE. 'y') GO TO 200
  121 OPEN (L4,FILE='PLOT.DAT',FORM='UNFORMATTED',STATUS='OLD',
     1ERR=122)
      GO TO 123
  122 OPEN (L4,FILE='PLOT.DAT',FORM='UNFORMATTED',STATUS='NEW',
     1ERR=503)
  123 WRITE (L4) TITLEF
      DATLIN='DATE: '//RDAT
      WRITE (L4) DATLIN
      WRITE (L4) (C(I), I=1,9)
      WRITE (L4) ((VCV(I,J), J=1,9), I=1,9)
      WRITE (L4) STD,N
      WRITE (L4) (X(I),Y(I),W(I),IFIO(I), I=1,N)
C
C *** ANALYZE ANOTHER DATA FILE?
C
  200 WRITE (*,956) DATAF
      READ (*,903) Q
      IF (Q .EQ. 'Y' .OR. Q .EQ. 'y') GO TO 50
      CLOSE (L4)
      CLOSE (L2)
      CLOSE (L1)
C
C *** PRINT VERSION NUMBER BEFORE ENDING
C
```

```
  300 IF (MODRES .GT. 2) WRITE (*,960) FNAM(MODRES)
      IF (MODLOG .EQ. 3 .OR. MODLOG .EQ. 4) WRITE (*,961) FNAM(MODLOG)
      WRITE (*,900) VERS
      STOP 1986
C
C *** ERROR MESSAGES
C
  500 WRITE (*,'('' UNABLE TO OPEN OPTIONS FILE'')')
      STOP 500
  501 WRITE (*,'('' UNABLE TO OPEN RESULTS FILE'')')
      STOP 501
  502 WRITE (*,'('' UNABLE TO OPEN SESSION.LOG FILE'')')
      STOP 502
  503 WRITE (*,'('' UNABLE TO OPEN PLOT.DAT FILE'')')
      STOP 503
C
C *** FORMAT STATEMENTS
C
  900 FORMAT (/'        NATIONAL BUREAU OF STANDARDS'/
     1' LOGISTIC CURVE FITTING PROGRAM, VERSION ',A8/)
  902 FORMAT (A)
  903 FORMAT (A1)
  904 FORMAT (I2)
  905 FORMAT (F10.4)
  910 FORMAT (' RECONFIGURE                    - Y/N? [N] ')
  911 FORMAT (' SELECT OPTIONS                 - Y/N? [N] ')
  912 FORMAT (' ENTER NAME OF RESULTS FILE      [RESULTS] ')
  940 FORMAT (' REFIT                          - Y/N? [N] ')
  941 FORMAT ('+REINITIALIZE PARAMETERS        - Y/N? [N] ')
  953 FORMAT ('+SAVE A PLOTTING FILE           - Y/N? [N] ')
  954 FORMAT (' ERROR IN PLOT ROUTINE, IRETCD = ',I4)
  955 FORMAT (/' REFIT DATA FOR ',A/)
  956 FORMAT (/' END OF ANALYSIS FOR DATA FILE: ',A//
     1'$ENTER ANOTHER FILE             - Y/N? [N] ')
  960 FORMAT (/' FINAL RESULTS CAN BE FOUND IN THE FILE: ',A)
  961 FORMAT (/' INTERMEDIATE RESULTS CAN BE FOUND IN THE FILE: ',A)
  992 FORMAT (' ANALYSIS OF DATA IN ',A/' DATE ',A/' TIME ',A/)
  993 FORMAT (/1X,A30/)
  994 FORMAT (/' BEGIN ANALYSIS AT ',A/' *** PLEASE WAIT ***'/)
  995 FORMAT (/' END ANALYSIS AT ',A/)
  996 FORMAT (/' START OF SESSION LOG FOR UNIT ',I2,
     1/' DATA FILE NAME: ',A/,' DATE ',A/,' TIME ',A/)
  997 FORMAT (/' CALL LGSTC AT ',A/)
  998 FORMAT (/' RETURN FROM LGSTC AT ',A/)
      END
```

```
C
C                    *********************
C                    *                   *
C                    *  SUBROUTINE OPTS  *
C                    *                   *
C                    *********************
C
C                       VERSION 5/20/86
C
C SUBROUTINE OPTS READS AND WRITES OPTIONS FILES OPENED BY THE MAIN
C PROGRAM AND ALLOWS THE SETTING OF OPTIONS THROUGH A QUESTION AND
C ANSWER SESSION WITH THE CONSOLE.  THE NEW OPTIONS MAY BE SAVED.
C SOME OPTIONS (BEGINNING AT STATEMENT 400) ARE AVAILABLE FOR CHANGING
C DURING THE COURSE OF THE EXECUTION OF THE MAIN PROGRAM.
C OTHER OPTIONS MAY BE CHANGED ONLY AT THE START OF THE PROGRAM.
C
      SUBROUTINE OPTS(NOPT)
      COMMON/CFGPARM/KTYPE,LOCK,IFPLOT,MODRES,MODLOG
      COMMON/IOPARM/L0,L1,L2,L3
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/CONTROL/IFINIT,INSEQ,NIT,NKOUT,YOUT,ITG(4),IFC(9),IFCIN(9)
      COMMON/DATA/N,X(200),Y(200),W(200),IFIO(200),NMARK,MARK(2)
      CHARACTER NAME(9)*2,Q*1,OPT(0:1)*1,OPT2(0:1)*1,OPT2L(0:1)
      CHARACTER MSG(4)*16
      CHARACTER FLINE*60,FD(0:8)*1
      DATA NAME/' A',' B','X0','D0','AS','BS','AQ','BQ',' Q'/
      DATA OPT/'N','Y'/
      DATA OPT2/'Y','N'/
      DATA OPT2L/'y','n'/
      DATA FD/'0','1','2','3','4','5','6','7','8'/
      DATA MSG/'CONVERGENCE TEST',
     1'INTERVAL TEST   ',
     2'UNCERTAINTY TEST',
     3'ASYMPTOTE TEST  '/
C
C *** IF  LOCK = 1, RESETTING OF OPTIONS WILL NOT BE ALLOWED DURING
C ***              SESSION
C
      GO TO (100,200,300,400) NOPT
C
C *** NOPT = 1, READ IN AN EXISTING OPTIONS FILE
C *** NOPT = 2, INITIATE A NEW OPTIONS FILE
C *** NOPT = 3, CHANGE AN EXISTING FILE
C *** NOPT = 4, CHANGE OPTIONS DURING PROGRAM EXECUTION
C
  100 READ (L3,904,END = 200,ERR = 200) KTYPE,LOCK,IFPLOT,MODRES,MODLOG
      READ (L3,904,END = 200,ERR = 200) (IFCIN(J), J = 1,9)
      READ (L3,904,END = 200,ERR = 200) (ITG(J), J = 1,4)
      READ (L3,905,END = 200,ERR = 200) NIT,NKOUT,YOUT
      READ (L3,*,END = 200,ERR = 200) (CIN(J), J = 1,9)
C
C *** IF THE FILE AND ALL ITS PARAMETERS ARE FOUND, RETURN.
C *** IF NOT SET DEFAULT VALUES, AND TREAT AS IF A NEW FILE.
```

```
C
      RETURN
C
C *** ENTER AT 200 IF NOPT = 2 OR IF READING OF OPTIONS FILE FAILED
C *** FIRST SET DEFAULT VALUES THEN ENTER QUESTION SESSION
C
  200 DO 210 J = 1,9
      CIN(J) = 0.0
  210 IFCIN(J) = 1
      DO 220 J = 1,4
  220 ITG(J) = 1
      KTYPE = 1
      LOCK = 0
      IFPLOT = 1
      MODRES = 1
      MODLOG = 1
      NIT = 11
      NKOUT = 0
      YOUT = 4.0
C
C                    CHANGE OPTIONS
C
C *** DEFAULT IS DISPLAYED IN [] AND IS CHANGED ONLY IF ENTRY
C *** IS OPPOSITE TO CURRENT SETTING OR IF A NUMBER IS ENTERED.
C *** CARRIAGE RETURN KEEPS CURRENT OPTION SETTING
C
C *** 1.  SELECT OUTPUT MODES:
C
  300 WRITE (*,911)
      WRITE (*,913) MODRES
      READ (*,902) IQ
      IF (IQ .GE. 1 .AND. IQ .LE. 5) MODRES = IQ
      WRITE (*,912)
      WRITE (*,913) MODLOG
      READ (*,902) IQ
      IF (IQ .GE. 1 .AND. IQ .LE. 5) MODLOG = IQ
C
C *** 2.  IDENTIFY DATA TYPE
C
      WRITE (*,914) KTYPE
      READ (*,902) IQ
      IF (IQ .GE. 1 .AND. IQ .LE. 3) KTYPE = IQ
C
C *** 3.  LOCK THE OPTIONS?
C
      WRITE (*,915) OPT(LOCK)
      READ (*,901) Q
      IF (Q .EQ. OPT2(LOCK) .OR. Q .EQ. OPT2L(LOCK)) LOCK = 1-LOCK
C
C *** 4.  AUTOMATICALLY SAVE DATA FILE FOR PLOTTING?
C
      WRITE (*,916) OPT(IFPLOT)
      READ (*,901) Q
      IF (Q .EQ. OPT2(IFPLOT) .OR. Q .EQ. OPT2L(IFPLOT))
     1IFPLOT = 1-IFPLOT
C
C *** 400 IS THE ENTRY POINT FOR CHANGING OPTIONS DURING THE SESSION
C *** IF WE ARE STILL SETTING UP THE INITIAL CONFIGURATION FILE, WE
C *** SKIP THE OPTION OF EDITING DATA.
C
C *** 5.  EDIT DATA?
C
  400 IFINIT = 1
      IF (NOPT .NE. 4) GO TO 410
      WRITE (*,921)
      READ (*,901) Q
      IF (Q .NE. 'Y' .AND. Q .NE. 'y') GO TO 410
      CALL SELECT
      IFINIT = INSEQ
C
C *** 6.  SELECT THE PARAMETERS TO BE VARIED.  SET THE VALUES OF
C ***     THOSE PARAMETERS THAT ARE NOT VARIED.
C
  410 WRITE (*,930)
      READ (*,901) Q
      IF (Q .NE. 'Y' .AND. Q .NE. 'y') GO TO 420
      DO 411 I = 1,9
      J = IFCIN(I)
      WRITE (*,931) NAME(I),OPT(J)
      READ (*,901) Q
      IF (Q .EQ. OPT2(J) .OR. Q .EQ. OPT2L(J)) IFCIN(I) = 1-IFCIN(I)
      IF (IFCIN(I) .EQ. 1) GO TO 411
      IF (I .GT. 4) GO TO 411
      ND = 3
      IF (CIN(I) .NE. 0.0) ND = INT(ABS(ALOG10(CIN(I))))+3
      IF (ND .LT. 3) ND = 3
      LS = 11-ND
      FLINE = '(''$          ENTER VALUE FOR '',A2,'//FD(LS)//
     1'X,''['',F'//FD(ND)//'.1,''] '')'
      WRITE (*,FLINE) NAME(I),CIN(I)
      READ (*,903) CVAL
      IF (CVAL .NE. 0.0) CIN(I) = CVAL
      IF (I .EQ. 4 .AND. CIN(I) .EQ. 0.0) IFCIN(I) = 1
  411 CONTINUE
C
C *** 7.  IGNORE POST-FITTING TESTS?
C
  420 WRITE (*,940)
      READ (*,901) Q
      IF (Q .NE. 'Y' .AND. Q .NE. 'y') GO TO 430
      DO 421, I = 1,4
      J = ITG(I)
      WRITE (*,941) MSG(I),OPT(J)
      READ (*,901) Q
  421 IF (Q .EQ. OPT2(J) .OR. Q .EQ. OPT2L(J)) ITG(I) = 1-ITG(I)

C
C *** 8.  REJECT OUTLIERS?
```

```
C
  430 J = 0
      IF (NKOUT .GT. 0) J = 1
      WRITE (*,950) OPT(J)
      READ (*,901) Q
      IF (Q .EQ. 'Y' .OR. Q .EQ. 'y') GO TO 431
      IF (Q .NE. 'N'.AND. Q .NE. 'n') GO TO 440
      NKOUT = 0
      YOUT = 4.0
      GO TO 440
  431 WRITE (*,951) YOUT
      READ (*,903) YOUT2
      IF (YOUT2 .NE. 0.0) YOUT = YOUT2
      WRITE (*,952) NKOUT
      READ (*,902) NKOUT2
      IF (NKOUT2 .NE. 0) NKOUT = NKOUT2
C
C *** 9.  CHANGE THE NUMBER OF ITERATIONS PER CYCLE?
C
  440 WRITE (*,960) NIT
      READ (*,902) IQ
      IF (IQ .NE. 0) NIT = IQ
C
C *** FINISHED WITH RECONFIGURATION.  IF NOPT = 4, RETURN
C *** OTHERWISE, GIVE THE OPTION OF SAVING THE FILE
C
  490 IF (NOPT .EQ. 4) RETURN
      WRITE (*,971)
      READ (*,901) Q
      IF (Q .NE. 'Y' .AND. Q .NE. 'y') RETURN
      REWIND (L3)
      WRITE (L3,904,ERR = 500) KTYPE,LOCK,IFPLOT,MODRES,MODLOG
      WRITE (L3,904,ERR = 500) (IFCIN(J), J = 1,9)
      WRITE (L3,904,ERR = 500) (ITG(J), J = 1,4)
      WRITE (L3,905,ERR = 500) NIT,NKOUT,YOUT
      WRITE (L3,*,ERR = 500) (CIN(J), J = 1,9)
      RETURN
C
C *** I/O ERROR MESSAGE
C
  500 WRITE (*,'('' UNABLE TO WRITE OPTIONS FILE'')')
      STOP 500
C
C *** FORMAT STATEMENTS
C
  901 FORMAT (A1)
  902 FORMAT (I2)
  903 FORMAT (F6.0)
  904 FORMAT (9I5)
  905 FORMAT (2I5,F6.2)
  911 FORMAT (' FINAL RESULTS OF ANALYSIS DIRECTED TO:'/)
  912 FORMAT ('+INTERMEDIATE RESULTS OF ANALYSIS DIRECTED TO:'/)
  913 FORMAT (5X,'[1] CONSOLE'/5X,'[2] PRINTER'/
     15X,'[3] RESULTS FILE'/5X,'[4] SESSION.LOG FILE'/
     25X,'[5] USER FILE'//
     3' WHICH? ENTER NUMBER 1 -> 5             [',I1,'] ')
  914 FORMAT ('+IDENTIFY DATA TYPE:'//5X,'[1] X AND Y'/
     15X,'[2] SEVERAL Y''S, TAKING RATIOS'/5X,'[3] X, Y, AND WEIGHTS'//
     2' WHICH? ENTER NUMBER 1 -> 3             [',I1,'] ')
  915 FORMAT ('+LOCK OPTIONS                         - Y/N? [',A,'] ')
  916 FORMAT ('+AUTOSAVE DATA FOR PLOTTING           - Y/N? [',A,'] ')
  921 FORMAT ('+EDIT DATA                            - Y/N? [N] ')
  930 FORMAT ('+SELECT PARAMETERS TO BE VARIED - Y/N? [N] ')
  931 FORMAT ('+     ALLOW ',A2,' TO VARY            - Y/N? [',A,'] ')
  932 FORMAT ('+          ENTER VALUE FOR ',A2,6X,'  [0.0] ')
  940 FORMAT ('+SELECT POST-FITTING TESTS?          - Y/N? [N] ')
  941 FORMAT ('+     INCLUDE ',A16,2X'- Y/N? [',A,'] ')
  950 FORMAT ('+IDENTIFY AND REJECT OUTLIERS        - Y/N? [',A,'] ')
  951 FORMAT ('+     OUTLIER REJECTION VALUE:       [',F4.2,'] ')
  952 FORMAT ('+     MAX. NO. OF RETRIES:             [',I1,'] ')
  960 FORMAT ('+ENTER NUMBER OF ITERATIONS/CYCLE:    [',I2,'] ')
  971 FORMAT (//' ^G*** FINISHED WITH CONFIGURATION OPTIONS ***'//
     1' SAVE THE OPTIONS AS DEFAULTS   - Y/N? [N] ')
      END
```

```
C                   **************************
C                   *                        *
C                   *   SUBROUTINE SELECT    *
C                   *                        *
C                   **************************
C
C                        VERSION 5/20/86
C
C DISPLAYS DATA AND ALLOWS USER TO SELECT DATA TO BE EXCLUDED FROM
C THE FIT EITHER AS SINGLE POINTS OR AS BLOCKS.
C ALLOWS THE USER TO DEFINE THE PRETRANSITION, TRANSITION, AND
C POSTTRANSITION REGIONS TO AID IN MAKING INITIAL ESTIMATES.
C
C THE FOLLOWING CALLS TO GRAPH LIBRARY ROUTINES ARE MADE:
C
C   0) SET THE SCREEN TO GRAPHICS MODE AND CLEAR THE SCREEN.
C      CALL AGRAF0(IRETCD)
C   1) DRAW THE AXES OF A GRAPH
C      CALL AGRAF1(IRETCD,IWIDTH,IAXES,XMIN,XMAX,YMIN,YMAX)
C   2) DISPLAY A TEXT STRING.
C      CALL AGRAF2(IRETCD,ISROW,ISCOL,ISTYPE,STRING)
C   3) PLOT A CURVE SEGMENT
C      CALL AGRAF3(IRETCD,IPTYPE,N,X,Y)
C   4) PRINT THE GRAPHICS SCREEN.
C      CALL AGRAF4(IRETCD)
C   5) SET THE SCREEN TO TEXT MODE AND CLEAR THE SCREEN.
C      CALL AGRAF5(IRETCD)
C   8) RETURN A LOGICAL VALUE DEPENDING ON KEYBOARD STATUS.
C      LVALUE = AGRAF8(IKTYPE,KEEP)
C
      SUBROUTINE SELECT
      COMMON/DATA/N,X(200),Y(200),W(200),IFIO(200),NMARK,MARK(2)
      CHARACTER FL(0:9),FX*5,FY*5,BL*63,FXY*30,CURSOR*80,FMARK*80
      CHARACTER STATUS(-1:1)*5,B*80
      DIMENSION XC(2),YC(2),XM(2),XMARK(2),YMARK(2)
      DIMENSION KEYS(11),ICONC(-1:1),ICOND(-1:1)
      LOGICAL LVALUE,AGRAF8
      DATA FL/'0','1','2','3','4','5','6','7','8','9'/
      DATA ICONC,ICOND/6922,0,6666,11018,0,2314/
      DATA STATUS/'  OUT',' ','   IN'/
      DATA B/' '/
      DATA NKEY,KEYS/11,19200,19712,21248,20992,16128,22528,16384
     1,22784,20224,13,17408/
      ICUR=256*179+10
      IMARK=256*186+10
      KHELP1 = 1
      MARGIN = 8
      NMARK = 0
      XMIN = X(1)
      XMAX = X(1)
      DO 5 I = 1,N
      XMIN = AMIN1(X(I),XMIN)
    5 XMAX = AMAX1(X(I),XMAX)
      EXT = (XMAX-XMIN)/20.
      XMAX = XMAX+EXT
      XMIN = XMIN-EXT
      XMAXIN = XMAX
      XMININ = XMIN
C
C *** ON THE BASIS OF THE MAGNITUDE OF X AND Y AND THE AVERAGE
C *** SEPARATION BETWEEN X AND Y VALUES, DETERMINE THE PROPER FORMAT FOR
C *** PRINTING X AND Y WITH AN ADEQUATE NUMBER OF SIGNIFICANT FIGURES.
C
      XFORM = AMAX1(ABS(XMAXIN),ABS(XMININ))
      L=INT(ALOG10(XFORM)+1.)
      IF (L .LT. 0) L=0
      DX=(XMAXIN-XMININ)/N
      LD=INT(2.0-ALOG10(DX))
      IF (LD .GT. 4) LD=4
      IF (LD .LT. 0) LD=0
      LX=L+LD+2
      IF (LX .GT. 9) LX=9
      FX=',F'//FL(LX)//'.'//FL(LD)
      Y1=ABS(Y(1))
      YN=ABS(Y(N))
      YFORM = AMAX1(Y1,YN)
      L=INT(ALOG10(YFORM)+1.)
      IF (L .LT. 0) L=0
      DY=(Y(N)-Y(1))/N
      DY=ABS(DY)
      LD=INT(2.0-ALOG10(DY))
      IF (LD .GT. 4) LD=4
      IF (LD .LT. 0) LD=0
      LY=L+LD+2
      IF (LY .GT. 9) LY=9
      FY=',F'//FL(LY)//'.'//FL(LD)
      FXY ='''  X = '''//FX//','' Y = '''//FY//',A5)'
C
C *** FIND THE FIRST AND LAST POINTS INCLUDED IN THE DISPLAY.  START
C *** HERE EACH TIME SCALE OF SCREEN IS CHANGED
C
   15 DO 16 I=1,N
   16 IF (X(I) .GE. XMIN) GO TO 17
   17 NLO=I
      DO 18 I=1,N
      NHI=N-I+1
   18 IF (X(NHI) .LE. XMAX) GO TO 19
C
C *** FIND THE MINIMUM AND MAXIMUM VALUES OF Y FOR SCALING DISPLAY
C
   19 YMIN = Y(NLO)
      YMAX = Y(NLO)
      NLP1 = NLO+1
      DO 20 I = NLP1,NHI
      YMIN = AMIN1(Y(I),YMIN)
   20 YMAX = AMAX1(Y(I),YMAX)
      YCUR=(YMAX-YMIN)/24
      EXT = (YMAX-YMIN)/20.
```

