GENFIT

A GENERAL LEAST SQUARES CURVE FITTING PROGRAM FOR MINI-COMPUTER

S. Shalev

INFORMAÇÃO IEA 57 CEN - AFR 16	JUNHO/1977

INF. IEA 57
CEN - AFR 16

JUNHO/1977

GENFIT

A GENERAL LEAST SQUARES CURVE FITTING PROGRAM FOR MINI-COMPUTER

S. Shalev

CENTRO DE ENGENHARIA NUCLEAR
Área de Física de Reatores

INSTITUTO DE ENERGIA ATÔMICA
SÃO PAULO – BRASIL

APROVADO PARA PUBLICAÇÃO EM ABRIL/1977

INSTITUTO DE ENERGIA ATÔMICA
Caixa Postal 11.049 (Pinheiros)
Cidade Universitária "Armando de Salles Oliveira"
SÃO PAULO — BRASIL

NOTA: Este trabalho foi conferido pelo autor depois de composto e sua redação está conforme o original, sem qualquer correção ou mudança.

CONTENTS

GENFIT
A GENERAL LEAST SQUARES CURVE FITTING PROGRAM FOR MINI-COMPUTERS

S. Shalev *

ABSTRACT

Genfit is a BASIC data processing program, suitable for small on line computers. Although initially programmed for an HP2116C computer with a 16 K magnetic memory punched tape input, teletype control and an HP graph plotter, it may be used with a wide range of similar mini computers.

In essence the program solves the curve fitting problem using the non-linear least squares method, the theory for which is given in some detail. A data set consisting of a series of points in the X-Y plane is fitted to a selected function whose parameters are adjusted to give the best fit in the least squares sence. The library of functions included in the program is easily extended.

Appropriate errors are specified for both the X and Y directions and each parameter in the fitting function may be designated as variable or constant. Convergence may be accelerated by modifying (or interchanging) the values of the constant parameters in accordance with results of previous calculations.

1 – INTRODUCTION

GENFIT is a data processing program suitable for small on-line computers. It was programmed for a HP2116C computer, with 16 K magnetic memory, punched tape input, teletype control and an HP-graph plotter, but is equally suitable for a wide range of similar mini-computers. The language is HP-BASIC.

Essentially the program solves the curve-fitting problem using the non-linear least squares method. A data set, consisting of a series of points in the X-Y plane, is fitted to a selected function whose parameters are adjusted to give the best fit in the least squares sense. A library of functions is incorporated in the program, and the addition of other functions is extremely simple. A variety of options are available for data input and output.

The points comprising the data set are weighted according to their relative precisions. For this purpose standard errors may be specified in both the X and Y directions.

Each parameter in the fitting function may be designated as variable or constant. Furthermore, the value at which a given parameter is held constant may be determined from a previous calculation in which it was variable. In this way convergence can usually be reached even for quite intractable non-linear functions.

2 – INPUT

There are two types of input instructions in the program, DATA INPUT, and CONTROL

*IAEA Technical Assistance Expert, Division of Nuclear Engineering, Instituto de Energia Atômica, São Paulo, Brasil.

INPUT. In the former, the data set of N points is provided, each point consisting of X and Y – coordinates and errors (variances) in the X and Y – directions. Control input consists of instructions concerning the choice of function, number and mode of parameters, output instructions for printing intermediate and final results, and the type of graph required. Input instructions are of the conversational type, where the operator responds to requests for information from the computer. In most cases the data input routines are self-checking, rejecting information which is meaningless or inappropriate.

The maximum number of data points is 150. However, options 2 and 3 provide the possibility of grouping data, so that although the final number of grouped points must be less than 150 there is no restriction on the number of ungrouped points.

2.1 – DATA INPUT

Three options (index L3) are incorporated in the program. All require the operator to state the number of data points, and they provide different possibilities of inputting the data set.

OPTION 1:

Teletype – This is the most general option, and also the simplest. The operator is asked to provide the number of points N, and for each point the (X,Y) coordinate and the errors (variances) in the X and Y directions. There are no restrictions on the values of X or Y, on the order in which the points are input, or on the errors (except that data with zero error on both the X and Y axis will be rejected). Regular spacing between the points is not necessary, and more then one point may have the same X (or Y) value. However, since the least squares method is used, there must be more points than variable parameters.