```
      YMAX = YMAX+EXT
      YMIN = YMIN-EXT
C
C *** TURN ON THE GRAPHICS SCREEN
C
      CALL AGRAF0(IRETCD)
C
C *** PRINT THE AXES
C
      CALL AGRAF1(IRETCD,IW,0,XMIN,XMAX,YMIN,YMAX)
C
C *** PLOT THE DATA POINT BY POINT.  DATA INCLUDED IN THE FIT
C *** SIGNIFIED BY o AND DATA EXCLUDED BY +.
C
      DO 30 I = 1,N
      IL = IFIO(I)
      IDAT = ICOND(IL)
      DATAY = Y(I)
      DATAX = X(I)
   30 CALL AGRAF3(IRETCD,IDAT,1,DATAX,DATAY)
C
C *** SET THE INITIAL VALUE OF THE CURSOR AND PLOT IT ON THE SCREEN
C
      NCUR = NLO
      IDIR = 1
      IDEL = -1
      IDARW = ICONC(1)
      DATAX=X(NCUR)
      DATAY=Y(NCUR)
      XCD=DATAX
      YCD=DATAY+2.0*YCUR
      IF (YCD .GT. YMAX) YCD=DATAY-2.0*YCUR
      XC(1)=DATAX
      XC(2)=DATAX
      YC(1)=DATAY+YCUR
      YC(2)=DATAY-YCUR
      CALL AGRAF3(IRETCD,ICUR,2,XC,YC)
      CALL AGRAF3(IRETCD,IDARW,1,XCD,YCD)
C
C *** SET THE INITIAL FORMAT FOR PRINTING CURSOR INFORMATION AND PRINT
C *** THE CURSOR DATA WITH LEADING BLANKS ON THE BOTTOM LINE OF THE
C *** SCREEN.
C
   70 ISTAT=IFIO(NCUR)
      CURSOR='('//FXY
      WRITE (BL,CURSOR) X(NCUR),Y(NCUR),STATUS(ISTAT)
      LB=63-LNBX(BL)
      BL=B(1:LB)//BL
      LTAB=INDEX(BL,'X')-2+MARGIN
      CALL AGRAF2(IRETCD,24,MARGIN,0,BL)
C
C *** PRINT THE HELP LINE AT THE TOP OF THE SCREEN AND START THE EDITING
C *** SESSION
C
      KHELP = KHELP1
      CALL HELP(KHELP)
C
C *** RETURN TO 80 FOLLOWING PROCESSING OF EACH KEY PRESS
C
   80 ISTAT = IFIO(NCUR)
      WRITE (BL,CURSOR) X(NCUR),Y(NCUR),STATUS(ISTAT)
      CALL AGRAF2(IRETCD,24,LTAB,0,BL)
   90 CALL KEY(KVAL,KEYS,NKEY)
C
C READ THE KEYBOARD FOR SPECIAL KEYS.  RETURN WITH THE INDEX 1 THROUGH
C NKEY FOR KVAL.  THE KEYS CORRESPONDING TO THE VALUES OF KVAL ARE:
C
C  KVAL  KEY         VALUE   ACTION                                   GO TO
C   1.   [<--]       19200  MOVE CURSOR LEFT                           100
C   2.   [-->]       19712  MOVE CURSOR RIGHT                          110
C   3.   DEL:        21248  EXCLUDE DATUM AND MOVE CURSOR              120
C   4.   INS:        20992  INCLUDE DATUM AND MOVE CURSOR              130
C   5.   F5:         16128  SET MARKER OR ZOOM                         300
C   6.   SHIFT-F5:   22528  CLEAR MARKER OR RESTORE INITIAL SCALE      350
C   7.   F6:         16384  DELETE BETWEEN MARKERS                     400
C   8.   SHIFT-F6:   22784  INSERT BETWEEN MARKERS                     400
C   9.   END:        20224  END EDITING SESSION                        600
C  10.   ENTER:         13  END EDITING SESSION                        600
C  11.   F10:        17408  PRINT SCREEN                               500
C
      GO TO (100,110,120,130,300,350,400,400,600,600,500) KVAL
  100 IF (IDIR .EQ. -1) GO TO 200
      CALL AGRAF3(IRETCD,IDARW,1,XCD,YCD)
      IDIR = -1
      IDARW=ICONC(-1)
      CALL AGRAF3(IRETCD,IDARW,1,XCD,YCD)
      GO TO 90
  110 IF (IDIR .EQ. 1) GO TO 200
      CALL AGRAF3(IRETCD,IDARW,1,XCD,YCD)
      IDIR = 1
      IDARW=ICONC(1)
      CALL AGRAF3(IRETCD,IDARW,1,XCD,YCD)
      GO TO 90
  120 IDEL=-1
      GO TO 200
  130 IDEL=+1
  200 IF (KVAL .LE. 2) GO TO 210
      IF (IDEL .EQ. IFIO(NCUR)) GO TO 210
        IDAT=ICOND(-IDEL)
        CALL AGRAF3(IRETCD,IDAT,1,DATAX,DATAY)
        IFIO(NCUR)=IDEL
        IDAT=ICOND(IDEL)
        CALL AGRAF3(IRETCD,IDAT,1,DATAX,DATAY)
  210 CALL AGRAF3(IRETCD,ICUR,2,XC,YC)
      CALL AGRAF3(IRETCD,IDARW,1,XCD,YCD)
      NCUR=NCUR+IDIR
      IF (NCUR .LT. NLO) NCUR = NHI
      IF (NCUR .GT. NHI) NCUR = NLO
```

```
      DATAX=X(NCUR)
      DATAY=Y(NCUR)
      XCD=DATAX
      YCD=DATAY+2.0*YCUR
      IF (YCD .GT. YMAX) YCD=DATAY-2.0*YCUR
      XC(1)=DATAX
      XC(2)=DATAX
      YC(1)=DATAY+YCUR
      YC(2)=DATAY-YCUR
      CALL AGRAF3(IRETCD,ICUR,2,XC,YC)
      CALL AGRAF3(IRETCD,IDARW,1,XCD,YCD)
      LVALUE=AGRAF8(KEYS(KVAL),0)
      IF (.NOT.LVALUE) GO TO 80
  230 LVALUE=AGRAF8(0,1)
      IF (LVALUE) GO TO 230
      GO TO 200
C
C *** PUT A MARKER ON THE SCREEN AT THE POSITION OF THE CURSOR.  IF
C *** THE SECOND MARKER IS AS THE SAME POSITION AS THE FIRST, THEN
C *** ZOOM TO HALF THE SCREEN CONTAINING THE MARKER.  IF TWO MARKERS
C *** ARE IN PLACE, ZOOM TO THE REGION BETWEEN THE TWO MARKERS.  IF
C *** MARKERS ARE IN EFFECT UPON RETURNING TO THE CALLING PROGRAM, THE
C *** SUBROUTINE ESTIM8 USES THE MARKERS TO MAKE THE FIRST ESTIMATES
C *** OF THE PARAMETERS.
C
  300 IF (NMARK .LT. 2) THEN
        CALL AGRAF3(IRETCD,ICUR,2,XC,YC)
        CALL AGRAF3(IRETCD,IMARK,2,XC,YC)
        CALL AGRAF3(IRETCD,ICUR,2,XC,YC)
        NMARK = NMARK+1
        MARK(NMARK) = NCUR
        XM(NMARK)=DATAX
        IF (NMARK .EQ. 2) THEN
C
C *** IF BOTH MARKERS ARE NOW ON AND ARE THE SAME WE ZOOM TO HALF THE
C *** REGION WITH THE MARKER AS CLOSE TO THE CENTER AS POSSIBLE
C
          IF (MARK(1) .EQ. MARK(2)) THEN
            XRANGE=(X(NHI)-X(NLO))/2.0
            XLO=XM(1)-XRANGE/2.0
            XHI=XM(1)+XRANGE/2.0
            IF (XLO .LT. XMIN) THEN
              XMAX=XMIN+XRANGE
            ELSE IF (XHI .GT. XMAX) THEN
              XMIN=XMAX-XRANGE
            ELSE
              XMIN=XLO
              XMAX=XHI
            END IF
            NMARK=0
            KHELP1=4
            GO TO 15
C
C *** OTHERWISE WE ENSURE THAT MARK(1) IS LESS THAN MARK(2) AND PRINT
C *** THE APPROPRIATE HELP MESSAGE AND CONTINUE
C
          ELSE IF (MARK(1) .GT. MARK(2)) THEN
            XM(2) = XM(1)
            M=MARK(2)
            XM(1) =  X(M)
            MARK(2) = MARK(1)
            MARK(1) = M
          END IF
          FMARK = '(''MARK1 = '''//FX//',''  MARK2 = '''//FX//','
     1//FXY
          WRITE (BL,FMARK) XM(1),XM(2),DATAX,DATAY,STATUS(ISTAT)
          KHELP = 3
C
C *** IF NMARK IS NOW LESS THAN 2 WE HAVE JUST MADE THE FIRST MARK.
C
        ELSE
          FMARK = '(''MARK1 = '''//FX//','//FXY
          WRITE (BL,FMARK) XM(1),DATAX,DATAY,STATUS(ISTAT)
          KHELP = 2
        END IF
        LB = 63-LNBX(BL)
        BL = B(1:LB)//BL
        CALL HELP(KHELP)
        CALL AGRAF2(IRETCD,24,MARGIN,0,BL)
        GO TO 80
      END IF
C
C *** IF BOTH MARKERS ARE ON AND ARE DIFFERENT, WE ZOOM TO THE REGION
C *** BETWEEN THE TWO
C
      XMIN=XM(1)
      XMAX=XM(2)
      NMARK=0
      KHELP1=4
      GO TO 15
C
C *** ERASE LAST MARKER OR UNZOOM
C
  350 IF (NMARK .GT. 0) THEN
        NM=MARK(NMARK)
        XMARK(1)=X(NM)
        XMARK(2)=X(NM)
        YMARK(1)=Y(NM)+YCUR
        YMARK(2)=Y(NM)-YCUR
        CALL AGRAF3(IRETCD,IMARK,2,XMARK,YMARK)
        NMARK=NMARK-1
        IF (NMARK .EQ. 1) THEN
          FMARK = '(''MARK1 = '''//FX//','//FXY
          WRITE (BL,FMARK) XM(1),DATAX,DATAY,STATUS(ISTAT)
          KHELP = 2
          LB = 63-LNBX(BL)
          BL = B(1:LB)//BL
          CALL AGRAF2(IRETCD,24,MARGIN,0,BL)
```

```
        CALL HELP(KHELP)
        GO TO 80
      END IF
      GO TO 70
    END IF
    IF (XMIN .EQ. XMININ .AND. XMAX .EQ. XMAXIN) GO TO 80
    XMIN=XMININ
    XMAX=XMAXIN
    KHELP=1
    GO TO 15
C
C *** IF TWO MARKERS HAVE BEEN SET, DELETE (OR INSERT IF KVAL = 8)
C *** ALL POINTS FROM THE FIT FALLING ON AND BETWEEN THE MARKERS
C
  400 IF (NMARK .LT. 2) GO TO 80
    IO=-1
    IF (KVAL .EQ. 8) IO=1
    N1=MARK(1)
    N2=MARK(2)
    DO 401 I=N1,N2
    IL = IFIO(I)
    IDAT = ICOND(IL)
    DATAX=X(I)
    DATAY=Y(I)
    CALL AGRAF3(IRETCD,IDAT,1,DATAX,DATAY)
    IFIO(I)=IO
    IDAT = ICOND(IO)
  401 CALL AGRAF3(IRETCD,IDAT,1,DATAX,DATAY)
C
C *** NOW ERASE BOTH MARKERS
C
    DO 402 I=1,2
    NM=MARK(I)
    XMARK(1)=X(NM)
    XMARK(2)=X(NM)
    YMARK(1)=Y(NM)+YCUR
    YMARK(2)=Y(NM)-YCUR
  402 CALL AGRAF3(IRETCD,IMARK,2,XMARK,YMARK)
    NMARK=0
    GO TO 70
C
C *** PRINT THE SCREEN.  IF PRINTER NOT ON, NOT ATTACHED, OR OTHERWISE
C *** NOT COMPATIBLE, PRINT A MESSAGE TO THAT EFFECT IN THE HELP LINE
C *** BEFORE CONTINUING
C
  500 CALL AGRAF4(IRETCD)
    IF (IRETCD .EQ. 0) GO TO 80
    CALL HELP(5)
  501 LVALUE=AGRAF8(0,1)
    IF (.NOT.LVALUE) GO TO 501
    CALL HELP(KHELP)
    GO TO 80
C
C *** END THE EDITING SESSION
C
  600 CALL AGRAF5(IRETCD)
    RETURN
    END
```

```
C                         **********************
C                         *                    *
C                         *   SUBROUTINE KEY   *
C                         *                    *
C                         **********************
C
C                              VERSION 5/20/86
C
C READ THE KEYBOARD FOR SPECIAL KEYS.  RETURN WITH THE INDEX 1 THROUGH
C N FOR KVAL.  DOES NOT RETURN UNTIL ONE OF THE KEYS INDICATED BY THE
C VALUES OF KEYCODE
C

      SUBROUTINE KEY(KVAL,KEYCODE,NUMKEYS)
      LOGICAL LVALUE,AGRAF8
      DIMENSION KEYCODE(20)
C
C *** START LOOKING FOR KEYSTROKES
C
   50 LVALUE = AGRAF8(0,0)
      IF (.NOT. LVALUE) GO TO 50
C
C *** ONCE WE HAVE FOUND ONE, TEST IT FOR RECOGNITION
C
      DO 52 K=1,NUMKEYS
      LVALUE=AGRAF8(KEYCODE(K),0)
      IF (LVALUE) THEN
        LVALUE=AGRAF8(0,1)
        KVAL=K
        RETURN
      END IF
   52 CONTINUE
C
C *** IF NOT RECOGNIZED, DISCARD IT AND START OVER
C
      LVALUE=AGRAF8(0,1)
      GO TO 50
      END
```

```
C
C                         ***********************
C                         *                     *
C                         *   SUBROUTINE HELP   *
C                         *                     *
C                         ***********************
C
C                              VERSION 5/13/86
C
C PRINT A HELP MESSAGE ON THE TOP LINE OF A GRAPH DEPENDING ON THE VALUE
C OF THE ARGUMENT KHELP.
C
      SUBROUTINE HELP(KHELP)
      CHARACTER HMSG(5)*80,B*80,TLINE*80
      DATA HMSG/'[<-> MOVE CURSOR] [DEL DELETE POINT] [INS INCLUDE POINT
     1] [F5 MARK] [END]',
     2'[<-> MOVE CURSOR] [DEL] [INS] [F5 MARK] [SHIFT-F5 UNMARK] [END]',
     3'[F5 ZOOM] [SHIFT-F5 UNMARK] [F6 DELETE BLOCK] [SHIFT-F6 INCLUDE B
     4LOCK] [END]',
     5'[<-> MOVE CURSOR] [DEL] [INS] [F5 MARK] [SHIFT-F5 UNZOOM] [END]',
     6'PRINTER NOT READY, NOT AVAILABLE, OR NOT COMPATIBLE - ANY KEY TO
     7CONTINUE'/
      DATA B/' '/
      IF (KHELP .EQ. 5) WRITE (*,'(''  ^G'')')
      TLINE=HMSG(KHELP)
      LH=80-LNBX(TLINE)
      TLINE=B(1:LH)//TLINE
      CALL AGRAF2(IRETCD,0,0,0,TLINE)
      RETURN
      END
```

```
      INTEGER FUNCTION LNBX (STRNG)
*   Returns index of last nonblank character in STRNG or zero if
*   STRNG is blank.
*
*      DECLARATIONS
*   Function value:
*      LNBX   - (output) INTEGER
*   Argument:
*      STRNG  - (input)
       CHARACTER*(*) STRNG
*   Globals:
       INTRINSIC LEN
*   Local:
       INTEGER i
*
*      PROCEDURE
       IF (STRNG .EQ. ' ') THEN
         LNBX = 0
       ELSE
*
         DO 1000 i=LEN (STRNG),1,-1
*
           IF (STRNG(i:i) .NE. ' ') THEN
             LNBX = i
             RETURN
           ENDIF
*
1000     CONTINUE
*
         PRINT *, 'LNBX function error; call S Stewart 301-921-2905'
       ENDIF
       RETURN
       END
```

```
C
C                          **************************
C                          *                        *
C                          *    SUBROUTINE RDDATA   *
C                          *                        *
C                          **************************
C
C                                VERSION 5/20/86
C
C READS IN A DATA FILE AND STOPS IF AN ERROR IS ENCOUNTERED
C THREE TYPES OF DATA CAN BE TREATED:
C
C   I.    X,Y VALUES ONLY
C   II.   X WITH SEVERAL VALUES OF Y WITH THE PROVISION OF ANALYZING
C         RATIOS OF Y VALUES
C   III.  X,Y VALUES ACCOMPANIED BY WEIGHTS
C
C IN ADDITION, THIS SUBROUTINE:
C
C        CALCULATES WEIGHTS FOR DATA
C        CALCULATES XMAX, XMIN, YMAX, YMIN
C        TESTS ORDERING OF DATA
C        ESTIMATES A LINEAR RELATIONSHIP BETWEEN THE WEIGHTS AND
C        THE VALUES OF Y:  W = W0 + W1*Y FOR THE PURPOSE OF ESTIMATING
C        THE UNCERTAINTY OF A HYPOTHETICAL DATUM AT THE CENTER OF THE
C        TRANSITION INTERVAL.
C
      SUBROUTINE RDDATA (DATAF,TITLEF)
      COMMON/CFGPARM/KTYPE,LOCK,IFPLOT,MODRES,MODLOG
      COMMON/IOPARM/L0,L1,L2,L3
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/CONTROL/IFINIT,INSEQ,NIT,NKOUT,YOUT,ITG(4),IFC(9),IFCIN(9)
      COMMON/MISC/W0,W1,KIT,NCYC
      COMMON/DATA/N,X(200),Y(200),W(200),IFIO(200),NMARK,MARK(2)
      COMMON/SCRATCH/Z(11,209)
      CHARACTER DATAF*24,TITLEF*30,YTLN*72,YTL(9)*8,Q*1
C
C *** ENTER NAME OF DATA FILE.  IF NO ENTRY, ASSUME PREVIOUS DATA FILE
C *** IF NO PREVIOUS DATA FILE, END.
C
      WRITE (*,901)
      READ (*,902) TITLEF
      IF (TITLEF .NE. ' ') DATAF=TITLEF
      IF (DATAF .EQ. ' ') GO TO 99
      OPEN (L3,FILE=DATAF,STATUS='OLD',ERR=99)
C
C *** NEXT READ TITLE LINE IN FILE
C *** DISCARD LEADING CARRIAGE RETURNS
C *** TEST FIRST NON-EMPTY LINE FOR LETTERS
C *** IF NO LETTERS ARE ENCOUNTERED, ASSUME THE FIRST NON-EMPTY LINE IS
C *** DATA.  REWIND FILE AND SUBSTITUTE THE FILE NAME FOR THE TITLE
C
      NBL = -1
   10 NBL = NBL+1
```

```
      READ (L3,902,ERR=99,END=99) TITLEF
      L=LNBX(TITLEF)
      IF (L .EQ. 0) GO TO 10
      LT = LEN(TITLEF)
      DO 20 I=1,LT
      Q=TITLEF(I:I)
      IF (Q .GE. 'A' .AND. Q .LE. 'Z') GO TO 30
   20 IF (Q .GE. 'a' .AND. Q .LE. 'z') GO TO 30
      TITLEF = 'FILE: '//DATAF
      LT = LEN(TITLEF)
      REWIND (L3)
      IF (NBL .EQ. 0) GO TO 30
      DO 25 I=1,NBL
   25 READ (L3,902,ERR=99,END=99) Q
   30 GO TO (100,200,300) KTYPE
   99 WRITE (*,999) DATAF,TITLEF,I
      CLOSE (L3)
      STOP 99
C
C                            TYPE I DATA
C
C *** READ THE VALUES OF X(I) AND Y(I).  THE DATA SHOULD BE
C *** IN ORDER OF INCREASING X FOR PROPER INITIALIZATION OF THE
C *** PARAMETERS IN SUBROUTINE LGSTC
C
  100 I = 1
  110 READ (L3,*,END=130,ERR=120) X(I),Y(I)
      I=I+1
      GO TO 110
  120 WRITE (*,999) DATAF,TITLEF,I
      CLOSE (L3)
      STOP 120
  130 CLOSE (L3)
      N=I-1
C
C *** SET WEIGHTS EQUAL TO UNITY
C *** IFIO(I) = 1 CAUSES DATUM TO BE INCLUDED IN FIT
C
      DO 140 I=1,N
  140 W(I)=1.0
      GO TO 400
C
C                            TYPE II DATA
C
C     DATA LIST CONTAINS SEVERAL Y VALUES FOR EACH X VALUE.  Y VECTORS
C     CAN BE FIT INDIVIDUALLY OR RATIOS OF THE Y VECTORS CAN BE FIT.  IF
C     THE RATIOS ARE FIT, THE WEIGHTS ARE SET ACCORDINGLY.  THE MAXIMUM
C     NUMBER OF VALUES OF Y FOR EACH VALUE OF X IS 9.
C
  200 READ (L3,902,END=220,ERR=220) YTLN
C
C *** YTLN CONTAINS THE TITLES OF EACH OF THE DATA COLUMNS THAT WILL
C *** BE READ.  EACH TITLE CAN BE AS LONG AS EIGHT CHARACTERS BUT CAN
C *** HAVE NO IMBEDDED BLANKS.  BLANKS ARE USED TO SEPARATE THE FIELDS.
C
      NTL=0
  210 LTLN=LEN(YTLN)
      IF (YTLN .EQ. ' ') GO TO 212
      L=INDEX(YTLN,' ')
      IF (L .GT. 1) GO TO 211
      YTLN=YTLN(2:LTLN)
      GO TO 210
  211 IF (NTL .EQ. 9) GO TO 212
      NTL=NTL+1
      YTL(NTL)=YTLN(1:L-1)
      YTLN=YTLN(L+1:LTLN)
      GO TO 210
  212 IF (NTL .EQ. 1) GO TO 100
      I = 1
C
C *** THE NUMBER TO ENTRIES ON THE TITLE LINE DETERMINES THE NUMBER OF
C *** Y VECTORS THAT ARE TO BE READ
C
  215 READ (L3,*,END=230,ERR=220) X(I),(Z(J,I),J=1,NTL)
      I=I+1
      GO TO 215
  220 WRITE (*,999) DATAF,TITLEF,I
      CLOSE (L3)
      STOP 220
  230 CLOSE (L3)
      N=I-1
      WRITE (*,920)
      DO 240 I=1,NTL
  240 WRITE (*,921) I, YTL(I)
  241 WRITE (*,922) NTL
      READ (*,923) NUMER
      IF (NUMER .EQ. 0) GO TO 241
      WRITE (*,924) NTL
      READ (*,923) NDENOM
      IF (NDENOM .EQ. 0) GO TO 260
      IF (NDENOM .EQ. NUMER) GO TO 260
C
C *** NOW ADJUST TITLE TO CONTAIN THE IDENTIFICATION OF THE RATIO
C
      LN=INDEX(YTL(NUMER),' ')-1
      LD=INDEX(YTL(NDENOM),' ')-1
      L = LNBX(TITLEF)
      IF (L+LN+LD+2 .GT. LT) L=LT-LN-LD-2
      TITLEF=TITLEF(1:L)//' '//YTL(NUMER)(1:LN)//'/'//YTL(NDENOM)(1:LD)
      DO 250 I=1,N
      Y(I)=Z(NUMER,I)/Z(NDENOM,I)
  250 W(I)=Z(NDENOM,I)*Z(NDENOM,I)/(Y(I)*Y(I)+1.0)
      GO TO 400
  260 LN=INDEX(YTL(NUMER),' ')-1
      L=LNBX(TITLEF)
      IF (L+LN+1 .GT. LT) L=LT-LN-1
      TITLEF=TITLEF(1:L)//' '//YTL(NUMER)(1:LN)
      DO 261 I=1,N
```

```
      Y(I)=Z(NUMER,I)
  261 W(I)=1.0
      GO TO 400
C
C                          TYPE III DATA
C
C THE DATA LIST CONTAINS THE WEIGHT FOR EACH DATUM
C
  300 I = 1
  310 READ (L3,*,END=330,ERR=320) X(I),Y(I),W(I)
      I=I+1
      GO TO 310
  320 WRITE (*,999) DATAF,TITLEF,I
      CLOSE (L3)
      STOP 320
  330 CLOSE (L3)
      N=I-1
C
C *** A LINEAR RELATIONSHIP BETWEEN Y AND W IS ESTIMATED: W = W0 + W1*Y
C *** THIS RELATIONSHIP IS USED TO ESTIMATE THE WEIGHT OF A MEASUREMENT
C *** AT THE MIDPOINT OF THE TRANSITION INTERVAL WHICH IS USED TO
C *** ESTIMATE THE AVERAGE UNCERTAINTY THROUGHOUT THE TRANSITION
C *** INTERVAL WHICH IS USED TO ESTIMATE THE WIDTH OF THE INTERVAL.
C *** THE WIDTH OF THE INTERVAL IS USED TO PLACE AN UPPER BOUND ON D0
C *** WHEN D0 CAN NOT BE DETERMINED BY THE LEAST SQUARES FIT.
C *** THE ORDERING OF THE DATA IS TESTED AND A WARNING IS PRINTED IF
C *** ANY VALUES OF X ARE OUT OF SEQUENCE.
C
C *** IFIO(I) = 1 IMPLIES THE DATUM IS TO BE INCLUDED IN THE LEAST
C *** SQUARES FIT OF THE DATA.  THE VALUES OF IFIO(I) ARE CHANGED
C *** BY THE USER DURING DATA EDITING AND BY LGSTC DURING OUTLIER
C *** IDENTIFICATION.
C
  400 IFIO(1) = 1
      XMAX = X(1)
      NOORDR = 0
      DO 410 I = 2,N
      IFIO(I) = 1
      IF (X(I) .LT. XMAX) NOORDR = NOORDR+1
  410 XMAX = MAX(XMAX,X(I))
      IF (NOORDR .GT. 0) WRITE(*,915) NOORDR
      CALL XYLINE(Y,W,W,IFIO,1,N,0.0,0,W0,W1)
      RETURN
C
C *** FORMAT STATEMENTS
C
  901 FORMAT ('$ENTER DATA FILE NAME: ')
  902 FORMAT (A)
  915 FORMAT(/' ^GCAUTION! ',I5,' DATA VALUES OUT OF SEQUENCE'/
     1' X VALUES SHOULD BE IN NUMERICAL ORDER'/)
  920 FORMAT (' SELECT Y VALUES FOR WHICH RATIOS ARE TO BE FIT. '/)
  921 FORMAT (5X,'[',I1,']  ',A8)
  922 FORMAT (/' SELECT NUMERATOR,   ENTER INDEX 1 THROUGH ',I1,': ')
  923 FORMAT (I5)
  924 FORMAT ('$SELECT DENOMINATOR, ENTER INDEX 1 THROUGH ',I1,': ')
  999 FORMAT (/' UNABLE TO OPEN OR INTERPRET DATA FILE'/
     15X,'DATA FILE NAME: ',A/5X,'DATA TITLE: ',A/
     25X,'LINE NUMBER:',I5/5X,'CHECK DATA FILE FOR PROPER FORMAT')
      END
```

```
C                       *************************
C                       *                       *
C                       *   SUBROUTINE LGSTC    *
C                       *                       *
C                       *************************
C
C                         VERSION 5/20/86
C
C THE FOLLOWING VARIABLES, CONTAINED IN THE COMMON BLOCKS, MUST BE
C PROVIDED BY THE CALLING PROGRAM:
C
C N:        THE NUMBER OF DATA TO BE ANALYZED.
C C(I):     THE INITIAL VALUES OF THE COEFFICIENTS.  IF NOT PROVIDED,
C           (SEE IFC(I) BELOW) THEY ARE CALCULATED BY LGSTC.
C X(I):     THE VALUES OF THE INDEPENDENT VARIABLE.  X(I) MUST BE
C           IN ORDER OF INCREASING MAGNITUDE.
C Y(I):     THE VALUES OF THE DEPENDENT VARIABLE
C W(I):     THE WEIGHT OF EACH DATUM.  EACH SHOULD BE PROPORTIONAL TO
C           THE INVERSE SQUARE OF THE UNCERTAINTY IN Y(I).  IF ALL
C           VALUES OF Y(I) HAVE THE SAME UNCERTAINTY, THEN ALL VALUES OF
C           W(I) MAY BE SET TO UNITY.
C IFIO(I):  AN INDICATOR, = 1 IF Y(I) IS TO BE INCLUDED IN THE FIT,
C           = -1 IF IT IS NOT.
C IFC(I):   INDICATOR DETERMINING WHICH PARAMETERS ARE TO BE VARIED AND
C           HOW THE INITIAL VALUES ARE TO BE SET.  IF IFC(I) = 0, THE
C           CORRESPONDING PARAMETER, C(I), IS NOT CALCULATED BY THE
C           LEAST SQUARES FIT BUT IS HELD FIXED AT SOME PREDETERMINED
C           VALUE.  IF IFC(I) = 1, C(I) IS CALCULATED BY THE LEAST
C           SQUARES FIT.  LGSTC MAY, IN THE COURSE OF THE ANALYSIS,
C           MAKE THE DETERMINATION THAT CERTAIN PARAMETERS SHOULD OR
C           SHOULD NOT BE CALCULATED FROM THE DATA AND WILL ALTER THE
C           VALUE OF IFC(I).  THEREFORE, THE INITIAL VALUES OF IFC(I)
C           ARE SAVED FOR FUTURE USE IN IFCIN(I). IFQ PERFORMS A
C           FUNCTION INTERMEDIATE BETWEEN IFC AND IFCIN FOR C(9) SINCE
C           C(9) CAN BE HELD FIXED DURING THE ITERATION PROCESS OR
C           DURING THE POSTFITTING PROCESS AND A MEANS IS NEEDED TO
C           DISTINGUISH BETWEEN THE TWO.  LGSTC RETURNS THE FINAL STATES
C           OF IFC(I) TO THE CALLING PROGRAM.
C
C NKOUT:    IF NOT 0, OUTLIERS WILL BE IDENTIFIED AND THE DATA WILL BE
C           REFIT WITH THE OUTLIERS DROPPED FROM THE FIT.  THE OUTLIER
C           REJECTION WILL BE PERFORMED NKOUT TIMES OR UNTIL NO MORE
C           OUTLIERS ARE FOUND.
C YOUT:     THE NUMBER OF STANDARD DEVIATIONS BY WHICH YOBS-YCALC
C           MUST DIFFER FROM 0 BEFORE A DATUM IS CONSIDERED TO BE AN
C           OUTLIER.  YOUT IS NEEDED ONLY IF NKOUT IS NOT 0.
C NIT:      THE NUMBER OF ITERATIONS TO BE PERFORMED.  A PRIME NUMBER
C           SHOULD BE USED TO PICK UP OSCILLATIONS BETWEEN TWO
C           SOLUTIONS.  11 IS USUALLY ADEQUATE.
C ITG(I):   "IF TEST GO" FLAGS, A SET OF FLAGS INDICATING IF ANY OF THE
C           POST-FITTING TESTS ARE TO BE IGNORED.
C           I = 1, CONVERGENCE TEST
C           I = 2, INTERVAL TEST
C           I = 3, UNCERTAINTY TEST
C           I = 4, ASYMPTOTE TEST
C           IF ITG(I) = 1, THE DATA ARE REFIT WHEN THE TEST IS FAILED
C           IF ITG(I) = 0, ONLY THE WARNING IS PRINTED
C IFINIT:   CONTROLS ESTIMATION OF INITIAL VALUES OF THE PARAMETERS
C           IN CONJUNCTION WITH KTYPE AND INSEQ
C
C THE VALUES OF THE FOLLOWING VARIABLES ARE RETURNED BY LGSTC FOR USE
C BY THE CALLING PROGRAM:
C
C C(I):      THE VALUES OF THE PARAMETERS GIVEN BY THE LAST ITERATION
C BS(I):     THE STANDARD DEVIATIONS OF THE VALUES OF THE PARAMETERS
C COR(I):    THE CORRECTIONS TO THE PARAMETERS FROM THE LAST ITERATION
C VCV(I,J):  THE ELEMENTS OF THE VARIANCE-COVARIANCE MATRIX
C STD:       THE STANDARD DEVIATION OF THE FIT
C REL:       THE RELATIVE STANDARD DEVIATION (IN % RELATIVE TO (A+B)/2).
C GMDT:      THE VALUE OF THE GRAM DETERMINANT, THE DETERMINANT OF THE
C            LEAST SQUARES MATRIX.
C
C THE DIMENSIONS TO BE GIVEN TO SOME OF THE VARIABLES DEPEND ON THE
C MAXIMUM NUMBER OF OBSERVATIONS TO BE FIT.  IF THE MAXIMUM NUMBER
C IS N, THEN N IS THE DIMENSION OF X,Y,W, AND IFIO, AND (11, N+9) ARE
C THE DIMENSIONS OF Z.
C
C OTHER PARAMETERS HELD IN COMMON BLOCKS ARE SHARED WITH VARIOUS
C SUBROUTINES OR ARE VALUES THAT SHOULD NOT BE LOST FROM ONE CALL TO
C LGSTC TO ANOTHER.  COMMON/SCRATCH/ IS USED BY THE LEAST SQUARES FIT,
C BUT CAN BE OVERWRITTEN BY OTHER SUBROUTINES (SUCH AS PLOTTING
C ROUTINES) SINCE THE DATA IN IT ARE REGENERATED WITH EACH USE.
C
      SUBROUTINE LGSTC
      COMMON/IOPARM/L0,L1,L2,L3
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/CONTROL/IFINIT,INSEQ,NIT,NKOUT,YOUT,ITG(4),IFC(9),IFCIN(9)
      COMMON/STAT/COR(9),BS(9),VCV(9,9),NZ,MZ,STD,REL,GMDT,ESTSTD
      COMMON/MISC/W0,W1,KIT,NCYC
      COMMON/DATA/N,X(200),Y(200),W(200),IFIO(200),NMARK,MARK(2)
      COMMON/SCRATCH/Z(11,209)
      CHARACTER F2(0:10)*1,DLINE*15,FLINE*66,FB1*1,FB4*4,FN4*4
      LOGICAL CONVRG
      DOUBLE PRECISION UCALC
      DATA F2/'0','1','2','3','4','5','6','7','8','9','10'/
      DATA FB1,FB4,FN4/' ','    ',' N '/
C
C                         *** SEGMENT 100 ***
C
C *** ENTRY INTO LGSTC.
C
C *** KIT COUNTS THE NUMBER OF ITERATIONS FOR REPEATED CYCLES BUT NOT
C ***    REFITS
C *** NCYC COUNTS THE NUMBER OF CYCLES INCLUDING REFITS WHEN TESTS ARE
C ***    FAILED.
C *** KOUT COUNTS OUTLIER REJECTION CYCLES TO BE PERFORMED
C *** FNSTD IS THE NUMBER OF STANDARD DEVIATIONS BY WHICH AN OBSERVED
C ***    VALUE OF Y MUST DIFFER FROM EACH OF THE CALCULATED ASYMPTOTES
```

```
C ***    OF Y IN ORDER FOR A DATUM TO BE CONSIDERED TO FALL WITHIN THE
C ***    TRANSITION INTERVAL.  THE CALCULATED TRANSITION INTERVAL IS A
C ***    FUNCTION OF THE VALUE OF FNSTD.
C
  100 FNSTD = 2.0
      IFT3 = 0
      NCYC = 0
      KOUT = 0
      INSEQ = 1
      DO 110 I=1,9
  110 IFC(I)=IFCIN(I)
      IFQ = IFC(9)
C
C *** ESTIM8(KTYPE,SIGN)  CALCULATES INITIAL VALUES OF THE PARAMETERS
C *** IF IFINIT = 1 AND INSEQ =1, THE INITIAL VALUES OF C(3) AND C(4)
C *** ARE OBTAINED BY LOCATING THE MIDPOINT OF THE TRANSITION AT THE
C *** AVERAGE VALUE OF Y, WHERE THE AVERAGE VALUE OF Y IS TAKEN AS THE
C *** AVERAGE OF THE INITIAL AND FINAL VALUES OF Y.  IF INSEQ=2, THE
C *** VALUE OF C(3) IS LOCATED AT THE MIDPOINT OF THE BIGGEST JUMP IN
C *** THE VALUE OF Y.
C *** IF IFINIT=3, THEN THE CURRENT VALUES OF THE PARAMETERS ARE TAKEN
C *** AS THE INITIAL VALUES AND ESTIM8 MERELY SETS THE STARTING VALUES
C *** C(I) EQUAL TO THE INITIAL VALUES CIN(I).
C *** CONTROL IS RETURNED TO 200 FOR REFITTING.
C *** SIGN = 1 IF A>B, = -1 IF A<B

  200 KTYPE = IFINIT
      IF (NMARK .EQ. 2) KTYPE = 2
      CALL ESTIM8(KTYPE,SIGN)
      KIT=0
C
C                        *** SEGMENT 400 ***
C
C *** BECAUSE Y IS A NON-LINEAR FUNCTION OF THE PARAMETERS, THE
C *** PARAMETERS MUST BE CALCULATED ITERATIVELY.  SEGMENT 400 PERFORMS
C *** A CYCLE OF NIT ITERATIONS.  IN EACH ITERATION, THE NUMBER OF
C *** DATA, NIN, FALLING MORE THAN FNSTD STANDARD DEVIATIONS ABOVE THE
C *** LOWER ASYMPTOTE AND BELOW THE UPPER ASYMPTOTE IS COUNTED.  BASED
C *** ON THIS NUMBER, C(9), C(4), AND/OR C(3) MAY BE HELD FIXED FOR
C *** THAT ITERATION.  FOLLOWING EACH ITERATION, THE VALUES OF C(I) ARE
C *** UPDATED SO THAT DERIVATIVES OF Y WITH RESPECT TO THE PARAMETERS
C *** ARE ALWAYS EVALUATED WITH THE CURRENT VALUES OF THE PARAMETERS.
C
      WRITE (*,'('' ITERATION NUMBER:'')')
  400 DO 460 NI = 1,NIT
      WRITE (*,'(''&'',I3)') NI
      KIT = KIT+1
C
C *** ESTIMATE THE STANDARD DEVIATION USING THE CURRENT VALUES OF THE
C *** PARAMETERS.  THIS ESTIMATE IS USED TO COUNT THE NUMBER OF DATA
C *** POINTS FALLING IN THE TRANSITION INTERVAL.  MZ = NUMBER OF
C *** PARAMETERS BEING EVALUATED.  NZ = NUMBER OF DATA BEING FIT.
C *** DERIV CALCULATES THE VALUE OF Y = YCALC FOR A GIVEN VALUE OF
C *** X = XVAL AND ALSO THE DERIVATIVES OF Y WITH RESPECT TO THE
C *** PARAMETERS C(I).
C
      MZ = 0
      DO 410 I = 1,9
  410 MZ = MZ+IFC(I)
      SUM = 0.0
      NZ = 0
      DO 420 I = 1,N
      IF (IFIO(I) .LT. 0) GO TO 420
      NZ = NZ+1
      XVAL = X(I)
      CALL DERIV
      SUM = SUM+W(I)*(Y(I)-YCALC)*(Y(I)-YCALC)
  420 CONTINUE
      NDF = NZ-MZ
      ESTSTD = SQRT(SUM/NDF)
C
C *** UYAV IS THE ESTIMATE OF THE UNCERTAINTY IN A HYPOTHETICAL
C *** MEASUREMENT OF Y(I) AT X0, I.E. AT Y(I) = YAV.  FX IS THE
C *** TRANSITION INTERVAL FACTOR, IS CALCULATED FROM THE VALUE OF A-B
C *** AND THE STANDARD DEVIATION OF THE FIT BUT IS NEVER ALLOWED TO BE
C *** LESS THAN 2.0.  THE TRANSITION INTERVAL IS DEFINED TO BE FX*DO
C *** AND IS THAT INTERVAL IN WHICH A MEASURED VALUE OF Y IS EXPECTED
C *** TO DIFFER FROM ITS NEAREST ASYMPTOTE BY MORE THAN FNSTD STANDARD
C *** DEVIATIONS.
C *** IF DO CANNOT BE CALCULATED, ITS VALUE IS SET DEPENDING ON THE
C *** NUMBER OF DATA POINTS IN THE TRANSITION INTERVAL AND THEIR
C *** AVERAGE SEPARATION, XSEP.  IF NONE, DO IS SET AT XSEP/(2.0*FX).
C *** IF ONE, DO IS SET AT XSEP/FX.  AT EACH ITERATION, XSEP IS
C *** RECALCULATED TO BE THE AVERAGE DATA SEPARATION IN THE TRANSITION
C *** INTERVAL.  2*XLIM APPROXIMATES THE WIDTH OF THE INTERVAL AND IS
C *** USED AS A WINDOW FOR COUNTING DATA IN THE INTERVAL
C
      YAV = (C(1)+C(2))/2.0
      WAV = W0+W1*YAV
      UYAV = FNSTD*ESTSTD/SQRT(WAV)
      FX = ABS(C(2)-C(1))/UYAV
      TEST = EXP(1.0)+1
      IF (FX .LT. TEST) FX = TEST
      FX = 2.0*ALOG(FX-1.0)
      XLIM = CIN(4)*FX/2.0
C
C *** NOW DEFINE THE TRANSITION INTERVAL
C
C *** NIN COUNTS THE NUMBER OF DATA POINTS IN THE TRANSITION INTERVAL
C *** EXCLUDE 1ST AND LAST POINTS FROM SEARCH AND TEST ONLY THOSE
C *** DATA WITHIN, JUST BEFORE, AND JUST AFTER THE BOUNDARIES OF THE
C *** TRANSITION INTERVAL AS DEFINED BY CIN(4)*FX = 2.0*XLIM
C
C *** UNC IS THE ESTIMATED UNCERTAINTY IN A PARTICULAR VALUE OF Y(I)
C *** ASYMP IS THE VALUE OF AN ASYMPTOTE AT X = X(I)
C
C *** NMP COUNTS THE NUMBER OF DATA FALLING WITHIN ONE STANDARD
C *** STANDARD DEVIATION OF YAV.  IT IS USED IN SEGMENT 500 TO TEST THE
```

```
C *** HYPOTHESIS THAT NO DATA FALL WITHIN THE TRANSITION INTERVAL.
C
C *** NA IS THE LAST POINT IN THE PRE-TRANSITION ASYMPTOTE
C *** NB IS THE FIRST POINT IN THE POST-TRANSITION ASYMPTOTE
C
C *** MP IS THE VALUE OF I FOR THE DATUM JUST BELOW YAV AND IS USED TO
C *** CALCULATE XO WHEN NIN=0.
C
      NIN = 0
      NMP = 0
      NA = 0
      DO 430 I = 2,N-1
      IP = I+1
      IM = I-1
      IF (IFIO(I) .LT. 0) GO TO 430
      TEST = C(3)-X(IP)
      IF (TEST .GT. XLIM) GO TO 430
      TEST = X(IM)-C(3)
      IF (TEST .GT. XLIM) GO TO 431
      IF (Y(IP) .EQ. YAV) MP=I
      TEST = Y(IP)-YAV
      IF (TEST .NE. 0) TEST = (Y(I)-YAV)/TEST
      IF (TEST .LE. 0) MP=I
      TEST = ABS(Y(I)-YAV)
      UNC = ESTSTD/SQRT(W(I))
      IF (TEST .LT. UNC) NMP = NMP+1
      XVAL = X(I)
      CALL DERIV
      UNC = FNSTD*UNC
      ASYMP = C(1)+(C(5)+C(7)*(X(I)-C(3)))*(X(I)-C(3))
      TEST = SIGN*(ASYMP-Y(I))
      IF (TEST .LT. UNC) GO TO 430
      ASYMP = C(2)+(C(6)+C(8)*(X(I)-C(3)))*(X(I)-C(3))
      TEST = SIGN*(Y(I)-ASYMP)
      IF (TEST .LT. UNC) GO TO 430
      IF (NA .EQ. 0) NA = I
      NB = I
      NIN = NIN+1
  430 CONTINUE
  431 IF (NIN .GT. 0) GO TO 432
      NA = MP+1
      NB = MP
  432 NA = NA-1
      NB = NB+1
      XSEP = (X(NB)-X(NA))/(NB-NA)
      IF (NIN .GT. 0) NIN = NIN-IFT3
C
C *** IFT3 IS USUALLY 0 BUT IS GREATER THAN 0 IF THE UNCERTAINTY TEST
C *** FAILS IN SEGMENT 500.  IF GREATER THAN 0 IT CAUSES MORE
C *** TRANSITION INTERVAL PARAMETERS (Q, DO, XO) TO BE HELD CONSTANT
C *** THAN THE VALUE OF NIN WOULD DICTATE
C
C *** NOW SET THE VALUES OF IFC(3,4,9) AND C(3,4,9) IF NIN < 3.
C
      IFC(3) = IFCIN(3)
      IFC(4) = IFCIN(4)
      IFC(9) = IFQ
      NCASE = NIN+1
      IF (NCASE .GT. 4) NCASE = 4
      GO TO (433,434,437,440) NCASE
  433 IFC(3) = 0
      IF (IFCIN(3) .EQ. 1) C(3) = (X(MP+1)+X(MP))/2.0
  434 IFC(4) = 0
      IF (IFCIN(4) .EQ. 1) C(4) = XSEP/((2-NIN)*FX)
  437 IFC(9) = 0
      C(9) = 0
C
C *** SUBROUTINE FIX SETS UP THE VECTORS FOR THE LEAST SQUARES FIT,
C *** KEEPING TRACK OF WHICH PARAMETERS ARE VARIED AND WHICH ARE HELD
C *** FIXED AND WHICH DATA ARE INCLUDED AND EXCLUDED FROM THE FIT.  FIX
C *** CALLS ORTHO WHICH PERFORMS THE LEAST SQUARES FIT AFTER WHICH THE
C *** VALUES OF THE PARAMATERS ARE CORRECTED AND THE PROCEDURE REPEATED
C
  440 CALL FIX
      CONVRG=.TRUE.
      DO 450 I = 1,9
      TEST=3
      IF (BS(I) .NE. 0.0) TEST=ALOG10(ABS(BS(I)/COR(I)))
      IF (TEST .LT. 3) CONVRG = .FALSE.
  450 C(I) = C(I)+COR(I)
C
C *** IF THE CONVERGENCE TEST PASSES AND IF NO PARAMETERS ARE BEING
C *** HELD CONSTANT BY LGSTC, STOP ITERATING AND CONTINUE
C
      ICTEST=IFCIN(3)+IFCIN(4)+IFQ-IFC(3)-IFC(4)-IFC(9)
      IF (ICTEST .EQ. 0 .AND. CONVRG) GO TO 461
  460 CONTINUE
  461 NCYC = NCYC+1
C
C *** ASSIGN UNCERTAINTIES TO XO AND DO IF THEY WERE NOT DETERMINABLE
C *** BUT WERE HELD FIXED AT MOST PROBABLE VALUES
C
      IF (IFC(3) .NE. IFCIN(3)) BS(3) = FX*C(4)
      IF (IFC(4) .NE. IFCIN(4)) BS(4) = C(4)