OPTION 2:

Punched tape – This option accepts punched tape through the optical reader. It is assumed that this tape has been obtained from a multichannel analyser, and consists of Y-values only, in increasing order along the X-axis. The first value is assumed to have special importance and is printed out without being transferred to the data set. The operator is asked to provide the number of points N, the first X-value (corresponding to the second Y-value on the tape), the X-step (difference between consecutive X-values) and the variance on all X-points (assumed to be equal for all the points). Points may be grouped, and the group size L1 is requested (minimum value = 1). Values of L1 and N will not be accepted which result in a number of groups greater than 150.

OPTION 3:

Data Instructions – This option is useful where several calculations are to be made on the same set, but no punched tape is available. The data is stored in the program in the form of data instructions (starting at line 6000), and read from there as required. Otherwise, the instructions are the same as Option 2.

2.2 – CONTROL INPUT

The operator is asked to denote a function number (H) and the number of parameters (K). The functions are stored in the library with index numbers in the range 1 – 9 (see section 4). Many functions are in the form of series, and may be truncated after one, two or more terms. Hence it is important to choose a number of parameters corresponding to whole terms in the series. The maximum number of parameters is 10, but this may be changed by a altering lines 70, 196 and 197.

The next control index (H1) gives three options with respect to the initial values of the parameters. If the function is linear, and errors have been specified only on the Y axis, then the program will probably converge even if all the parameters are initially set equal to zero. In other cases, it will be necessary to give an initial value for each parameter. if a previous calculation has been made, the final results of which are acceptable as initial values for the current calculation, this possibility is allowed for under the heading space "parameters held over".

The use of Option H1 = 0 (parameters initially set to zero) is forbidden in the case of zero errors on the Y-values. In this rather unlikely case, it is necessary to use H1 = 1 and provide non-zero initial values for the parameters.

The index K is equal to the total number of parameters to be used in the function. Each parameter may be held at a constant value or be considered variable, with a final value to be determined by the least squares fitting procedure. This option is controlled by index W1, which is the number of parameters to be held constant. Obviously $0 < W1 < K$. For the case where W1 is not zero, the operator will be asked to give the index value of each constant parameter, and the value at which it is to be held constant.

The type of output is determined by the indeces B2 and B1, which permit the printing of the matrices or the parameter values during each iteration, if this is required. If convergence is not reached in 20 iterations, the program will stop. After convergence, the final values of the parameters are printed with their errors, together with the sum of the squares of the residuals. Further options permit the tabulation of all the input data, fitted curve and residuals (index B3), and a graph plotted with a linear or logarithmic ordinate scale (Index E5). A final option gives the choice of plotting each point, plotting the fitted function, and also plotting the error curves (index B4).

3 – THE FUNCTION LIBRARY

Nine functions may be incorporated in the library, and called into use by means of the control index H. The program has been writen in such a way that the addition of functions is a relatively simple procedure. Each function has been allocated a section of the memory:

Function Number (Index H)	Memory Location	Function	Permitted Number of parameters (Index K)
1	1100 – 1199	$Y = \sum_{i=1}^{10} P_i X^{i-1}$	1,2,3,4,5,6,7,8,9,10
2	1200 – 1299	$Y = P_1 + \sum_{i=1}^{4} P_{2i} \exp(P_{2i}$ [illegible] $X)$	3,5,7,9
3	1300 – 1399	spare	
4	1400 – 1499	$Y = \sum_{i=2}^{9} P_i \mathrm{Sin}[\frac{(i-1)\pi X}{P_1}]$	2,4,6,8,10
5	1500 – 1599	$Y = P_2 \mathrm{Sin}(\frac{\pi X}{P_1}) + P_3$	3
6	1600 – 1699	Spare	
7	1700 – 1799	Spare	
8	1800 – 1899	Spare	
9	1900 – 1999	Spare	

The input control indeces H and K define the choice of function and the maximum number of parameters to be used. Of course, the value of K must be a permitted value (see above table).

The addition of new functions to the library entails inserting the appropriate instructions in the reserved memory locations. The first instructions should be in the first reserved location (i.e. 1600 for function number 6) and the last instructions should be "GOTO 1020". In addition, instructions should be given defining the function F(I), its differentials G(I) with respect to each parameter P(J), and its differential (Z) with respect to X.