C
C *** PRINT PARAMETERS AND ASSOCIATED STATISTICAL INFORMATION.
C *** IF, ON RETURNING, THE VALUE OF NOCON IS GREATER THAN 0,
C *** THE CONVERGENCE TEST FAILS.
C *** OUTPUT GOES TO UNIT L2
C
      CALL EPARM(L2,NOCON,NIN)
C
C                            *** SEGMENT 500 ***
C
C *** PERFORM POST FITTING TESTS.  IF THE TESTS FAIL AND THE
C *** CORRESPONDING VALUE OF ITG(I) = 1, THE DATA WILL BE REFIT HOLDING
C *** ONE OR MORE OF THE PARAMETERS CONSTANT.  IF ITG(I) = 0, ONLY THE
```

```
C *** WARNING IS PRINTED.
C
C *** TEST 1, THE CONVERGENCE TEST.  IF FAILED, THE DATA ARE REFIT
C *** USING THE BIGGEST JUMP IN Y AS THE INITIAL ESTIMATE OF C(3).
C *** IF THE CONVERGENCE FAILS A SECOND TIME, OR IF INSEQ WAS ALREADY
C *** SET = 2 BY ESTIM8, REFIT HOLDING THE SLOPES AS AND BS = 0.
C *** IF AS AND BS ARE ALREADY 0, THE DATA ARE REFIT HOLDING Q = 0.
C
C *** TEST 2, THE INTERVAL TEST.  IF NIN = 0 AND NMP > 0 THE TEST FAILS
C *** AND THE DATA ARE REFIT AS IN THE CASE OF THE FAILURE OF TEST 1.
C
C *** TEST 3, THE UNCERTAINTY TEST.  THIS TEST IS PERFORMED TO
C *** DETERMINE IF THE NUMBER OF PARAMETERS VARIED IN THE TRANSITION
C *** INTERVAL WAS WARRANTED.  IF IT FAILS, IT FAILS WHEN THE NUMBER
C *** OF DATA IN THE INTERVAL, NIN, IS EQUAL TO THE NUMBER OF
C *** INTERVAL PARAMETERS BEING VARIED.  IF YCALC - TWICE ITS STANDARD
C *** DEVIATION IS LESS THAN THE LOW ASYMPTOTE - TWICE ITS
C *** STANDARD DEVIATION OR IF YCALC + TWICE ITS STANDARD DEVIATION IS
C *** GREATER THAN THE HIGH ASYMPTOTE + TWICE ITS STANDARD
C *** DEVIATION, THEN THE UNCERTAINTY IN YCALC IS CONSIDERED TO BE
C *** UNACCEPTABLY LARGE AND THE TEST FAILS.  THE TEST IS PERFORMED IN
C *** THE INTERVAL FROM X0 - 5*D0 TO X0 + 5*D0 IN STEPS OF D0/10.
C *** IF NIN IS EQUAL TO 3 OR LESS, THE DATA ARE REFIT HOLDING ONE MORE
C *** INTERVAL PARAMETER (Q, D0, AND X0) CONSTANT.  IF NIN IS GREATER
C *** THAN 3, THE UNCERTAINTY TEST IS NOT PERFORMED.  IFT3 IS A COUNTER
C *** FOR REDUCING THE NUMBER OF INTERVAL PARAMETERS VARIED.
C *** ON REFITTING, THE REFITTING BEGINS WITH THE CURRENT VALUES OF THE
C *** PARAMETERS.
C
C *** TEST 4, THE ASYMPTOTE TEST.  THE NUMBER OF DATA LYING WITHIN TWO
C *** STANDARD DEVIATIONS OF EACH ASYMPTOTE IS COUNTED.  IF MORE THAN
C *** FOUR, THAT ASYMPTOTE TEST PASSES.  IF FOUR OR LESS, THE STANDARD
C *** DEVIATION OF AS (OR BS) IS COMPARED WITH THE MAGNITUDE OF AS (OR
C *** BS).  IF AS IS LESS THAN THREE TIMES ITS STANDARD DEVIATION, THE
C *** ASYMPTOTE TEST FAILS AND THE FIT IS REDONE HOLDING AS AND AQ = 0.
C *** SIMILARLY FOR BS AND BQ.
C
C
C *** CONVERGENCE TEST
C
  500 IF (NOCON .NE. 0) THEN
        WRITE (L2,950)
        IF (ITG(1) .NE. 1) GO TO 510
        IF (INSEQ .EQ. 1) THEN
          INSEQ = 2
          GO TO 200
        END IF
        INSEQ = 1
        GO TO 511
      END IF
C
C *** INTERVAL TEST
C
  510 IF (NIN .GT. 0) GO TO 520
      IF (NMP .EQ. 0) GO TO 520
      WRITE (L2,951) NMP
      IF (ITG(2) .NE. 1) GO TO 520
C
C *** CONVERGENCE AND INTERVAL TEST FAILURE INVOKE THE SAME PROTOCOL
C
  511 IF (IFC(5) .EQ. 0 .AND. IFC(6) .EQ. 0) GO TO 512
      IFC(5) = 0
      IFC(6) = 0
      IFC(7) = 0
      IFC(8) = 0
      IFC(9) = IFQ
      WRITE (L2,952)
      GO TO 200
  512 IF (IFQ .EQ. 0) GO TO 520
      IFQ = 0
      WRITE (L2,953)
      GO TO 200
C
C *** UNCERTAINTY TEST
C
C *** FIND THE MAXIMUM VALUE OF YCALC + TWICE ITS UNCERTAINTY
C *** AND THE MINIMUM VALUE OF YCALC - TWICE ITS UNCERTAINTY
C *** THIS TEST AND THE ASYMPTOTE TEST ARE NOT MEANINGFUL AND
C *** THEREFORE NOT PERFORMED IF C(1) OR C(2) ARE NOT BEING VARIED.
C *** THIS TEST IS ALSO NOT MEANINGFUL AND THEREFORE NOT PERFORMED
C *** IF NIN IS GREATER THAN THE NUMBER OF TRANSITION INTERVAL
C *** PARAMETERS BEING VARIED.
C
  520 IF (IFC(1) .EQ. 0 .OR. IFC(2) .EQ. 0) GO TO 700
      NINVAR = IFCIN(3)+IFCIN(4)+IFQ
      IF (NIN .GT. NINVAR) GO TO 530
      IF (NIN-IFT3 .LE. 0) GO TO 530
      NUNC = 0
      UMIN = (C(1)+C(2))/2.0
      UMAX = UMIN
      DO 523 I = -50,50
      XVAL = C(3)+C(4)*I/10.0
      CALL DERIV
      UCALC = 0.0
      DO 521 J = 1,9
      DO 521 K = 1,9
  521 UCALC = UCALC+G(J)*VCV(J,K)*G(K)
      UTEST = YCALC-2.0*DSQRT(UCALC)
      IF (UTEST .GT. UMIN) GO TO 522
      UMIN = UTEST
      IMIN = I
  522 UTEST = YCALC+2.0*DSQRT(UCALC)
      IF (UTEST .LT. UMAX) GO TO 523
      UMAX = UTEST
      IMAX = I
  523 CONTINUE
C
C *** COMPARE THE MAXIMUM WITH THE VALUE OF THE ASYMPTOTE + TWICE ITS
```

```
C *** STANDARD DEVIATION AND THE MINIMUM WITH THE VALUE OF THE
C *** ASYMPTOTE - TWICE ITS STANDARD DEVIATION.  USE IMIN TO DETERMINE
C *** WHICH ASYMPTOTE IS THE UPPER AND WHICH IS THE LOWER.
C
      IF (IMIN .GT. 0) GO TO 524
      IL = 1
      JL = 5
      IU = 2
      JU = 6
      GO TO 525
  524 IL = 2
      JL = 6
      IU = 1
      JU = 5
  525 XAS = C(4)*IMIN/10.0
      ASYMP = C(IL)+C(JL)*XAS
      UNC = VCV(IL,IL)+XAS*(2.0*VCV(IL,JL)+XAS*VCV(JL,JL))
      YTEST = ASYMP-2.0*SQRT(UNC)
      IF (YTEST .GT. UMIN) NUNC = NUNC+1
      XAS = C(4)*IMAX/10.0
      ASYMP = C(IU)+C(JU)*XAS
      UNC = VCV(IU,IU)+XAS*(2.0*VCV(IU,JU)+XAS*VCV(JU,JU))
      YTEST = ASYMP+2.0*SQRT(UNC)
      IF (YTEST .LT. UMAX) NUNC = NUNC+1
      IF (NUNC .EQ. 0) GO TO 530
      WRITE (L2,955)
      IF (ITG(3) .NE. 1) GO TO 530
  527 WRITE (L2,956)
      IFT3 = IFT3+1
      GO TO 400
C
C *** ASYMPTOTE TEST
C
  530 IF (NIN .EQ. 0) GO TO 700
      IF (IFC(5) .EQ. 0 .AND. IFC(6) .EQ. 0) GO TO 700
      NL = 0
      NU = 0
      IFT4 = 0
      DO 531 I=1,N
      XVAL=X(I)
      CALL DERIV
      ASLIM = FNSTD*STD/SQRT(W(I))
      TEST = ABS(YCALC - C(1) - (C(5)+C(7)*(X(I)-C(3)))*(X(I)-C(3)))
      IF (TEST .LT. ASLIM) NL=NL+1
      TEST = ABS(YCALC - C(2) - (C(6)+C(8)*(X(I)-C(3)))*(X(I)-C(3)))
  531 IF (TEST .LT. ASLIM) NU=NU+1
      IF (IFC(5) .EQ. 0) GO TO 538
      NS5 = INT(ABS(C(5)/BS(5)))
      IF (NL .GT. 4) GO TO 538
      IF (NS5 .GE. 3) GO TO 538
      WRITE (L2,957) NL,NS5
      IF (ITG(4) .NE. 1) GO TO 538
      IFC(5) = 0
      IFC(7) = 0
      IFT4 = IFT4+1
  538 IF (IFC(6) .EQ. 0) GO TO 539
      NS6 = INT(ABS(C(6)/BS(6)))
      IF (NU .GT. 4) GO TO 539
      IF (NS6 .GE. 3) GO TO 539
      WRITE (L2,958) NU,NS6
      IF (ITG(4) .NE. 1) GO TO 539
      IFC(6) = 0
      IFC(8) = 0
      IFT4 = IFT4+1
  539 IF (IFT4 .EQ. 0) GO TO 700
      WRITE (L2,959)
      GO TO 200
C
C                            *** SEGMENT 700 ***
C
C *** TEST FOR OUTLIERS AND/OR PRINT THE VALUES OF THE DATA ALONG
C *** WITH THEIR ASSOCIATED STATISTICS.
C *** THE TESTING OF OUTLIERS IS CONTROLLED BY NOUT, NKOUT, AND YOUT.
C *** IF NKOUT = 0 OR IF KOUT = NKOUT, NO FURTHER OUTLIERS ARE
C *** DROPPED, THOUGH THEY WILL BE FLAGGED, AND THE DATA ARE PRINTED.
C *** IF NKOUT > 0, AND KOUT < NKOUT, OUTLIERS, THAT IS, DATA FOR
C *** WHICH THE STANDARDIZED RESIDUALS ARE GREATER THAN YOUT,
C *** ARE IDENTIFIED.  IF IDENTIFIED, THE ANALYSIS IS REPEATED,
C *** STARTING AT STATEMENT 200, WITH THE IDENTIFIED OUTLIERS DROPPED
C *** FROM THE LEAST SQUARES FIT AND ANALYZED APART FROM THE OTHER
C *** DATA.
C *** NOUT COUNTS THE NUMBER FOUND.   IF KOUT < NKOUT, OUTLIERS
C *** IDENTIFIED THUS FAR ARE PRINTED IN THE INTERMEDIATE RESULTS FILE.
C *** IF NONE ARE FOUND, KOUT IS SET EQUAL TO NKOUT AND CONTROL IS
C *** RETURNED TO STATEMENT 700 FOR PRINTING OF ALL THE DATA IN THE
C *** FINAL RESULTS FILE.
C
  700 NOUT = 0
      NOTE = 0
      LY=6-INT(ALOG10(STD)+5)
      IF (LY .LT. 0) LY=0
      IF (LY .GT. 5) LY=5
      LX=7-INT(ALOG10(ABS(XSEP))+5)
      IF (LX .LT. 0) LX=0
      IF (LX .GT. 5) LX=5
      DLINE='F8.'//F2(LX)//',3F9.'//F2(LY)//',F6.1'
      FLINE='(I4,'//DLINE//')'
      IF (KOUT .EQ. NKOUT) WRITE (L1,970)FB1,FB1
      DO 750 I = 1,N
      XVAL = X(I)
      CALL DERIV
C
C *** CALCULATE THE VARIANCE OF A CALCULATED VALUE OF Y FROM THE
C *** VARIANCE-COVARIANCE MATRIX, VCV, RETURNED BY ORTHO.  UOBS IS THE
C *** VARIANCE OF A MEASURED Y(I).
C
      UCALC = 0.0
      DO 710 J = 1,9
```

```
      DO 710 K = 1,9
  710 UCALC = UCALC + G(J)*VCV(J,K)*G(K)
      UOBS = STD*STD/W(I)
C
C *** CALCULATE THE STANDARDIZED RESIDUALS, TT.  NOTE: THE CALCULATION
C *** DEPENDS ON WHETHER A MEASURED VALUE WAS INCLUDED IN THE FIT OR
C *** NOT.  VALUES OF TT GREATER THAN 4 MAY BE OUTLIERS.  FOR ILL
C *** CONDITIONED POINTS, TT IS SET TO 99 AS A WARNING.  IT SIGNALS
C *** THAT THE PARAMETERS OF THE FIT MAY BE POORLY DETERMINED.
C
      UTEST = UOBS - IFIO(I)*UCALC
      DIFF = Y(I)-YCALC
      TT=99.
      IF (UTEST .GT. 0) TT=DIFF/SQRT(UTEST)
      UOBS = SQRT(UOBS)
C
C *** IF KOUT<NKOUT WE TEST FOR OUTLIERS.
C *** DO NOT COUNT ILL-CONDITIONED POINTS, AS THEY REFLECT
C *** INSTABILITY IN THE FIT AND ARE NOT NECESSARILY OUTLIERS.
C
      IF (KOUT .GE. NKOUT) GO TO 720
      IF (ABS(TT) .LT. YOUT) GO TO 750
      IF (ABS(TT) .GT. 98.) GO TO 750
      IF (NOTE .GT. 0) GO TO 715
      WRITE (L2, 971)
      WRITE (L2,970) FN4,FB4
      NOTE=NOTE+1
  715 WRITE (L2,FLINE) I,X(I),Y(I),YCALC,DIFF,TT
      IF (IFIO(I) .EQ. 1) NOUT = NOUT+1
      IFIO(I) = -1
      GO TO 750
C
C *** AFTER ALL OUTLIERS HAVE BEEN IDENTIFIED, KOUT IS SET EQUAL TO
C *** NKOUT (IF NOT ALREADY THERE) AND THE ENTIRE DATA SET WITH
C *** ASSOCIATED STATISTICS IS PRINTED.
C *** OMA (FOR OBSERVED MINUS ASYMPTOTE) IS THE DIFFERENCE BETWEEN AN
C *** OBSERVED VALUE FOR Y AND ITS NEAREST ASYMPTOTE DIVIDED BY THE
C *** UNCERTAINTY IN Y.  ONLY VALUES OF OMA WITHIN THE TRANSITION
C *** INTERVAL, AS DEFINED BY NA AND NB, ARE PRINTED.
C
  720 IF (IFIO(I) .EQ. 1) GO TO 721
      NOTE=NOTE+1
      FLINE='('//DLINE//','' *'')'
      GO TO 740
  721 IF (ABS(TT) .LT. YOUT) GO TO 722
      IF (ABS(TT) .GT. 98.) GO TO 722
      NOUT=NOUT+1
      FLINE='('//DLINE//','' **'')'
      GO TO 740
  722 IF (I .LE. NA .OR. I .GE. NB) GO TO 730
      OMA1 = ABS(Y(I)-C(1)-(C(5)+C(7)*(X(I)-C(3)))*(X(I)-C(3)))
      OMA2 = ABS(Y(I)-C(2)-(C(6)+C(8)*(X(I)-C(3)))*(X(I)-C(3)))
      OMA = MIN(OMA1,OMA2)/UOBS
      KOMA = OMA
      FLINE='('//DLINE//',I4)'
      WRITE (L1,FLINE) X(I),Y(I),YCALC,DIFF,TT,KOMA
      GO TO 750
  730 FLINE='('//DLINE//')'
  740 WRITE (L1,FLINE) X(I),Y(I),YCALC,DIFF,TT
  750 CONTINUE
      IF (KOUT .GE. NKOUT) GO TO 800
      IF (NOUT .GT. 0) GO TO 760
      KOUT=NKOUT
      GO TO 700
  760 KOUT=KOUT+1
      WRITE (L2,'(/'' REPEAT, EXCLUDING ABOVE DATA FROM ANALYSIS''/)')
      DO 761 I=1,9
  761 IFC(I)=IFCIN(I)
      IFQ=IFC(9)
      IFT3=0
      GO TO 200
C
C *** WE REACH STATEMENT 800 WHEN FINISHED WITH THE ANALYSIS.  BEFORE
C *** RETURNING TO THE MAIN PROGRAM, THE PARAMETERS AND THEIR
C *** ASSOCIATED STATISTICS ARE PRINTED IN THE FINAL RESULTS FILE.
C
  800 IF (NOTE .GT. 0) WRITE (L1,'(/'' *  NOT INCLUDED IN THE FIT'')')
      IF (NOUT .GT. 0) WRITE (L1,'(/'' ** POSSIBLE OUTLIER'')')
      CALL EPARM(L1,NOCON,NIN)
      RETURN
C
C *** FORMAT STATEMENTS
C
  950 FORMAT (' CONVERGENCE TEST FAILED')
  951 FORMAT (' NMP =',I3,'.  INTERVAL TEST FAILED')
  952 FORMAT (' REFIT HOLDING AS AND BS = 0'/)
  953 FORMAT (' REFIT HOLDING Q = 0'/)
  955 FORMAT (' UNCERTAINTY TEST FAILED')
  956 FORMAT (' REFIT HOLDING ONE MORE'/
     1' INTERVAL PARAMETER CONSTANT'/)
  957 FORMAT (' NL =',I2,' & NS5 =',I2,'.  LOW ASYMPTOTE TEST FAILED')
  958 FORMAT (' NU =',I2,' & NS6 =',I2,
     1'.  HIGH ASYMPTOTE TEST FAILED')
  959 FORMAT (' REFIT HOLDING SLOPE(S) = 0'/)
  970 FORMAT(A,' X(OBS)  Y(OBS)   Y(CALC)    DIFF   STANDARD'/
     1A,36X'RESIDUALS'/)
  971 FORMAT(' THE FOLLOWING HAVE BEEN IDENTIFIED AS OUTLIERS'/)
      END
```

```
C
C                ***********************************
C                *                                 *
C                *   SUBROUTINE ESTIM8(ICASE,SIGN) *
C                *                                 *
C                ***********************************
C
C                         VERSION 5/20/86
C
C CALCULATES INITIAL VALUES OF THE PARAMETERS FROM THE DATA ACCORDING
C TO THE VALUE OF ICASE.  IF ICASE = 1, THE DATA ARE ASSUMED TO HAVE
C UNPRONOUNCED SLOPES IN THE ASYMPTOTIC REGIONS AND MAY HAVE A VERY
C SHARP TRANSITION REGION OR RELATIVELY NOISY DATA.  IF ICASE=2, THE
C DATA MAY HAVE PRONOUNCED SLOPES.  THE INITIAL ESTIMATES ARE BASED
C ON IDENTIFYING THE PRE-TRANSITION, POST-TRANSITION AND TRANSITION
C REGIONS IN THE DATA EDITING ROUTINE.  ICASE=2 IS ALWAYS ASSUMED
C WHEN THE DATA EDITING SUBROUTINE RETURNS WITH TWO MARKERS ACTIVE.
C SIGN IS THE SIGN OF C(1)-C(2) AND IS USED BY LGSTC TO DISTINGUISH
C GROWTH AND DECAY PROFILES.  IF ICASE = 3, NO NEW INITIAL ESTIMATES
C ARE MADE.
C
      SUBROUTINE ESTIM8(ICASE,SIGN)
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/CONTROL/IFINIT,INSEQ,NIT,NKOUT,YOUT,ITG(4),IFC(9),IFCIN(9)
      COMMON/DATA/N,X(200),Y(200),W(200),IFIO(200),NMARK,MARK(2)
C
      GO TO (100,200,300) ICASE
C
C                       ***   CASE I   ***
C
C *** CALCULATE INITIAL VALUES OF C(1) AND C(2)
C *** CIN(1) = AVERAGE OF FIRST THREE DATA POINTS INCLUDED IN THE FIT.
C *** CIN(2) = AVERAGE OF THE LAST THREE DATA POINTS.  YAV IS THE FIRST
C *** APPROXIMATION TO THE VALUE OF Y AT THE MIDPOINT OF THE TRANSITION
C *** INTERVAL.
C
  100 IF (IFC(1) .EQ. 0) GO TO 121
      NLO=0
      I = 0
      J = 0
      CIN(1) = 0
  120 I = I+1
      IF (IFIO(I) .LT. 0) GO TO 120
      IF (NLO .EQ. 0) NLO=I
      CIN(1) = CIN(1) + Y(I)
      J = J+1
      IF (J .LT. 3) GO TO 120
      CIN(1) = CIN(1)/3.0
  121 IF (IFC(2) .EQ. 0) GO TO 123
      NHI=0
      I = N+1
      J = 0
      CIN(2) = 0
  122 I = I-1
      IF (IFIO(I) .LT. 0) GO TO 122
      IF (NHI .EQ. 0) NHI=I
      CIN(2) = CIN(2) + Y(I)
      J = J+1
      IF (J .LT. 3) GO TO 122
      CIN(2) = CIN(2)/3.0
  123 YAV = (CIN(1)+CIN(2))/2.0
      SIGN = (CIN(1)-CIN(2))/ABS(CIN(1)-CIN(2))
C
C *** CALCULATE INITIAL VALUES OF C(3) AND C(4).  IF INSEQ=1
C *** THE DATUM JUST BELOW YAV IS LOCATED BY NOTING THE VALUE OF Y AT
C *** WHICH Y-YAV CHANGES SIGN.
C *** A TANGENT LINE TO THE TRANSITION INTERVAL IS ESTIMATED BY A
C *** STRAIGHT LINE THROUGH FOUR DATA POINTS, TWO JUST BELOW AND
C *** TWO JUST ABOVE YAV.
C *** CIN(3) IS THE VALUE OF X FROM THE TANGENT LINE WHERE Y = YAV.
C *** CIN(4) IS ONE FOURTH THE DIFFERENCE IN THE VALUES OF X ON THE
C *** TANGENT LINE FOR WHICH Y = C(1) AND Y = C(2)
C
      IF (INSEQ .EQ. 2) GO TO 150
      DO 130 I = 2,N-1
      IM = I-1
      IF (IFIO(I) .LT. 0) GO TO 130
      IF (Y(I) .EQ. YAV) GO TO 131
      TEST = (Y(I)-YAV)/(Y(IM)-YAV)
      IF (TEST .LT. 0) GO TO 131
  130 CONTINUE
  131 MP = IM
      N1 = MP - 1
      N2 = MP + 2
      X0=0
      CALL XYLINE(X,Y,W,IFIO,N1,N2,X0,1,C0,C1)
      IF (IFC(3) .EQ. 1) CIN(3) = (YAV - C0)/C1
      IF (IFC(4) .EQ. 1) CIN(4) = 0.25*(CIN(2) - CIN(1))/C1
C
C *** IF CIN(4) < 0 THE ESTIMATE IS NOT VALID.  WE NEXT
C *** ATTEMPT TO ESTIMATE CIN(4) BY LOCATING THE LARGEST CHANGE
C *** IN Y CONSISTENT WITH THE SIGN OF CIN(1)-CIN(2)
C *** WE ALSO GO DIRECTLY TO 150 WHEN THE CONVERGENCE TEST HAS FAILED
C *** IN LGSTC.
C *** IF CIN(4) < 0 AND INSEQ=2, MERELY REVERSE SIGN OF CIN(4) TO
C *** KEEP ANALYSIS GOING.
C
      IF (CIN(4) .GT. 0) GO TO 170
      IF (INSEQ .EQ. 2) THEN
        CIN(4) = -CIN(4)
        GO TO 200
      END IF
  150 DMAX=0.0
      NLO1=NLO+1
      DO 151 I=NLO1,NHI
      IM1=I-1
      TEST = (Y(IM1)-Y(I))*SIGN
      IF (TEST .LT. DMAX) GO TO 151
```

```
      DMAX=TEST
      IM=IM1
  151 CONTINUE
      INSEQ = 2
      GO TO 131
C
C *** SET INITIAL VALUES OF C(5) THRU C(9) = 0
C *** SET CURRENT VALUES OF PARAMETERS C(I) EQUAL TO INITAL VALUES
C
  170 DO 171 I=5,9
  171 CIN(I) = 0.0
      GO TO 300
C
C                           *   CASE III   *
C
C *** THE MARKERS SET IN THE DATA EDITING SESSION (IF SET) ARE USED
C *** TO DISTINGUISH THE THREE REGIONS.  STRAIGHT LINES ARE ESTIMATED
C *** IN EACH OF THE REGIONS TO PROVIDE THE INITIAL ESTIMATES OF THE
C *** LOGISTIC PARAMETERS
C
C *** FIRST ESTIMATE A LINE THROUGH THE TRANSITION REGION
C
  200 NLO = MARK(1)
      NHI = MARK(2)
      X0 = 0
      CALL XYLINE(X,Y,W,IFIO,NLO,NHI,X0,1,C0,C1)
      X0=(Y(NLO)+Y(NHI)-2.0*C0)/(2.0*C1)
      IF (IFC(3) .EQ. 1) CIN(3)=X0
      IF (IFC(4) .EQ. 1) CIN(4)=(X(NHI)-X(NLO))/4.0
      NLO = 1
      NHI = MARK(1)
      CALL XYLINE(X,Y,W,IFIO,NLO,NHI,X0,1,C0,C1)
      IF (IFC(1) .EQ. 1) CIN(1)=C0
      IF (IFC(5) .EQ. 1) CIN(5)=C1
      NLO = MARK(2)
      NHI = N
      CALL XYLINE(X,Y,W,IFIO,NLO,NHI,X0,1,C0,C1)
      IF (IFC(2) .EQ. 1) CIN(2)=C0
      IF (IFC(6) .EQ. 1) CIN(6)=C1
      SIGN=(CIN(1)-CIN(2))/ABS(CIN(1)-CIN(2))
  300 DO 301 I = 1,9
  301 C(I) = CIN(I)
      RETURN
      END
```

```
C
C                     **************************
C                     *                        *
C                     *   SUBROUTINE DERIV     *
C                     *                        *
C                     **************************
C
C                         VERSION 3/19/86
C
C CALCULATES THE VALUE OF Y AND THE DERIVATIVES OF Y WITH RESPECT
C TO ALL SEVEN LOGISTIC PARAMETERS FOR A GIVEN VALUE OF X = XVAL.
C RETURNS WITH THE VALUE OF Y IN YCALC AND THE DERIVATIVES IN G(I)
C
      SUBROUTINE DERIV
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
C
      DX = XVAL-C(3)
      QDX = C(9)*DX
      EMAX = 38.
      IF (QDX .GT. EMAX) QDX = EMAX
      IF (QDX .LT. -EMAX) QDX = -EMAX
      EQDX = EXP(QDX)
      D = (2.0*C(4))/(1+EQDX)
      Z = DX/D
      IF (Z .GT. EMAX) Z = EMAX
      IF (Z .LT. -EMAX) Z = -EMAX
   20 EP = 1.+EXP(Z)
      EM = 1.+EXP(-Z)
      TRMA = (C(1)+C(5)*DX+C(7)*DX*DX)/EP
      TRMB = (C(2)+C(6)*DX+C(8)*DX*DX)/EM
      YCALC = TRMA+TRMB
      DYDZ = TRMB/EP-TRMA/EM
      DZDX0 = -(1.+ EQDX + QDX*EQDX)/(2.0*C(4))
      DZDD0 = -Z/C(4)
      DZDQ = DX*DX*EQDX/(2.0*C(4))
      G(1) = 1/EP
      G(2) = 1/EM
      G(3) = DYDZ*DZDX0 - (C(5)+2.0*C(7)*DX)/EP - (C(6)+2.0*C(8)*DX)/EM
      G(4) = DYDZ*DZDD0
      G(5) = DX/EP
      G(6) = DX/EM
      G(7) = DX*DX/EP
      G(8) = DX*DX/EM
      G(9) = DYDZ*DZDQ
  200 RETURN
      END
```

```
C
C                        ***********************
C                        *                     *
C                        *   SUBROUTINE FIX    *
C                        *                     *
C                        ***********************
C
C                           VERSION 5/20/86
C
C SUBROUTINE FIX IS CALLED BY SUBROUTINE LGSTC AND CALLS
C SUBROUTINES DERIV AND ORTHO
C
C THIS SUBROUTINE PREPARES THE FITTING VECTORS FOR ORTHO.  ONLY THOSE
C VECTORS CORRESPONDING TO PARAMETERS BEING VARIED ARE CALCULATED.
C BEFORE RETURNING TO LGSTC, THE VECTORS COR(I) AND BS(I) ARE
C ADJUSTED BY INSERTING 0'S FOR THOSE PARAMETERS NOT BEING VARIED
C SO THAT THE CORRECTIONS TO THE COEFFICIENTS RETURNED BY ORTHO
C ARE IN CORRECT CORRESPONDANCE TO THE COEFFICIENTS OF LGSTC.
C THE VARIANCE-COVARIANCE MATRIX, VCV, IS ALSO ADJUSTED ACCORDINGLY.
C
C NOTE THAT ALTHOUGH ORTHO RETURNS WITH THE DEVIATION VECTORS IN
C Z(MZ+1,I), THEY ARE NOT IN CORRESPONDANCE WITH THE ORIGINAL DATA
C BECAUSE OF DATA THAT ARE EXCLUDED FROM THE FIT
C
      SUBROUTINE FIX
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/CONTROL/IFINIT,INSEQ,NIT,NKOUT,YOUT,ITG(4),IFC(9),IFCIN(9)
      COMMON/STAT/COR(9),BS(9),VCV(9,9),NZ,MZ,STD,REL,GMDT,ESTSTD
      COMMON/DATA/N,X(200),Y(200),W(200),IFIO(200),NMARK,MARK(2)
      COMMON/SCRATCH/Z(11,209)
C
C *** I,II INDEX THE DATA
C *** J,JJ INDEX THE PARAMETERS
C *** SINGLE LETTERS ARE INDICES USED BY LGSTC
C *** DOUBLE LETTERS ARE INDICES USED BY ORTHO
C
C *** DERIV CALCULATES YCALC AND THE DERIVATIVES OF Y WITH RESPECT
C *** TO THE SEVEN PARAMETERS C(I) AND RETURNS WITH THE DERIVATIVES
C *** IN THE VECTOR G(I).  THESE DERIVATIVES ARE THE COMPONENTS
C *** OF THE INDEPENDENT VECTORS Z(J,I)
C
      II = 0
   50 DO 110 I = 1,N
      IF (IFIO(I) .LT. 0) GO TO 110
      II = II+1
      XVAL = X(I)
      CALL DERIV
      JJ = 0
      DO 100 J = 1,9
      IF (IFC(J) .EQ. 0) GO TO 100
      JJ = JJ+1
      Z(JJ,II) = G(J)
  100 CONTINUE
      K = JJ+1
      Z(K,II) = Y(I)-YCALC
      K = JJ+2
      Z(K,II) = W(I)
  110 CONTINUE
      MZ = JJ
      NZ = II
      CALL ORTHO
C
C *** ORTHO PERFORMS THE LEAST SQUARES ON THE VECTORS IN Z.
C *** ON RETURNING, INSERT 0'S INTO COR, BS, AND VCV CORRESPONDING
C *** TO THE PARAMETERS BEING HELD CONSTANT
C
C *** SINGLE LETTERS ARE INDICES USED BY LGSTC
C *** DOUBLE LETTERS ARE INDICES USED BY ORTHO
C *** NOTE THAT THE INDICES OF Z ARE NOT READJUSTED TO CORRESPOND
C *** TO THE INDICES OF THE DATA SINCE THE ORIGINAL DATA STILL
C *** RESIDES IN X AND Y.
C
      II = MZ
      DO 190 IM = 1,9
      I = 10-IM
      IF (IFC(I) .EQ. 0) GO TO 170
      JJ = II
      DO 160 JM = 1,I
      J = I+1-JM
      IF (IFC(J) .EQ. 0) GO TO 160
      VCV(I,J) = VCV(II,JJ)
      VCV(J,I) = VCV(JJ,II)
      JJ = JJ-1
  160 CONTINUE
      COR(I) = COR(II)
      BS(I)  = BS(II)
      II = II-1
      GO TO 190
  170 DO 180 J = 1,9
      VCV(I,J) = 0.0
  180 VCV(J,I) = 0.0
      COR(I) = 0.0
      BS(I) = 0.0
  190 CONTINUE
      RETURN
      END
```

```
C
C                    *************************
C                    *                       *
C                    *   SUBROUTINE ORTHO    *
C                    *                       *
C                    *************************
C
C BASED ON THE VERSION ORIGINALLY PUBLISHED BY WALSH, P. J.,
C ALGORITHM 127, ORTHO, COMMUNICATIONS OF THE ACM, 5, 1962, 511-13.
C
C SEE ALSO WAMPLER, R. H.,  J. RES. NAT. BUR. STDS., 73B, 1969, 59-90,
C FOR AN EVALUATION OF PERFORMANCE
C
C                          VERSION 3/5/86

C ORTHO PERFORMS A LINEAR LEAST SQUARES FIT OF A SINGLE DEPENDENT
C VECTOR TO A SET OF INDEPENDENT VECTORS.  THE CALCULATION PROCEEDS BY
C ORTHONORMALIZING THE DEPENDENT VECTORS.  THE LEAST SQUARES SOLUTION
C FOR A SET OF ORTHONORMAL INDEPENDENT VECTORS IS TRIVIAL.  STABILITY
C IS GAINED IN THE FIT BY REORTHOGONALIZING THE DEPENDENT VECTORS AFTER
C NORMALIZING.  THE LEAST SQUARES MATRIX IS NEVER FORMED.
C
C SUBROUTINE VARIABLES
C
C  NZ      THE NUMBER OF DATA POINTS TO BE FIT.
C
C  MZ      THE NUMBER OF INDEPENDENT VARIABLES TO BE FIT AND HENCE
C          THE NUMBER OF PARAMETERS TO BE DETERMINED.
C
C  Z(J,I)  CONTAINS THE VALUES OF THE DEPENDENT VARIABLES, X(J,I), FOR
C          J = 1 TO MZ, THE INDEPENDENT VARIABLES, Y(I), FOR J = MZ+1,
C          AND THE WEIGHTS, W(I), FOR J=MZ+2.
C          THE CALLING PROGRAM ASSIGNS THE VALUES FOR ALL THE ELEMENTS
C          OF Z(J,I).
C          THE FIRST MZ ROWS OF Z(J,I) ARE AUGMENTED BY MZ COLUMNS FROM
C          I = NZ+1 TO I = NZ+MZ TO FORM AN MZ X MZ UNIT MATRIX.  THE
C          LAST ROW, J = MZ+1, IS AUGMENTED WITH MZ 0'S FROM I = NZ+1
C          TO I = NZ+MZ. THE ORTHONORMALIZATION TRANSFORMATIONS OPERATE
C          ON THE AUGMENTED PORTIONS OF Z(J,I) THOUGH THE COEFFICIENTS
C          OF THE TRANSFORMATIONS ARE CALCULATED FROM THE UNAUGMENTED
C          PORTIONS OF THE VECTORS.  AFTER ORTHONORMALIZATION, THE
C          AUGMENTED PORTION OF Z CONTAINS THE TRANSFORMATION MATRIX IN
C          ROWS 1 TO MZ AND COLUMNS NZ+1 TO NZ+MZ AND THE NEGATIVE
C          VALUES OF THE COEFFICIENTS OF THE LEAST SQUARES FIT IN ROW
C          MZ+1, COLUMNS NZ+1 TO NZ+MZ.  THE VARIANCE-COVARIANCE MATRIX
C          IS CALCULATED FROM THIS TRANSFORMATION MATRIX.
C
C  K       INDEXES THE VECTOR CURRENTLY BEING ORTHONORMALIZED.
C  QK(K)   CONTAINS THE INNER PRODUCTS OF THE KTH VECTOR WITH ITSELF
C          AND WITH THE K-1 PREVIOUSLY ORTHONORMALIZED VECTORS.
C  DK      THE NORM OF THE MOST RECENTLY ORTHOGONALIZED VECTOR.
C  DK2     THE SQUARE OF DK
C  GMDT    THE GRAM DETERMINANT, THE DETERMINANT OF THE LEAST SQUARES
C          MATRIX.
C  VAR     THE VARIANCE
C  STD     THE STANDARD DEVIATION (SQUARE ROOT OF THE VARIANCE).
C  VCV     THE MZ X MZ VARIANCE-COVARIANCE MATRIX, THE CORRELATION
C          MATRIX MULTIPLIED BY THE VARIANCE.
C  COR     THE MZ COEFFICIENTS OF THE FIT.  FOR AN ITERATED, NON-LINEAR
C          LEAST SQUARES FIT, THESE ARE THE CORRECTIONS TO THE CURRENT
C          VALUES OF THE COEFFICIENTS
C  NPASS   CONTROL SWITCH.  NPASS = 1 DURING INITIAL ORTHOGONALIZATION
C          AND NORMALIZATION OF A VECTOR AND -1 DURING
C          REORTHOGONALIZATION.
C
C NOTE THAT THE SUBROUTINE RETURNS WITH THE ORTHONORMALIZED VECTORS IN
C Z, THE ORTHONORMALIZATION TRANSFORMATION MATRIX IN THE AUGMENTED
C PORTION OF Z, THE DEVIATION VECTOR (OBS-CALC) IN THE LAST ROW,
C Z(MZ+1,I), AND THE NEGATIVE VALUES OF THE DETERMINED COEFFICIENTS IN
C THE AUGMENTED PORTION OF Z(MZ+1,NZ+1) TO Z(MZ+1,NZ+MZ).
C
C DIMENSIONS
C
C IF MZ IS THE MAXIMUM NUMBER OF PARAMETERS TO BE DETERMINED (MAXIMUM
C NUMBER OF INDEPENDENT VECTORS) AND NZ IS THE MAXIMUM NUMBER OF DATA
C POINTS TO BE FIT, THEN THE FOLLOWING SHOULD BE THE DIMENSIONS OF
C THE DIMENSIONED VARIABLES IN ORTHO:
C
C   Z(MZ+2,NZ+MZ)
C   VCV(MZ,MZ)
C   BS(MZ)
C   COR(MZ)
C   QK(MZ+1)
C
      SUBROUTINE ORTHO
      COMMON/STAT/COR(9),BS(9),VCV(9,9),NZ,MZ,STD,REL,GMDT,ESTSTD
      COMMON/SCRATCH/Z(11,209)
      DIMENSION QK(9)
      DOUBLE PRECISION SUM,QK,DK,DK2
C
C *** BEGIN THE CALCULATION BY SETTING UP SOME LIMITS AND SOME INITIAL
C *** VALUES.
C
      NPM = NZ + MZ
      NP1 = NZ + 1
      MP1 = MZ + 1
      MP2 = MZ + 2
      GMDT = 1.0
      K = 0
      NPASS = -1
C
C *** AUGMENT THE Z MATRIX WITH AN MXM UNIT MATRIX AND A ROW OF 0'S.
C
      DO 10 J = 1,MP1
      DO 10 I = NP1,NPM
   10 Z(J,I) = 0.0
      DO 20 J = 1,MZ
      I = NZ + J
```

```
   20 Z(J,I) = 1.0
C
C *** RETURN TO STATEMENT 100 AT THE BEGINNING OF THE
C *** ORTHONORMALIZATION OF EACH VECTOR.
C
  100 KM1 = K
      K = K + 1
C
C *** RETURN TO STATEMENT 110 FOR REORTHOGONALIZATION
C
  110 NPASS = -NPASS
      DO 130 J = 1,K
      SUM = 0.0
      DO 120 I = 1,NZ
  120 SUM = SUM + Z(K,I)*Z(MP2,I)*Z(J,I)
  130 QK(J) = SUM
      IF (KM1) 170,170,140
  140 DO 150 J = 1,KM1
      DO 150 I = 1,NPM
  150 Z(K,I) = Z(K,I)-QK(J)*Z(J,I)
C
C *** IF WE HAVE JUST COMPLETED REORTHOGONALIZATION (NPASS = -1),
C *** WE STOP HERE AND GO BACK TO STATEMENT 100 TO START WORK ON THE
C *** NEXT VECTOR.
C
  170 IF (NPASS) 100,180,180
C
C *** IF WE HAVE JUST COMPLETED THE INITIAL ORTHOGONALIZATION OF THE
C *** CURRENT VECTOR (NPASS = 1) WE PROCEED WITH NORMALIZATION.
C
  180 DK2 = 0.0
      DO 190 I = 1,NZ
  190 DK2 = DK2 + Z(K,I)*Z(MP2,I)*Z(K,I)
C
C *** THE SQUARE OF THE NORM OF THE ORTHOGONALIZED VECTOR, DK2, AND
C *** THE SQUARE OF THE NORM OF THE UNORTHOGONALIZED VECTOR, QK(K),
C *** ARE USED TO CALCULATE THE GRAM DETERMINANT.
C
      IF (K-MP1) 200,300,300
C
C *** IF WE HAVE JUST COMPLETED CALCULATING THE NORM OF THE DEPENDENT
C *** VECTOR (I.E. IF K = MP1) WE JUMP AHEAD TO STATEMENT 300 TO
C *** CALCULATE THE DESIRED PARAMETERS AND STATISTICS AND RETURN.
C *** OTHERWISE, WE CONTINUE WITH THE NORMALIZATION OF THE
C *** ORTHOGONALIZED VECTOR AND GO BACK TO 110 TO REORTHOGONALIZE.
C
  200 DK = DSQRT(DK2)
      GMDT = GMDT*DK2/QK(K)
      DO 210 I = 1,NPM
  210 Z(K,I) = Z(K,I)/DK
      GOTO 110
C
C *** IF WE HAVE FINISHED WITH THE ORTHONORMALIZATION OF THE
C *** INDEPENDENT VECTORS AND THE ORTHOGONALIZATION OF THE DEPENDENT
C *** VECTOR, WE CALCULATE THE VARIANCE AND THE STANDARD DEVIATION OF
C *** THE FIT, THE VARIANCE-COVARIANCE MATRIX AND THE VALUES OF THE
C *** COEFFICIENTS AND THEIR STANDARD DEVIATIONS.  WE THEN RETURN TO
C *** THE CALLING PROGRAM.
C
  300 VAR = DK2/(NZ-MZ)
      STD = SQRT(VAR)
      DO 330 I = 1,MZ
      NI = NZ + I
      DO 320 J = I,MZ
      NJ = NZ + J
      SUM = 0.0
      DO 310 L = 1,MZ
  310 SUM = SUM + Z(L,NI)*Z(L,NJ)
      VCV(I,J) = SUM*VAR
  320 VCV(J,I) = SUM*VAR
      COR(I) = -Z(MP1,NI)
  330 BS(I) = SQRT(VCV(I,I))
      RETURN
      END
```