4 – CURVE FITTING BY THE METHOD OF LEAST SQUARES

4.1 – Statement of Problem

We start with a set of m data points in n + 1 dimensional space $(X_{1i}, X_{2i}, \ldots, X_{ji}, \ldots, X_{ni}, Y_i)$ where i = 1, . . . , m and with an assumed functional relationship with n independent variables and p parameters

$$y = f(x_1, x_2, \ldots, x_n, a_1, a_2, \ldots, a_p) .$$

We are interested in finding the values of the p parameters which will give a "best fit" to the data.

"Best fit" is defined as follows. With each value of each variable there is an associated uncertainty $\sigma_i(X_j)$ where $i = 1, \ldots, m$, $j = 1, \ldots, n$. The reciprocals of the squares of these uncertainties are defined as the weights:

$$W_i(Y) \equiv \frac{1}{\sigma_i^2(Y)}$$

$$W_i(X_j) \equiv \frac{1}{\sigma_i^2(X_j)} .$$

The residual $R_i(x_j)$ is defined as the difference between the observed value of the variable and its value as computed from the fitting function f.

$$R_i(y) \equiv Y_i - y_i$$

$$R_i(x_j) \equiv X_{ji} - x_{ji}$$

The parameters which give the best fit to the function are those which minimize the function S defined as the weighted sum of the squares of the residuals:

$$S = \sum_{i=1}^{m} [\, W_i(Y) R_i^2(y) + \sum_{j=1}^{m} W_i(X_j) R_i^2(x_j) \,] \qquad (1)$$

4.2 – Summary of Mathematics of solution*

To solve the problem we must make initial guesses of the parameters. The initial guess of the k^{th} parameter is denoted as a_{k_o}. If the problem is linear, that is if the function f is linear with respect to the parameters, then the initial guesses are of no importance and convergence to a solution is reached in one iteration regardless of their value. In nonlinear problems choice of initial guesses can be critical and drastically affect convergence.

The conditional function is defined as

$$F^i \equiv y_i - f(x_{1i}, x_{2i}, \ldots, x_{ji}, \ldots, x_{ni}, a_1, \ldots, a_p) \tag{2}$$

and

$$F^i = 0 \quad \text{for } i = 1, 2, \ldots, m .$$

The estimated value of F^i is

$$F^i_o = Y_i - f(X_{1i}, \ldots, X_{ni}, a_{10}, \ldots, a_{p_0}) \tag{3}$$

Where X_{ji} and Y_i are the data points and

$$F^i_o \neq 0 \text{ for } i = 1, 2, \ldots, m .$$

If F^i is expanded in a Taylor series about the data points and initial guesses of parameters and terms of higher order are neglected:

$$0 = F^i = F^i_o + \frac{\partial F^i}{\partial y_i}(y_i - Y_i) + \sum_j \frac{\partial F^i}{\partial x_{ji}}(x_{ji} - X_{ji}) + \sum_k \frac{\partial F^i}{\partial a_k}(a_k - a_{k_o})$$

and therefore

$$F^i_o = \frac{\partial F^i}{\partial y_i} R_i(y) + \sum_j \frac{\partial F^i}{\partial x_{ji}} R_i(x_j) + \sum_k \frac{\partial F^i}{\partial a_k} A_k \tag{4}$$

where

$$A_k = a_{k_o} - a_k . \tag{4a}$$

* For a fuller exposition see WOLBERG, J. R. *Prediction analysis.* New York, Van Nostrand, 1967.

The residuals in equation (4) can be varied slightly with a consequent variation in the parameters but in such a way that the F^i_o do not change because they are not functions of the residuals. The differential of equation (4) is therefore:

$$\frac{\partial F^i}{\partial y_i}\delta R_i(y) + \sum_{j=1}^{n}\frac{\partial F^i}{\partial X_{ji}}\delta R_i(x_j) + \sum_{k=1}^{p}\frac{\partial F^i}{\partial a_k}\delta A_k = 0 \;;\; i = 1,\ldots, m \tag{5}$$

Considering equation (1), since S is to be minimized, small variations of the residuals will not affect it, therefore:

$$\frac{1}{2}\,\delta S = 0 = \sum_{i=1}^{m}[W_i(Y)\,R_i(y)\,\delta R_i(y) + \sum_{j=1}^{n} W_i(X_j)\,R_i(x_j)\,\delta R_i(x_j)] \tag{6}$$