```
C
C                       ************************
C                       *                      *
C                       *   SUBROUTINE EPARM   *
C                       *                      *
C                       ************************
C
C                            VERSION 5/20/86
C
C *** EVALUATES AND PRINTS PARAMETERS AND ASSOCIATED STATISTICAL
C *** INFORMATION.
C *** ICON APPROXIMATES THE NUMBER OF SIGNIFICANT FIGURES IN THE
C *** PARAMETERS, BEYOND THE STANDARD DEVIATION, THAT REMAIN UNCHANGED
C *** IN THE MOST RECENT ITERATION.  IT IS USED AS A TEST FOR
C *** CONVERGENCE. NOCON IS INCREMENTED IF THE TEST FAILS.
C *** IFEND = 1 AFTER ANALYSIS IS COMPLETE AND OUTPUT GOES TO UNIT
C *** L1.  OTHERWISE, OUTPUT GOES TO UNIT L2 THE SESSION LOG.
C
      SUBROUTINE EPARM(IOU,NOCON,NIN)
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/CONTROL/IFINIT,INSEQ,NIT,NKOUT,YOUT,ITG(4),IFC(9),IFCIN(9)
      COMMON/STAT/COR(9),BS(9),VCV(9,9),NZ,MZ,STD,REL,GMDT,ESTSTD
      COMMON/MISC/W0,W1,KIT,NCYC
      LOGICAL NOTE
      CHARACTER NAME(9)*2
      CHARACTER F1(2)*16,F2(0:10)*1,F3(0:1)*12,F4*10,
     1F5(0:1)*8,FLINE*66
      DATA NAME/' A',' B','X0','D0','AS','BS','AQ','BQ',' Q'/
      DATA F1/'(1X,A2,'' ='',F8.','(1X,A2,'' <'',F8.'/
      DATA F2/'0','1','2','3','4','5','6','7','8','9','10'/
      DATA F3/',12X',','' +/-'',F8.'/
      DATA F4/','' ('',F8.'/
      DATA F5/',''  *)'')',',I3,'')'')'/
C
      WRITE (IOU,'(/'' CYCLE'',I3,'' ITERATION NO.'',I4/)') NCYC,KIT
      NOTE = .FALSE.
      NOCON = 0
      DO 200 I = 1,9
      L=1
      DBASE=BS(I)
      IF (DBASE .EQ. 0.0) DBASE=10*ABS(C(I))
      IF (DBASE .NE. 0.0) L=6-INT(ALOG10(DBASE)+5)
      IF (L .GT. 5) L=5
      IF (L .LT. 0) L=0
      LIN=L
      IF (CIN(I) .EQ. 0.0) LIN=1
      IF (IFC(I) .EQ. 0) GO TO 100
      FCON = 10.0
      IF (COR(I) .NE. 0.0) FCON = -ALOG10(ABS(COR(I)/BS(I)))
      ICON = FCON
      IF (ICON .LT. 1) NOCON = NOCON+1
      FLINE=F1(1)//F2(L)//F3(1)//F2(L)//F4//F2(LIN)//F5(1)
      WRITE (IOU,FLINE) NAME(I),C(I),BS(I),CIN(I),ICON
      GO TO 200
  100 IF (IFCIN(I) .EQ. 0) GO TO 130
      NOTE = .TRUE.
      IF (I .NE. 3) GO TO 110
      FLINE = F1(1)//F2(L)//F3(1)//F2(L)//F4//F2(LIN)//F5(0)
      WRITE (IOU,FLINE) NAME(I),C(I),BS(I),CIN(I)
      GO TO 200
  110 IF (I .NE. 4) GO TO 120
      FLINE=F1(2)//F2(L)//F3(0)//F4//F2(LIN)//F5(0)
      CVAL=2*C(I)
      WRITE (IOU,FLINE) NAME(I),CVAL,CIN(I)
      GO TO 200
  120 FLINE = F1(1)//F2(L)//F3(0)//F4//F2(LIN)//F5(0)
      WRITE (IOU,FLINE) NAME(I),C(I),CIN(I)
      GO TO 200
  130 IF (I .GT. 4) GO TO 200
      NOTE = .TRUE.
      FLINE = F1(1)//F2(L)//',''  *'')'
      WRITE (IOU,FLINE) NAME(I),C(I)
  200 CONTINUE
      IF (NOTE) WRITE (IOU,'(/'' * NOT VARIED IN THE ANALYSIS'')')
      YAV=(C(1)+C(2))/2.0
      WAV=W0+W1*YAV
      REL = 100*(STD/SQRT(WAV))/YAV
      LNZ = INT(ALOG10(REAL(NZ)))+2
      LNIN = 2
      IF (NIN .NE. 0) LNIN = INT (ALOG10(REAL(NIN)))+2
      FLINE='(/I'//F2(LNZ)//','' DATA POINTS,'',I'//
     1F2(LNIN)//','' IN THE INTERVAL'')'
      WRITE (IOU,FLINE) NZ,NIN
      K=INT(ALOG10(STD)+3)
      L=5-K
      IF (L .LT. 0) L=0
      IF (L .GT. 5) L=5
      IF (L .EQ. 1) K=5
      IF (L .GT. 1) K=L+3
      FLINE='('' STANDARD DEVIATION ='',F'//F2(K)//'.'//F2(L)//
     1','' ('',F'//F2(K)//'.'//F2(L)//','')'')'
      WRITE (IOU,FLINE) STD,ESTSTD
      WRITE (IOU,900) REL
      WRITE (IOU,903) GMDT
      RETURN
  900 FORMAT (' RELATIVE DEVIATION = ',F6.2,'%')
  903 FORMAT (' GRAM DETERMINANT   = ',E14.6/)
      END
```

```
C
C                      ***************************
C                      *                         *
C                      *   SUBROUTINE DELX(F)    *
C                      *                         *
C                      ***************************
C
C                          VERSION 5/20/86
C
C  DELX IS A SUBROUTINE TO ACCOMPANY SUBROUTINE LGSTC.  IT CALCULATES
C  THE INTERVAL, DX, OF THE TRANSITION REGION IN WHICH Y VARIES FROM
C  A FRACTION F OF COMPLETION TO A FRACTION 1-F.  THE MOST COMMON
C  INTERVAL IS FROM 10% TO 90%.  IF THE ASYMMETRY PARAMETER Q
C  IS ZERO, THE CALCULATION IS TRIVIAL.  IF NOT, THE INTERVAL MUST BE
C  SOLVED BY SUCCESSIVE APPROXIMATIONS.
C
      SUBROUTINE DELX(F)
      COMMON/IOPARM/L0,L1,L2,L3
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/CONTROL/IFINIT,INSEQ,NIT,NKOUT,YOUT,ITG(4),IFC(9),IFCIN(9)
      COMMON/STAT/COR(9),BS(9),VCV(9,9),NZ,MZ,STD,REL,GMDT,ESTSTD
      IF (IFC(4) .EQ. 0) GO TO 200
      D0 = C(4)
      Q = C(9)
      XF = ALOG((1-F)/F)
      CLIM = 1.0E-06
      B =  2.0*XF*D0
      ZM = -B/2.0
      ZP = +B/2.0
C
C *** DX IS THE INTERVAL AND UDX IS ITS UNCERTAINTY
C
      DX = B
      UDX = 2.0*XF*BS(4)
      IF (IFC(9) .EQ. 0) GO TO 100
      IF (Q) 20, 100, 10
C
C *** IF Q IS 0 BY DESIGN (IFC(9) = 0) OR BY ACCIDENT, WE ARE DONE
C *** BECAUSE THE DERIVATIVE OF DX WITH RESPECT TO Q VANISHES AT Q = 0
C *** THE INITIAL ESTIMATES OF THE VALUE AT F (ZM) AND THE VALUE AT
C *** 1-F (ZP) DEPEND ON THE SIGN OF Q.
C *** DZM AND DZP ARE THE DERIVATIVES OF ZM AND ZP WITH RESPECT TO B
C *** CT IS THE CORRECTION TERM FOR THE NEXT ESTIMATE
C
   10 ZM = -B/(1+EXP(-B*Q))
      ZP = ALOG(B*Q)/Q
      GO TO 30
   20 ZM = ALOG(-B*Q)/Q
      ZP = B/(1+EXP(B*Q))
   30 EZM = EXP(Q*ZM)
      DZM = -1.0/(1.0+EZM+Q*ZM*EZM)
      CT = (B + ZM*(1+EZM))*DZM
      ZM = ZM + CT
      IF (ABS(CT) .GT. ABS(CLIM*ZM)) GO TO 30
   40 EZP = EXP(Q*ZP)
      DZP = 1.0/(1.0+EZP+Q*ZP*EZP)
      CT = (B - ZP*(1+EZP))*DZP
      ZP = ZP + CT
      IF (ABS(CT) .GT. ABS(CLIM*ZP)) GO TO 40
      DX = ZP - ZM
C
C *** WE ARE NOW FINISHED WITH THE CALCULATION OF DX.  DX4 AND DX9 ARE
C *** THE DERIVATIVES OF DX WITH RESPECT TO C(4) AND C(9) RESPECTIVELY
C
      DX4 = (B/D0)*(DZP-DZM)
      DX9 = -ZP*ZP*EZP*DZP - ZM*ZM*EZM*DZM
      UDX = DX4*DX4*VCV(4,4) + 2.0*DX4*DX9*VCV(4,9) + DX9*DX9*VCV(9,9)
      UDX = SQRT(UDX)
      ETA = -(ZP+ZM)/DX
      QD0 = D0*Q
      UQD0 = Q*Q*VCV(4,4) + 2.0*D0*Q*VCV(4,9) + D0*D0*VCV(9,9)
      UQD0 = SQRT(UQD0)
      DE4 = (B/(D0*DX*DX))*(ZM*DZP-ZP*DZM)
      DE9 = -(2.*ZM*ZP/(DX*DX))*(ZP*EZP*DZP+ZM*EZM*DZM)
      UETA = DE4*DE4*VCV(4,4)+2.0*DE4*DE9*VCV(4,9)+DE9*DE9*VCV(9,9)
      UETA = SQRT(UETA)
C
C *** AS A CHECK ON THE CALCULATION, WE CALCULATE THE % COMPLETION
C *** FOR THE VALUES WE HAVE DETERMINED FOR ZM AND ZP
C
  100 XVAL = C(3)+ZM
      CALL DERIV
      FRAC = ABS((YCALC-C(1)-C(5)*ZM)/(C(2)+C(6)*ZM-C(1)-C(5)*ZM))
      FRAC = 100.0*FRAC
      WRITE (L1,900)
      WRITE (L1,901) XVAL,FRAC
      XVAL = C(3)+ZP
      CALL DERIV
      FRAC = ABS((YCALC-C(1)-C(5)*ZP)/(C(2)+C(6)*ZP-C(1)-C(5)*ZP))
      FRAC = 100.0*FRAC
      WRITE (L1,901) XVAL,FRAC
      WRITE (L1,902) DX,UDX
      IF (Q .EQ. 0) RETURN
      WRITE (L1,903) ETA,UETA
      WRITE (L1,904) QD0,UQD0
      RETURN
  200 R10MAX = 4.0*ALOG((1-F)/F)*C(4)
      FL=100.0*F
      FU=100.0-FL
      WRITE (L1,905) FL,FU,R10MAX
      RETURN
  900 FORMAT (1H )
  901 FORMAT (' AT X  =',F6.2,' TRANSITION IS ',F4.1,'% COMPLETE')
  902 FORMAT (' RANGE =',F6.2,'   +/- ',F5.2)
  903 FORMAT (' ETA   =',F8.4,' +/- ',F7.4)
  904 FORMAT (' Q*D0  =',F8.4,' +/- ',F7.4/)
  905 FORMAT (/F4.0,'% ->',F4.0,'% RANGE < ',F6.2)
      END
```

```
C                          ********************
C                          *                  *
C                          *       PLOT       *
C                          *                  *
C                          ********************

C                            VERSION 6/11/86


C THIS PROGRAM OPERATES ON DATA STORED IN THE FILE PLOT.DAT BY THE
C PROGRAM LOGIT, AND PLOTS THE RESULTS OF THE LOGIT ANALYSIS.  FOR
C EACH DATA FILE ANALYZED BY LOGIT AND WHOSE RESULTS ARE STORED IN
C PLOT.DAT, PLOT PRESENTS FOUR GRAPHS.  THE FIRST CONSISTS OF:
C    THE OBSERVED VALUES OF X AND Y WITH POINTS EXCLUDED FROM THE FIT
C        APPEARING AS + AND POINTS IN THE FIT AS o.
C    THE CALCULATED CURVE, APPEARING AS A SOLID LINE
C    THE CALCULATED CURVE PLUS AND MINUS TWO STANDARD DEVIATIONS,
C        APPEARING AS DASHED LINES
C    THE ESTIMATED UNCERTAINTY IN EACH MEASURED VALUE OF Y
C        APPEARING AS A VERTICAL LINE AT EACH VALUE OF Y AND
C        EQUAL TO TWO STANDARD DEVIATIONS DIVIDED BY THE SQUARE
C        ROOT OF THE WEIGHT OF THE DATUM USED IN THE LEAST SQUARES
C        FIT.
C    THE VALUES OF THE PARAMETERS PRINTED IN THE MOST EMPTY QUADRANT
C        OF THE GRAPH.
C THE SECOND CONSISTS OF:
C    THE FIRST DERIVATIVE OF Y WITH RESPECT TO X
C THE THIRD CONSISTS OF:
C    THE CALCULATED VALUES AND THE OBSERVED VALUES, BUT THE CURSOR
C    FOLLOWS THE CALCULATED VALUES
C THE FOURTH CONSISTS OF:
C    A NORMALIZED PLOT OF THE DATA.  ONCE SELECTED, ALL REMAINING DATA
C    FILES, WHEN PLOTTED AS NORMALIZED DATA, WILL BE PLOTTED WITH THE
C    SAME NORMALIZATION FACTOR AND THE SAME X AND Y AXES AS THE FIRST
C    FILE.
C THE FIFTH CONSISTS OF:
C    THE VALUES OF THE STANDARDIZED RESIDUALS, AGAIN DISTINGUISHED BY
C        o AND + DEPENDING ON WHETHER THE DATA WERE INCLUDED IN THE
C        LEAST SQUARES ANALYSIS OR NOT.
C        IF A VALUE OF A STANDARD RESIDUAL IS GREATER THAN 5 OR LESS
C        THAN -5, IT IS PLOTTED AS IF IT WERE ± 5.
C    THE STANDARD DEVIATION OF THE FIT IS PRINTED AT THE TOP OF THE
C        GRAPH.

C THE FOLLOWING CALLS TO GRAPH LIBRARY ROUTINES ARE MADE:

C   0) SET THE SCREEN TO GRAPHICS MODE AND CLEAR THE SCREEN.
C      CALL AGRAF0(IRETCD)
C   1) DRAW THE AXES OF A GRAPH
C      CALL AGRAF1(IRETCD,IWIDTH,IAXES,XMIN,XMAX,YMIN,YMAX)
C   2) DISPLAY A TEXT STRING.
C      CALL AGRAF2(IRETCD,ISROW,ISCOL,ISTYPE,STRING)
C   3) PLOT A CURVE SEGMENT
C      CALL AGRAF3(IRETCD,IPTYPE,N,X,Y)
C   4) PRINT THE GRAPHICS SCREEN.
C      CALL AGRAF4(IRETCD)
C   5) SET THE SCREEN TO TEXT MODE AND CLEAR THE SCREEN.
C      CALL AGRAF5(IRETCD)
C   8) RETURN A LOGICAL VALUE DEPENDING ON KEYBOARD STATUS.
C      LVALUE = AGRAF8(IKTYPE,KEEP)

C  THE ARRAYS XP AND YP ARE FILLED BY DPLOT WITH THE VECTORS GIVEN
C  TO DPLOT.  THUS XP AND YP ALWAYS CONTAIN THE VECTORS CURRENTLY
C  DISPLAYED ON THE SCREEN.  XP AND YP ARE USED PRIMARILY FOR CURSOR
C  DISPLAY.

      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/LIMITS/NLO,NHI,NLOFIT,NHIFIT,IQMIN,ZOOM,VCV(9,9),CL975
      COMMON/DATA/NP,IFIO(300),XP(300),YP(300)
      COMMON/CDATA/NCUR,ICUR,YCUR,X(300),XC(300),N,NC,XCR(2),YCR(2)

C  YMININ AND YMAXIN ARE THE MINIMUM AND MAXIMUM VALUES OF Y FOR EACH
C  OF THE FIVE PLOTS FOR THE FULL RANGE OF X.  IN "ZOOMED" PLOTS, THE
C  EXTREMA YMIN AND YMAX ARE IN EFFECT.  XMININ, XMAXIN, XMIN, AND XMAX
C  HAVE THE CORRESPONDING SIGNIFICANCE FOR THE EXTREMA OF X.  DPLOT
C  SETS THE VALUES OF YMIN AND YMAX (EXCEPT FOR THE STANDARDIZED RESID-
C  UALS AND THE NORMALIZED PLOTS.)

      COMMON/EXTREMA/XMIN,XMAX,YMIN,YMAX,YMAXIN(5),YMININ(5)
      CHARACTER PARLIN(10)*30,STDLIN*30,FLINE*80
      CHARACTER TITLEF*30,DATLIN*16
      CHARACTER CH*80,BLANK*80,BLINE*50,BLIN1*50,STATUS(-1:1)*5
      CHARACTER FL(0:9)*1,FXX*5,FXM*30,XMARK*18
      DIMENSION P10(-4:4),TVAL(31)
      DIMENSION Y(300),W(300),EB(300)
      DIMENSION YN(300),YNC(300),EBN(300)
      DIMENSION XD(300),YD(300),UN(300),R(300)
      DIMENSION YC(300),YDC(300),YDB(300),UC(300)
      DIMENSION X2(2),Y2(2),X0(2),Y0(2),XM(2),YM(2)
      DIMENSION KEYSET1(14)
      DATA P10/.0001,.001,.01,.1,1.,10.,100.,1000.,10000./
      DATA TVAL/12.706,4.303,3.182,2.776,2.571,2.447,2.365,2.306,2.262
     1,2.228,2.201,2.179,2.160,2.145,2.131,2.120,2.110,2.101,2.093,2.086
     2,2.080,2.074,2.069,2.064,2.060,2.056,2.052,2.048,2.045,2.042,2.0/
      DATA FL/'0','1','2','3','4','5','6','7','8','9'/
      DATA NKEY1,KEYSET1/14,19200,19712,18176,15104,15360,15616,15872
     1,17152,17408,16128,22528,20224,13,26368/
      DATA STATUS/'  OUT',' ','  IN'/
      DATA BLANK/' '/
      DOUBLE PRECISION U
      LOGICAL FIRST,TTLON,LVALUE,AGRAF8,ZOOM
      ICUR=256*186+10
      IMARK=256*219+10

C FIRST IS USED TO CONTROL THE NORMALIZATION OF NORMALIZED PLOTS.  AFTER
C THE FIRST PLOT HAS BEEN NORMALIZED, FIRST IS RESET TO "FALSE".
```

```
      FIRST = .TRUE.
      L3 = 1
      OPEN (L3,FILE='PLOT.DAT',FORM='UNFORMATTED',STATUS='OLD',
     1ERR=900)
    5 READ (L3,END=800) TITLEF
      READ (L3,END=800) DATLIN
      READ (L3,END=800) (C(I), I=1,9)
      READ (L3,END=800) ((VCV(I,J), J=1,9), I=1,9)
      READ (L3,END=800) STD,N
      READ (L3,END=800) (X(I),Y(I),W(I),IFIO(I), I=1,N)

C  ADJUST THE MINIMUM AND MAXIMUM VALUES OF X AND Y TO INCLUDE THE
C  ± 99% VALUES OF THEY ARE NOT INCLUDED IN THE SPAN OF THE DATA
C  AND EXPAND THESE VALUES BY 5% IN ORDER TO ENSURE THAT ALL POINTS
C  FALL WITHIN THE LIMITS OF THE AXES

C  KVAL IS AN INDEX, USUALLY SET BY SUBROUTINE KEY, THAT DETERMINES
C  WHICH GRAPH IS TO BE DRAWN AND/OR WHAT ACTION IS TO BE TAKEN
C  FOLLOWING A KEY PRESS.  NCUR IS AN INDEX MARKING THE POSITION OF
C  THE CURSOR AND MARK IS AN INDEX MARKING THE POSITION OF THE ZOOM
C  MARKER.  KTYPE IS AN INDEX DIRECTING THE TITLING INFORMATION AT
C  TOP OF THE GRAPH.  TTLON IS A LOGICAL TOGGLE DIRECTING SUB-
C  ROUTINE TITLE TO PRINT A TITLE (TTLON = .TRUE.) OR A MINI HELP
C  LINE (TTLON = .FALSE.)  TTLON IS "FLIPPED" BY SUBROUTINE TITLE.
C  THEREFORE TTLON INDICATES THE TITLE THE NEXT TIME TITLE IS CALLED.

      NM1 = N-1
      NCUR = 0
      MARK = 0
      KTYPE = 1
      KVAL = 4
      IF (FIRST) THEN
        XMAXIN=X(N)
        XMININ=X(1)
        X99 = C(4)*ALOG(99.)
        XMAXIN = MAX(XMAXIN,C(3)+X99)
        XMININ = MIN(XMININ,C(3)-X99)
        EXT = (XMAXIN-XMININ)/20.
        XMININ = XMININ-EXT
        XMAXIN = XMAXIN+EXT

C  MAKE LIMITS EVEN MULTIPLES CORRESPONDING TO THE RANGE IN VALUES
C  DETERMINE FORMAT FOR PRINTING OF THE MINIMUM AND MAXIMUM VALUES
C  OF X INCLUDED IN THE FIT.

        RANGE = XMAXIN-XMININ
        L=INT(6-ALOG10(RANGE))-4
        XMININ = INT(XMININ*P10(L))/P10(L)
        XMAXIN = INT(XMAXIN*P10(L))/P10(L)
        XFORM = MAX(ABS(XMAXIN),ABS(XMININ))
        L=INT(ALOG10(XFORM)+1.)
        IF (L .LT. 0) L=0
        DX=(XMAXIN-XMININ)/N
        LD=INT(2.0-ALOG10(DX))
        IF (LD .GT. 4) LD=4
        IF (LD .LT. 0) LD=0
        L=L+LD+2
        IF (L .GT. 9) L=9
        FXX=',F'//FL(L)//'.'//FL(LD)
        FXM = '(''XMARK = '''//FXX//')'
        FLINE = '('' ENTER LOWER LIMIT FOR X:  ['''//FXX//','''] '')'
        WRITE (*,901)
        WRITE (*,FLINE) XMININ
        READ (*,'(A)') CH
        IF (CH .NE. ' ') READ (CH,'(G4.0)',ERR=10) XMININ
        FLINE = '(''+ENTER UPPER LIMIT FOR X:  ['''//FXX//','''] '')'
        WRITE (*,FLINE) XMAXIN
        READ (*,'(A)') CH
        IF (CH .NE. ' ') READ (CH,'(G4.0)',ERR=10) XMAXIN
        END IF
   10 XMIN = XMININ
      XMAX = XMAXIN

C  PRINT A WAIT MESSAGE

      WRITE (*,914)

C  DETERMINE THE "HIGHEST" PARAMETER VARIED, THE NUMBER OF DEGREES
C  OF FREEDOM, AND THE HIGHEST AND LOWEST POINTS INCLUDED IN THE FIT.

      NPV=0
      NCMAX=0
      DO 11 J=9,1,-1
      IF (VCV(J,J) .NE. 0.0) THEN
        IF (NCMAX .EQ. 0) NCMAX = J
        NPV=NPV+1
      END IF
   11 CONTINUE
      NDF=-NPV
      NLOFIT = 0
      DO 15 I=1,N
      IF (IFIO(I) .EQ. 1 .AND. NLOFIT .EQ. 0) NLOFIT=I
   15 IF (IFIO(I) .EQ. 1) NDF=NDF+1
      CL975=2.00
      IF (NDF .LE. 30) CL975=TVAL(NDF)
      DO 17 I=1,N
      NHIFIT=N-I+1
   17 IF (IFIO(NHIFIT) .EQ. 1) GO TO 18

C  LOGPAR FORMS STRINGS (PARLIN(I)) CONTAINING THE NAMES, VALUES, AND
C  STANDARD DEVIATIONS OF THE PARAMETERS; A STRING CONTAINING THE
C  MINIMUM AND MAXIMUM VALUES OF X INCLUDED IN THE FIT (BLIN1) TO BE
C  PRINTED AS BOTTOM LINE DATA; AND A STRING CONTAINING THE STANDARD
C  DEVIATION OF THE FIT (STDLIN).  NPL IS THE NUMBER OF PARAMETER
C  LINES TO BE PRINTED TO THE SCREEN AND IS CALCULATED BY LOGPAR.
C  FXX IS THE FORMAT FOR PRINTING X(NLOFIT) AND X(NHIFIT)

   18 XLO=X(NLOFIT)
```

```
      XHI=X(NHIFIT)
      CALL LOGPAR(XLO,XHI,VCV,STD,NPL,PARLIN,STDLIN,FXX,BLIN1)
      XVAL = XMAXIN
      CALL DERIV
      YXMAX = YCALC
      XVAL = XMININ
      CALL DERIV
      YXMIN = YCALC
      YMININ(1) = MIN(YXMIN,YXMAX)
      YMAXIN(1) = MAX(YXMIN,YXMAX)
      YDMID = (C(5)+C(6))/2.0 + (C(2)-C(1))/(4.0*C(4))
      YMININ(2) = MIN(C(5),C(6),YDMID)
      YMAXIN(2) = MAX(C(5),C(6),YDMID)
      YMININ(3) = YMININ(1)
      YMAXIN(3) = YMAXIN(1)
      YMININ(5) = -5.0
      YMAXIN(5) = +5.0
      ZOOM = .FALSE.

C  CALCULATE THE "OBSERVED" VALUES FOR THE FIRST DERIVATIVE OF
C  Y WITH RESPECT TO X AND THE "OBSERVED" VALUES OF Y CORRECTED
C  FOR THE SLOPING BASELINE.
C  CALCULATE THE VALUES OF THE STANDARDIZED RESIDUALS, R(I)

      DO 30 I=1,N
      XVAL = X(I)
      CALL DERIV
      DIFF = Y(I)-YCALC

C  EB(I) ARE THE ESTIMATED ERROR BARS FOR THE OBSERVED VALUES OF Y

      EB(I) = CL975*STD/SQRT(W(I))

C  NEXT DO THE STANDARDIZED RESIDUALS.  U IS THE UNCERTAINTY IN A
C  CALCULATED VALUE OF Y.  IT IS MADE DOUBLE PRECISION TO AVOID
C  ARITHMETIC UNDERFLOW.  IF A STANDARDIZED RESIDUAL IS GREATER IN
C  MAGNITUDE THAN 4.9, IT IS SET EQUAL TO ±4.9 FOR DISPLAY PURPOSES.

      U = 0.0
      DO 25 K=1,NCMAX
      DO 25 L=1,NCMAX
   25 U = U+G(K)*G(L)*VCV(K,L)
      SR = STD*STD/W(I)-IFIO(I)*U
      R(I) = 0.0
      IF (DIFF .NE. 0) R(I) = 4.9*DIFF/ABS(DIFF)
      IF (SR .GT. 0.0) R(I) = DIFF/SQRT(SR)
      R(I) = MIN(R(I),4.9)
      R(I) = MAX(R(I),-4.9)

C  FINALLY, DO THE FIRST DERIVATIVE VALUES

      IF (I .EQ. N) GO TO 30
      IP1=I+1
      XD(I) = (X(I)+X(IP1))/2
      YD(I) = YD(I-1)
      DIFF  = X(IP1)-X(I)
      IF (DIFF .NE. 0.0) YD(I) = (Y(IP1)-Y(I))/DIFF
   30 CONTINUE

C  CALCULATE THE COORDINATES FOR A VERTICAL LINE AT X0

      XVAL = C(3)
      X0(1) = XVAL
      X0(2) = XVAL
      CALL DERIV
      YX0 = YCALC
      YDX0 = -G(3)

C  NOW CALCULATE THE CALCULATED CURVES AND THE STANDARD
C  DEVIATION OF THE CALCULATED CURVES
C  YDB(I) IS A BASELINE FOR THE DERIVATIVE PLOT

      NC = 300
   40 STEP = (XMAX-XMIN)/(NC-1)
      XVAL=XMIN
      DO 42 I=1,NC
      CALL DERIV
      XC(I) = XVAL
      YC(I) = YCALC
      YDC(I) = -G(3)
      DX = XVAL-C(3)
      YDB(I) = (C(5)+2.0*C(7)*DX)*G(1)+(C(6)+2.0*C(8)*DX)*G(2)
      U = 0.0
      DO 41 J=1,NCMAX
      DO 41 K=1,NCMAX
   41 U = U+G(J)*VCV(J,K)*G(K)
      UC(I) = CL975*DSQRT(U)
   42 XVAL = XVAL+STEP
      GO TO 99

C  KEY READS THE KEYBOARD WITH THE LOGICAL FUNCTION AGRAF8
C  AND RETURNS A VALUE FOR THE INDEX KVAL CORRESPONDING TO
C  THE KEY PRESSED:

C  KVAL  KEY       SCAN CODE   ACTION                        STATEMENT NO.
C    1.  [<--]       19200  MOVE CURSOR LEFT                  100
C    2.  [-->]       19712  MOVE CURSOR RIGHT                 110
C    3.  HOME:       18176  TOGGLE CURSOR                     130
C    4.  F1:         15104  STANDARD PLOT                     150
C    5.  F2:         15360  DERIVATIVE PLOT                   200
C    6.  F3:         15616  CALCULATED PLOT                   300
C    7.  F4:         15872  NORMALIZED PLOT                   400
C    8.  F9:         17152  STANDARDIZED RESIDUAL PLOT        500
C    9.  F10:        17408  PRINT SCREEN                      600
C   10.  F5:         16128  SET MARKER OR ZOOM                650
C   11.  SHIFT-F5:   22528  CLEAR MARKER OR UNZOOM            700
C   12.  END:        20224  END PLOT OR PROGRAM               750
C   13.  ENTER:         13  TOGGLE TITLE/HELP LINES            90
```

```
C  14.  CTRL-F10:   26368  PRINT SCREEN AS IS            610
C
   90 CALL TITLE(TTLON,KTYPE,TITLEF,DATLIN,STDLIN,XMARK)
   95 IF (NCUR .EQ. 0) THEN
        BLINE=BLIN1
      ELSE
        IF (IDEN .EQ. 3) THEN
          WRITE (BLINE,FLINE) XP(NCUR),YP(NCUR),UC(NCUR)
        ELSE
          IO=IFIO(NCUR)
          WRITE (BLINE,FLINE) XP(NCUR),YP(NCUR),STATUS(IO)
        END IF
        L=LNBX(BLINE)
        LB=50-L
        BLINE=BLANK(1:LB)//BLINE(1:L)
      END IF
      CALL AGRAF2(IRETCD,24,15,0,BLINE)
   98 CALL KEY(KVAL,KEYSET1,NKEY1)
      IF (KVAL .EQ. IDEN+3) GO TO 90
   99 GO TO (100,110,130,150,200,300,400,500,600,650,700,750,90,610)
     1 KVAL

C  MOVE CURSOR LEFT

  100 IF (NCUR .EQ. 0) GO TO 98
  101 CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
      NCUR=NCUR-1
      IF (NCUR .LT. NLO) NCUR=NHI
      XCR(1) = XP(NCUR)
      XCR(2) = XP(NCUR)
      YCR(1) = YP(NCUR)+YCUR
      YCR(2) = YP(NCUR)-YCUR
      CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
      LVALUE=AGRAF8(19200,0)
      IF (.NOT.LVALUE) GO TO 95
  102 LVALUE=AGRAF8(0,1)
      IF (LVALUE) GO TO 102
      GO TO 101

C  MOVE CURSOR RIGHT

  110 IF (NCUR .EQ. 0) GO TO 98
  111 CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
      NCUR=NCUR+1
      IF (NCUR .GT. NHI) NCUR=NLO
      XCR(1) = XP(NCUR)
      XCR(2) = XP(NCUR)
      YCR(1) = YP(NCUR)+YCUR
      YCR(2) = YP(NCUR)-YCUR
      CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
      LVALUE=AGRAF8(19712,0)
      IF (.NOT.LVALUE) GO TO 95
  112 LVALUE=AGRAF8(0,1)
      IF (LVALUE) GO TO 112
      GO TO 111

C  TOGGLE CURSOR ON AND OFF
C  IF NCUR=0, THE CURSOR IS PRESUMED OFF
C  IF NCUR NOT = 0, THE CURSOR IS PRESUMED ON
C  KTYPE SPECIFIES THE HELP LINE AT THE TOP OF THE SCREEN

  130 TTLON = .FALSE.
      IF (NCUR .NE. 0) THEN
        CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
        NCUR=0
        MARK=0
        KTYPE=1
        IF (XMIN .NE. XMININ .OR. XMAX .NE. XMAXIN) KTYPE=4
        GO TO 90
      END IF
      NCUR=NLO
  131 XCR(1) = XP(NCUR)
      XCR(2) = XP(NCUR)
      YCR(1) = YP(NCUR)+YCUR
      YCR(2) = YP(NCUR)-YCUR
      CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
      KTYPE = 2
      GO TO 90

C  DATA PLOT

  150 CALL DPLOT(1,IW,N,X,Y,EB,FLINE,YCUR)
      CALL AGRAF3(IRETCD,1,NC,XC,YC)
      DO 151 I=1,NC
  151 UN(I) = YC(I)+UC(I)
      CALL AGRAF3(IRETCD,0,NC,XC,UN)
      DO 152 I=1,NC
  152 UN(I) = YC(I)-UC(I)
      CALL AGRAF3(IRETCD,0,NC,XC,UN)
      Y0(1) = YMIN
      Y0(2) = YX0
      CALL AGRAF3(IRETCD,1,2,X0,Y0)
      CALL PPARM(IQMIN,IW,NPL,PARLIN)
      IF (NCUR .NE. 0)
     1CALL CURSOR(1,IDEN)
      IDEN = 1
      GO TO 90

C  DERIVATIVE PLOT

  200 CALL DPLOT(2,IW,NM1,XD,YD,EB,FLINE,YCUR)
      CALL AGRAF3(IRETCD,1,NC,XC,YDC)
      CALL AGRAF3(IRETCD,0,NC,XC,YDB)
      Y0(1) = YMIN
      Y0(2) = YDX0
      CALL AGRAF3(IRETCD,1,2,X0,Y0)
      CALL PPARM(IQMIN,IW,NPL,PARLIN)
      IF (NCUR .NE. 0) CALL CURSOR(2,IDEN)
```

```
      IDEN = 2
      GO TO 90

C  FUNCTION PLOT

  300 CALL DPLOT(3,IW,NC,XC,YC,UC,FLINE,YCUR)
      CALL AGRAF3(IRETCD,2304,N,X,Y)
      YO(1) = YMIN
      YO(2) = YXO
      CALL AGRAF3(IRETCD,1,2,XO,YO)
      CALL PPARM(IQMIN,IW,NPL,PARLIN)
      IF (NCUR .NE. 0) CALL CURSOR(3,IDEN)
      IDEN = 3
      GO TO 90

C  PLOT NORMALIZED DATA

  400 IF (FIRST) THEN
        CALL AGRAF0(IRETCD)
        CALL AGRAF5(IRETCD)
        NORM = 1
        WRITE (*,940)
        READ (*,'(A)') CH
        IF (CH .NE. ' ') READ (CH,'(I2)',ERR=401) NORM
  401   IF (NORM .LT. 1 .OR. NORM .GT. 4) NORM=1
        IF (NORM .EQ. 1) B=MAX(C(1),C(2))
        IF (NORM .EQ. 2) B=MIN(C(1),C(2))
        IF (NORM. EQ. 3) THEN
          B=Y(1)
          DO 403 I=2,N
  403     B=MAX(B,Y(I))
        ELSE IF (NORM .EQ. 4) THEN
          B=Y(1)
          DO 404 I=2,N
  404     B=MIN(B,Y(I))
        END IF
        YNMIN=YMININ(1)/B
        YNMAX=YMAXIN(1)/B
        WRITE (*,941) YNMIN,YNMAX,YNMIN
        READ (*,'(A)') CH
        IF (CH .NE. ' ') READ (CH,'(G6.0)',ERR=405) YNMIN
  405   WRITE (*,942) YNMAX
        READ (*,'(A)') CH
        IF (CH .NE. ' ') READ (CH,'(G6.0)',ERR=406) YNMAX
  406   YMININ(4)=YNMIN
        YMAXIN(4)=YNMAX
        FIRST = .FALSE.
      END IF
      DO 407 I=1,N
      YN(I)=Y(I)/B
  407 EBN(I)=EB(I)/B
      CALL DPLOT(4,IW,N,X,YN,EBN,FLINE,YCUR)
      DO 408 I=1,NC
  408 YNC(I)=YC(I)/B
      CALL AGRAF3(IRETCD,1,NC,XC,YNC)
      DO 409 I=1,NC
  409 UN(I) = YNC(I)+UC(I)/B
      CALL AGRAF3(IRETCD,0,NC,XC,UN)
      DO 410 I=1,NC
  410 UN(I) = YNC(I)-UC(I)/B
      CALL AGRAF3(IRETCD,0,NC,XC,UN)
      YO(1) = YMIN
      YO(2) = YXO/B
      CALL AGRAF3(IRETCD,1,2,XO,YO)
      CALL PPARM(IQMIN,IW,NPL,PARLIN)
      IF (NCUR .NE. 0) CALL CURSOR(4,IDEN)
      IDEN = 4
      GO TO 90

C  STANDARDIZED RESIDUALS PLOT

  500 CALL DPLOT(5,IW,N,X,R,EB,FLINE,YCUR)
      X2(1) = XMIN
      X2(2) = XMAX
      Y2(1) = 0.0
      Y2(2) = 0.0
      CALL AGRAF3(IRETCD,1,2,X2,Y2)
      Y2(1) = CL975
      Y2(2) = CL975
      CALL AGRAF3(IRETCD,2,2,X2,Y2)
      Y2(1) = -CL975
      Y2(2) = -CL975
      CALL AGRAF3(IRETCD,2,2,X2,Y2)
      YO(1) = -5.0
      YO(2) = 5.0
      CALL AGRAF3(IRETCD,1,2,XO,YO)
      IF (NCUR .NE. 0) CALL CURSOR(5,IDEN)
      IDEN = 5
      GO TO 90

C  PRINT THE SCREEN AFTER ERASING HELP LINE AND PRINT THE TITLE,
C  DATE, AND STANDARD DEVIATION ON THE TOP LINE OF THE GRAPH
C  ERASE CURSOR AND MARKER (IF ON) BEFORE PRINTING
C  AFTER PRINTING, MAKE CERTAIN THAT ALL PENDING KEY PRESSES HAVE
C  BEEN CLEARED IN ORDER TO PREVENT REPEATED PRINTING.

  600 IF (TTLON) CALL TITLE(TTLON,KTYPE,TITLEF,DATLIN,STDLIN,XMARK)
      IF (NCUR .NE. 0) CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
      IF (MARK .NE. 0) CALL AGRAF3(IRETCD,IMARK,2,XM,YM)
      CALL AGRAF2(IRETCD,24,20,0,BLIN1)
      CALL AGRAF4(IRETCD)
      IF (IRETCD. NE. 0) THEN
  601   LVALUE=AGRAF8(0,1)
        IF (LVALUE) GO TO 601
        TTLON = .FALSE.
        CALL TITLE(TTLON,5,TITLEF,DATLN,STDLIN,XMARK)
  602   LVALUE=AGRAF8(0,1)
        IF (.NOT.LVALUE) GO TO 602
```

```
      END IF
      IF (NCUR .NE. 0) CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
      IF (MARK .NE. 0) CALL AGRAF3(IRETCD,IMARK,2,XM,YM)
      GO TO 90
  610 CALL AGRAF4(IRETCD)
      GO TO 90

C  PUT A MARKER ON THE SCREEN AT THE POSITION OF THE CURSOR.  IF A
C  MARKER IS ALREADY IN PLACE, ZOOM.  IF THE CURSOR IS THE SAME
C  POSITION AS THE MARKER, ZOOM TO HALF OF THE CURRENT DISPLAY.
C  IF THE CURSOR IS IN A DIFFERENT POSITION FROM THE MARKER, ZOOM
C  TO THE RANGE BETWEEN THE MARKER AND THE CURSOR.
C  IF THE CURSOR IS NOT ON, WE CANNOT MARK, SO RETURN AND CHANGE
C  TITLE LINE AS A REMINDER.  SIMILARLY IF WE ARE EXAMINING THE
C  CALCULATED PLOT ITSELF (IDEN=3).

  650 IF (NCUR .EQ. 0) GO TO 90
      IF (IDEN .EQ. 3) GO TO 90
      TTLON = .FALSE.
      IF (MARK .EQ. 0) THEN
        CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
        XM(1) = XCR(1)
        XM(2) = XCR(2)
        YM(1) = YCR(1)
        YM(2) = YCR(2)
        CALL AGRAF3(IRETCD,IMARK,2,XM,YM)
        CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
        MARK = NCUR
        WRITE (XMARK,FXM) X(MARK)
        KTYPE = 3
        GO TO 90
      END IF

C  IF MARK IS NOT ZERO, WE ZOOM

      IF (MARK .NE. NCUR) THEN
        XMIN=MIN(X(MARK),X(NCUR))
        XMAX=MAX(X(MARK),X(NCUR))
        XRANGE=XMAX-XMIN
      ELSE
        XRANGE=(XMAX-XMIN)/2.0
        XLO=X(MARK)-XRANGE/2.0
        XHI=X(MARK)+XRANGE/2.0
        IF (XLO .LT. XMIN) THEN
          XMAX=XMIN+XRANGE
        ELSE IF (XHI .GT. XMAX) THEN
          XMIN=XMAX-XRANGE
        ELSE
          XMIN=XLO
          XMAX=XHI
        END IF
      END IF
      L=INT(6-ALOG10(XRANGE))-4
      XMIN=INT(XMIN*P10(L))/P10(L)
      XMAX=INT(XMAX*P10(L)+1)/P10(L)
      KVAL=IDEN+3
      KTYPE=4
      MARK=0
      NCUR=0
      ZOOM=.TRUE.
      CALL AGRAF0(IRETCD)
      CALL AGRAF2(IRETCD,10,37,0,'ZOOMING')
      GO TO 40

C  IF MARKER IS ON, TURN IT OFF
C  IF MARKER IS OFF, RESTORE THE ORIGINAL X RANGE
C  IF WE ARE IN THE ORIGINAL X RANGE, CHANGE THE TITLE LINE

  700 KVAL = IDEN+3
      IF (MARK .EQ. 0) THEN
        IF (XMIN .EQ. XMININ .AND. XMAX .EQ. XMAXIN) GO TO 90
        XMIN = XMININ
        XMAX = XMAXIN
        KTYPE = 1
        ZOOM = .FALSE.
        CALL AGRAF0(IRETCD)
        CALL AGRAF2(IRETCD,10,36,0,'UNZOOMING')
        TTLON = .FALSE.
        GO TO 40
      ELSE
        CALL AGRAF3(IRETCD,IMARK,2,XM,YM)
        MARK = 0
        KTYPE = 2
        TTLON = .FALSE.
        GO TO 90
      END IF

C  END PLOT FOR THIS FILE.  SEE IF ANY DATA REMAIN TO BE PLOTTED
C  IF NOT, END PROGRAM

  750 CALL AGRAF0(IRETCD)
      GO TO 5
  800 CLOSE (L3)
      CALL AGRAF0(IRETCD)
      CALL AGRAF5(IRETCD)
      STOP 1986
  900 WRITE (*,999)
      CLOSE (L3)
      STOP 900
  901 FORMAT (' ENTER VALUES FOR LOWER AND UPPER LIMITS OF X'/
     1' TO BE PLOTTED.  PRESSING [ENTER] ALONE ACCEPTS DEFAULT VALUE'/)
  904 FORMAT (' FOR A NORMALIZED PLOT, THE MAXIMUM VALUE OF Y IS 1.0'/
     1' ENTER THE MINIMUM VALUE OF Y: [',F5.2,'] ')
  914 FORMAT ('+CALCULATING . . .PLEASE WAIT')
  920 FORMAT ('X(min) = ',F6.2,',  X(max) = ',F6.2)
  940 FORMAT (' NORMALIZE PLOT ON:'/'      [1] UPPER ASYMPTOTE'/
     1'      [2] LOWER ASYMPTOTE'/'      [3] MAXIMUM VALUE OF Y'/
     2'      [4] MINIMUM VALUE OF Y'//' WHICH? ENTER NUMBER: [1] ')
```

```
941 FORMAT (' FOR THE NORMALIZED DATA, Y(min) = ',F8.4,/26X,
   1'Y(max) = ',F8.4,//' FOR THE GRAPH:'/
   2'     ENTER THE MINIMUM VALUE OF Y [',F8.4,'] ')
942 FORMAT ('+    ENTER THE MAXIMUM VALUE OF Y [',F8.4,'] ')
998 FORMAT (' GRAPH LIBRARY FAILURE AT KVAL = ',I2,' IRETCD = ', I5)
999 FORMAT (' UNABLE TO OPEN OR INTERPRET PLOT.DAT FILE')
    END
```

```
C
C                       ************************
C                       *                      *
C                       *   SUBROUTINE LOGPAR  *
C                       *                      *
C                       ************************
C
C                            VERSION 5/1/86
C
C
C "PRINTS" THE PARAMETERS, DEVIATIONS, AND UNITS AS STRING VARIABLES
C          FOR PRINTING ON PLOTS IN PPARM.
C "PRINTS" THE STANDARD DEVIATION TO A STRING TO BE PRINTED IN THE TOP
C          LINE OF THE PLOT.
C "PRINTS" THE MAXIMUM AND MINIMUM VALUES OF X INCLUDED IN THE FIT
C          AS A STRING TO BE PRINTED ON THE BOTTOM DISPLAY LINE IN PLOT
C
C INPUT:  XLO,XHI,VCV,STD,FXX
C OUTPUT: NPL,PARLIN,STDLIN,BLIN1
C
      SUBROUTINE LOGPAR(XLO,XHI,VCV,STD,NPL,PARLIN,STDLIN,FXX,BLIN1)
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      CHARACTER PARLIN(10)*30,STDLIN*30,FXX*5,BLIN1*50,B*40
      CHARACTER CNOTE*30
      CHARACTER NAME(9)*2
      CHARACTER F1*12,F2(0:9)*1,F3*7,FORM*80
      DIMENSION VCV(9,9)
      LOGICAL ALLVAR
      DATA NAME/' A',' B','X0','D0','AS','BS','AQ','BQ',' Q'/
      DATA F1/'(A2,'' ='',F8.'/
      DATA F2/'0','1','2','3','4','5','6','7','8','9'/
      DATA F3/',''  ±'',F'/
      DATA B/' '/
      DATA CNOTE/'* NOT VARIED IN THE L.S. FIT'/
C
      NPL=0
      ALLVAR=.TRUE.
      DO 220 I = 1,9
      IF (I .GT. 4 .AND. C(I) .EQ. 0.0) GO TO 220
      CVAL=C(I)
      NPL=NPL+1
      L=1
      BS=SQRT(VCV(I,I))
      DBASE=BS
      IF (DBASE .EQ. 0.0) DBASE=10*ABS(CVAL)
      L=0
      IF (DBASE .NE. 0.0) L=6-INT(ALOG10(DBASE)+5)
      L=MAX0(L,0)
      L=MIN0(L,5)
      IF (BS .EQ. 0.0) THEN
        ALLVAR=.FALSE.
        IF (I .EQ. 4) THEN
          CVAL=2.0*C(I)
          FORM = '(A2,'' <'',F8.'//F2(L)//','' *'')'
```

```
        ELSE
          FORM = F1//F2(L)//','' *'')'
        END IF
        WRITE (PARLIN(NPL),FORM) NAME(I),CVAL
      ELSE
        K=INT(ALOG10(DBASE)+1)
        K=MAX0(K,1)
        K=K+L+2
        FORM=F1//F2(L)//F3//F2(K)//'.'//F2(L)//')'
        WRITE (PARLIN(NPL),FORM) NAME(I),CVAL,BS
      END IF
  220 CONTINUE
      IF (ALLVAR) GO TO 230
      NPL=NPL+1
      PARLIN(NPL)=' '
      NPL=NPL+1
      PARLIN(NPL)=CNOTE
  230 K=INT(ALOG10(STD)+3)
      L=5-K
      L=MAX0(L,0)
      L=MIN0(L,5)
      IF (L .EQ. 1) K=5
      IF (L .GT. 1) K=L+3
      FORM='('' STANDARD DEVIATION ='',F'//F2(K)//'.'//F2(L)//')'
      WRITE (STDLIN,FORM) STD
      FORM='(''X(min) = '''//FXX//','''  X(max) = '''//FXX//')'
      WRITE (BLIN1,FORM) XLO,XHI
      L=LNBX(BLIN1)
      LB=40-L
      BLIN1=B(1:LB)//BLIN1(1:L)
      RETURN
      END
```

```
C                        *************************
C                        *                       *
C                        *   SUBROUTINE DPLOT    *
C                        *                       *
C                        *************************

C                            VERSION 6/11/86

C SWITCH TO GRAPHICS SCREEN AND CLEAR GRAPHICS SCREEN.
C CALCULATE MINIMUM AND MAXIMUM VALUES OF Y.  THESE WILL VARY
C DEPENDING ON WHETHER THE GRAPH HAS BEEN "ZOOMED" OR NOT.
C DRAW THE GRAPH OF THE OBSERVED POINTS AND THEIR STANDARD ERROR BARS.
C DETERMINE THE EMPTIEST QUADRANT OF THE SCREEN FOR PRINTING THE VALUES
C OF THE PARAMETERS.
C CALCULATE THE FORMAT FOR PRINTING OF CURSOR INFORMATION ON
C THE BOTTOM LINE OF THE GRAPH.
C CALCULATE THE SPACING TO BE USED FOR CURSORS AND MARKERS.

      SUBROUTINE DPLOT(LABEL,IW,N,X,Y,E,FLINE,YCUR)
      COMMON/PARAM/XVAL,YCALC,C(9),CIN(9),G(9)
      COMMON/LIMITS/NLO,NHI,NLOFIT,NHIFIT,IQMIN,ZOOM,VCV(9,9),CL975
      COMMON/DATA/NP,IFIO(300),XP(300),YP(300)
      COMMON/EXTREMA/XMIN,XMAX,YMIN,YMAX,YMAXIN(5),YMININ(5)
      CHARACTER FX*5,FY*5,FU*5,FLINE*80
      CHARACTER YLBL(5)*20,FL(0:8)
      LOGICAL EBAR(5),ZOOM
      DOUBLE PRECISION U
      DIMENSION X(300),Y(300),E(300)
      DIMENSION NQ(0:3),ISYM(-1:1),XB(2),YB(2)
      DATA YLBL /'     INTENSITY','DERIVATIVE INTENSITY',
     1'CALCULATED INTENSITY','NORMALIZED INTENSITY',
     2' STANDARD RESIDUALS'/
      DATA FL/'0','1','2','3','4','5','6','7','8'/
      DATA EBAR/.TRUE.,.FALSE.,.TRUE.,.TRUE.,.FALSE./
      DATA ISYM /11520,0,2304/

      ICON = 63744
      ISYM(-1)=11520
      IF (LABEL .EQ. 5) ISYM(-1)=11008
      NP=N
      CALL AGRAFO(IRETCD)

C  NLO = LOWEST POINT, NHI = HIGHEST POINT PLOTTED.

      DO 5 I=1,N
      NLO=I
    5 IF (X(NLO) .GE. XMIN) GO TO 6
    6 DO 7 I=1,N
      NHI=N+1-I
    7 IF (X(NHI) .LE. XMAX) GO TO 8
    8 IF (NHI .LT. NLO) NHI=NLO
      YMIN=YMININ(LABEL)
      YMAX=YMAXIN(LABEL)
      IF (LABEL .LT. 4) THEN
```

```
      IF (ZOOM .AND. LABEL .NE. 2) THEN
        XVAL=XMIN
        CALL DERIV
        Y1=YCALC
        U=0.0
        DO 10 J=1,9
        DO 10 K=1,9
10      U=U+G(J)*VCV(J,K)*G(K)
        U1=CL975*SQRT(U)
        XVAL=XMAX
        CALL DERIV
        Y2=YCALC
        U=0.0
        DO 11 J=1,9
        DO 11 K=1,9
11      U=U+G(J)*VCV(J,K)*G(K)
        U2=CL975*SQRT(U)
        IF (Y1 .LT. Y2) THEN
          Y1=Y1-U1
          Y2=Y2+U2
        ELSE
          Y1=Y1+U1
          Y2=Y2-U2
        END IF
        YMIN=MIN(Y1,Y2)
        YMAX=MAX(Y1,Y2)
      END IF
      DO 12 I=NLO,NHI
      YMIN=MIN(YMIN,Y(I))
12    YMAX=MAX(YMAX,Y(I))
      EXT = (YMAX-YMIN)/20
      YMAX=YMAX+EXT
      YMIN=YMIN-EXT
    END IF

C  DRAW THE AXES

    CALL AGRAF1(IRETCD,IW,0,XMIN,XMAX,YMIN,YMAX)
    IW=IW-2
    IF (IW .LT. 0) IW=0

C  LABEL THE Y AXIS

    CALL AGRAF2(IRETCD,21,IW,11,YLBL(LABEL))

C  PLOT THE DATA VALUES INDIVIDUALLY AND CALCULATE THE EMPTIEST
C  QUADRANT ON THE SCREEN WHERE THE VALUES OF THE PARAMETERS WILL
C  BE PRINTED.  NQ(I) COUNTS THE NUMBER OF DATA VALUES IN THE
C  FOUR QUADRANTS NUMBERED AS FOLLOWS:
C                    .........
C                    : 0 : 2 :
C                    :...:...:
C                    : 1 : 3 :
C                    :...:...:

C  THE QUADRANTS CONTAINING THE ASYMPTOTES ARE EACH GIVEN AN EXTRA
C  COUNT TO GIVE AN ADVANTAGE TO QUADRANTS NOT CONTAINING THE ASYMPTOTE
C
C  X AND Y ARE STORED IN XP AND YP TO BE USED FOR CURSOR POSITION

      XAV = (XMIN+XMAX)/2
      YAV = (YMIN+YMAX)/2
      J=0
      IF (C(1) .GT. C(2)) J=1
      NQ(0)=J
      NQ(1)=1-J
      NQ(2)=1-J
      NQ(3)=J
      EMIN=YMAX-YMIN
      EMAX=0.0
      DO 30 I=NLO,NHI
      XP(I)=X(I)
      YP(I)=Y(I)
      J=0
      IF (X(I) .GT. XAV) J=2
      IF (Y(I) .LT. YAV) J=J+1
      NQ(J) = NQ(J)+1
      IF (LABEL .NE. 3) THEN
        IL=IFIO(I)
        ICON=ISYM(IL)
      END IF
      XB(1)=X(I)
      YB(1)=Y(I)
      CALL AGRAF3(IRETCD,ICON,1,XB,YB)
      IF (EBAR(LABEL)) THEN
        YB(1)=Y(I)+E(I)
        YB(2)=Y(I)-E(I)
        XB(2)=XB(1)
        CALL AGRAF3(IRETCD,1,2,XB,YB)
      END IF
      EMAX=MAX(EMAX,E(I))
   30 EMIN=MIN(EMIN,E(I))
      IQMIN = 0
      DO 40 I=1,3
   40 IF (NQ(I) .LT. NQ(IQMIN)) IQMIN=I
      YCUR=(YMAX-YMIN)/20

C PRINT A PAIR OF ARROWS AT THE LIMITS OF THE POINTS INCLUDED IN THE
C FIT IF THESE ARE NOT THE FIRST AND LAST POINTS.
C CALCULATE YCUR, THE RANGE IN Y FOR POSITIONING THE CURSOR.

      IF (LABEL .NE. 3) THEN
        IF (NLOFIT .GT. 1) THEN
          XB(1)=X(NLOFIT)
          YB(1)=Y(NLOFIT)-YCUR
          CALL AGRAF3(IRETCD,24*256,1,XB,YB)
          YB(1)=Y(NLOFIT)+YCUR
          CALL AGRAF3(IRETCD,25*256,1,XB,YB)
        END IF
```

```
        IF (NHIFIT .LT. N) THEN
          XB(1)=X(NHIFIT)
          YB(1)=Y(NHIFIT)-YCUR
          CALL AGRAF3(IRETCD,24*256,1,XB,YB)
          YB(1)=Y(NHIFIT)+YCUR
          CALL AGRAF3(IRETCD,25*256,1,XB,YB)
        END IF
      END IF

C SET FORMAT FOR BOTTOM LINE CURSOR INFORMATION

      UNX = (XMAX-XMIN)/1000.0
      XBAS=MAX(ABS(XMIN),ABS(XMAX))
      K=5-INT(ALOG10(UNX)+5)
      IF (K .LT. 0) K=0
      IF (K .GT. 5) K=5
      L=INT(ALOG10(XBAS)+1)
      IF (L .LT. 0) L=0
      IF (L .GT. 6) L=6
      L1=K+L+2
      IF (L1 .LT. 5) L1=5
      IF (L1 .GT. 8) L1=8
      L2=K+L-6
      IF (L2 .LT. 0) L2=0
      L2=K-L2
      IF (L2 .LT. 0) L2=0
      L3=3-L-K
      IF (L3 .LT. 0) L3=0
      L2=MAX(L2,L3)
      FX=',F'//FL(L1)//'.'//FL(L2)
      YBAS=MAX(ABS(YMIN),ABS(YMAX))
      K=5-INT(ALOG10(EMIN)+5)
      IF (K .LT. 0) K=0
      IF (K .GT. 5) K=5
      L=INT(ALOG10(YBAS)+1)
      IF (L .LT. 0) L=0
      IF (L .GT. 6) L=6
      L1=K+L+2
      IF (L1 .LT. 5) L1=5
      IF (L1 .GT. 8) L1=8
      L2=K+L-6
      IF (L2 .LT. 0) L2=0
      L2=K-L2
      IF (L2 .LT. 0) L2=0
      L3=3-L-K
      IF (L3 .LT. 0) L3=0
      L2=MAX(L2,L3)
      FY=',F'//FL(L1)//'.'//FL(L2)
      K=5-INT(ALOG10(EMAX)+4)
      IF (K .LT. 0) K=0
      IF (K .GT. 5) K=5
      L=INT(ALOG10(EMAX))
      IF (L .LT. 0) L=0
      IF (L .GT. 5) L=5
      L=K+L+2
      FU=',F'//FL(L)//'.'//FL(K)
      IF (LABEL .NE. 3) FLINE='(''X = '''//FX//','' Y = '''//FY//',A5)'
      IF (LABEL .EQ. 3) FLINE='(''X = '''//FX//
     1',''  Y(CALC) = '''//FY//','' ± '''//FU//')'
      RETURN
      END



C
C                      *************************
C                      *                       *
C                      *   SUBROUTINE PPARM    *
C                      *                       *
C                      *************************
C
C                           VERSION 5/12/86
C
C PRINT THE VALUES OF THE PARAMETERS AS STRINGS IN A QUADRANT OF THE
C GRAPHICS SCREEN SPECIFIED BY NQUAD:
C
C                              .........
C                              : 0 : 2 :
C                              :...:...:
C                              : 1 : 3 :
C                              :...:...:
C
C
      SUBROUTINE PPARM(NQUAD,IW,NPL,PARLIN)
      CHARACTER PARLIN(10)*30
      ISCOL=IW+5
      IF (NQUAD .GT. 1) ISCOL=79-30
      J=NQUAD/2
      ISROW=2+(19-NPL)*(NQUAD-2*J)
      DO 40 I=1,NPL
      CALL AGRAF2(IRETCD,ISROW,ISCOL,0,PARLIN(I))
   40 ISROW=ISROW+1
      RETURN
      END
```

```
C                             **************
C                             *            *
C                             *   CURSOR   *
C                             *            *
C                             **************

C                            VERSION 6/8/86

C EACH TIME A NEW PLOT IS DRAWN, THE CURSOR INDEX NCUR IS TESTED.
C IF THE CURSOR IS ON (NCUR > 0), SUBROUTINE CURSOR IS CALLED.
C CURSOR TESTS WHETHER THE PREVIOUS PLOT WAS A CALCULATED PLOT
C OR AN OBSERVER PLOT AND IF THE PREVIOUS PLOT AND NEW PLOT ARE
C DIFFERENT, IT MATCHES THE NEW POSITION OF THE CURSOR WITH THE OLD
C POSITION AS CLOSELY AS POSSIBLE.  BEFORE RETURNING, CURSOR PLACES
C THE CURSOR ON THE SCREEN.

      SUBROUTINE CURSOR(IDNEW,IDOLD)
      COMMON/DATA/NP,IFIO(300),XP(300),YP(300)
      COMMON/CDATA/NCUR,ICUR,YCUR,X(300),XC(300),N,NC,XCR(2),YCR(2)
      IF (IDNEW .EQ. 3 .AND. IDOLD .NE. 3) THEN
        XVAL=X(NCUR)
        DMIN=ABS(XC(NC)-XC(1))
        DO 10 I=1,NC
        DIFF=ABS(XC(I)-XVAL)
        IF (DIFF .GT. DMIN) GO TO 11
   10   DMIN=DIFF
   11   NCUR=I-1
      ELSE IF (IDNEW .NE. 3 .AND. IDOLD .EQ. 3) THEN
        XVAL=XC(NCUR)
        DMIN=ABS(X(N)-X(1))
        DO 20 I=1,N
        DIFF=ABS(X(I)-XVAL)
        IF (DIFF .GT. DMIN) GO TO 21
   20   DMIN=DIFF
   21   NCUR=I-1
      END IF
      XCR(1) = XP(NCUR)
      XCR(2) = XP(NCUR)
      YCR(1) = YP(NCUR)+YCUR
      YCR(2) = YP(NCUR)-YCUR
      CALL AGRAF3(IRETCD,ICUR,2,XCR,YCR)
      RETURN
      END
```

```
C
C                        *************************
C                        *                       *
C                        *   SUBROUTINE TITLE    *
C                        *                       *
C                        *************************
C
C                            VERSION 3/20/86
C
C  CLEAR THE TOP LINE OF THE SCREEN AND PRINT THE TITLE OR THE HELP
C  MENU IN THE TOP LINE.
C
      SUBROUTINE TITLE(TTLON,KTYPE,TITLEF,DATLIN,STDLIN,XMARK)
      LOGICAL TTLON
      CHARACTER TITLEF*30,DATLIN*16,STDLIN*30,XMARK*14
      CHARACTER TLINE*80,HELP(5)*80,B*80
      DATA HELP/'[F1 PLOT] [F2 DERIV] [F3 CALC] [F4 NORM] [F9 ERR] [F10
     1PRINT] [HOME CURS] [END]',
     2'[F1-F4 PLOT] [F9 ERR] [HOME CURSOR OFF] [<--> CURSOR MOVE] [F5 MA
     3RK] [END]',
     4'[F1-F4 PLOT] [HOME CURSOR OFF] [SHIFT-F5 MARKER OFF] [F5 ZOOM]',
     5'[F1-F4 PLOT] [F9 ERR] [F10 PRINT] [HOME CURSOR] [END] [SHIFT-F5 U
     6NZOOM]',
     7'PRINTER NOT READY, NOT AVAILABLE, OR NOT COMPATIBLE - ANY KEY TO
     8CONTINUE'/
      DATA B/' '/
      IF (TTLON) THEN
        LT=LNBX(TITLEF)
        LD=LNBX(DATLIN)
        LS=LNBX(STDLIN)
        NB=80-LT-LD-LS
        N1=NB/2
        N2=NB-N1
        TLINE=TITLEF(1:LT)//B(1:N1)//DATLIN(1:LD)//B(1:N2)//STDLIN(1:LS)
      ELSE
        IF (KTYPE .EQ. 3) THEN
          LX=LNBX(XMARK)
          TLINE=XMARK(1:LX)//'  '//HELP(KTYPE)
        ELSE
          TLINE=HELP(KTYPE)
        END IF
        LH=80-LNBX(TLINE)
        TLINE=B(1:LH)//TLINE
      END IF
      CALL AGRAF2(IRETCD,0,0,0,TLINE)
      TTLON=.NOT.TTLON
      RETURN
      END
```

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NISTIR-88-3803	2. Performing Organ. Report No.	3. Publication Date MARCH 1989

4. TITLE AND SUBTITLE

Logistic Function Data Analysis Program: LOGIT

5. AUTHOR(S)

William H. Kirchhoff

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

A FORTRAN program has been written for the statistical analysis of experimental data in terms of an extended logistic function that includes non-horizontal asymptotes and asymmetry in the pre- and post-transition portions of growth and decay curves. The program is robust in that situations in which few or no data fall within the transition interval can be analyzed by the program. Individual weighting of the data is allowed for situations where the errors in experimental data are not uniform. The primary parameters describing the transition region include a location parameter, X_o, a width parameter, D_o, and an asymmetry parameter Q. Six more parameters describe the two quadratic asymptotic regions. The prgram does not require initial estimates for the parameters and provides statistical estimates of all parameters. A provision is included for the identification and exclusion of outliers. Sufficient information is given to allow the development of companion subroutines for the graphing of the function and its standard deviations as well as for displaying the original data with error bars. The purpose of the program is to provide a means for systematically parameterizing sigmoidal profiles for the comparison of measurements made with different instruments on different systesm and for the comparison of measurements with simulation models. The program is extensively documented.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

autocatalytic function; hyperbolic function; logistic function; sigmoid; statistical analysis

13. AVAILABILITY

☒ Unlimited
☐ For Official Distribution. Do Not Release to NTIS
☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

93

15. Price

$14.95

USCOMM-DC 6043-P80