Since we are considering an extreme problem where equation (6) is to be minimized and equations (5) are the m restrictions, we will use the method of Lagrange multipliers to arrive at the solution. Multiplying equations (5) by m Lagrange multipliers, $\lambda_i (i = 1, \ldots, m)$ and subtracting these m equations from equation (6) we get:

$$\sum_i (W_i(Y)\,R_i(y) - \lambda_i\frac{\partial F^i}{\partial y_i})\,\delta R_i(y) + \sum_i\{\sum_j (W_i(X_j)\,R_i(x_j) - \lambda_i\frac{\partial F^i}{\partial x_{ji}}\;\delta R_i(x_j)\}$$

$$-\sum_k(\sum_i\lambda_i\frac{\partial F^i}{\partial a_k})\,\delta A_k = 0 \tag{7}$$

In order that this equation be satisfied, each coefficient must equal zero.

$$R_i(y) = \frac{\lambda_i\frac{\partial F^i}{\partial y_i}}{W_i(Y)} \qquad i = 1,\ldots, m \tag{8}$$

$$R_i(x_j) = \frac{\lambda_i\frac{\partial F^i}{\partial x_{ji}}}{W_i(X_j)} \qquad \begin{matrix} i = 1,\ldots, m \\ j = 1,\ldots, n\end{matrix} \tag{9}$$

$$\sum_{i=1}^{m}\lambda_i\frac{\partial F^i}{\partial a_k} = 0 \qquad k = 1,\ldots, p \tag{10}$$

Substituting (8) and (9) into equation (4) we get:

$$F^i_o = \frac{\lambda_i}{W_i(Y)}\left(\frac{\partial F^i}{\partial y_i}\right)^2 + \sum_j\frac{\lambda_i}{W_i(X_j)}\left(\frac{\partial F^i}{\partial x_{ji}}\right)^2 + \sum_k\frac{\partial F^i}{\partial a_k}A_k \qquad i = 1,\ldots, m \tag{11}$$

Equations (10) and (11) are a set of m + p equations. Solving for λ_i in equations (11) and substituting it into equation (10) gives the set of p normal equation needed to solve for the p A_k's.

$$\begin{array}{l} A_1 \sum_i \frac{F^i_{a_1} F^i_{a_1}}{L_i} + A_2 \sum_i \frac{F^i_{a_1} F^i_{a_2}}{L_i} + \ldots + A_p \sum_i \frac{F^i_{a_1} F^i_{a_p}}{L_i} = \sum_i \frac{F^i_{a_1} F^i_o}{L_i} \\ \vdots \\ A_1 \sum_i \frac{F^i_{a_p} F^i_{a_1}}{L_i} + A_2 \sum_i \frac{F^i_{a_p} F^i_{a_2}}{L_i} + \ldots + A_p \sum_i \frac{F^i_{a_p} F^i_{a_p}}{L_i} = \sum_i \frac{F^i_{a_p} F^i_o}{L_i} \end{array} \quad (12)$$

where

$$F^i_{a_k} \equiv \frac{\partial F^i}{\partial a_k}$$

$$L_i \equiv \frac{\left(\frac{\partial F^i}{\partial y_i}\right)^2}{W_i(Y)} + \sum_{j=1}^{n} \frac{\left(\frac{\partial F^i}{\partial x_{ji}}\right)^2}{W_i(X_j)} \qquad i = 1, \ldots, m$$

$$= \sigma_i^2(Y) + \sum_{j=1}^{n} \left(\sigma_i(X_j) \frac{\partial F^i}{\partial x_{ji}}\right)^2 \quad (13)$$

where

$$\frac{\partial F^i}{\partial y_i} = 1 \text{ from equation (2).}$$

The p normal equations (12) can be written in matrix form

$$CA = V$$

where

$$C_{k\ell} = C_{\ell k} = \sum_{i=1}^{m} \frac{F^i_{a_\ell} F^i_{a_k}}{L_i} \quad (14)$$

$$V_k = \sum_{i=1}^{m} \frac{F^i_{a_k} F^i_o}{L_i} \tag{15}$$

The A_k's are then easily solved for by inverting the matrix C:

$$C^{-1} CA = A = C^{-1} V$$

or

$$A_k = \sum_{j=1}^{p} C^{-1}_{k_j} V_j \qquad k = 1, \ldots, p. \tag{16}$$

From equation (4a) the parameters a_k are:

$$a_k = a_{k_o} - A_k \,. \tag{17}$$

As mentioned earlier, if the problem is non-linear, these a_k will not be the final solution but are taken as the first guesses of the next iteration. The problem is said to have converged to a solution when the condition

$$\left| \frac{A_k}{a_k} \right| < \varepsilon \qquad k = 1, \ldots, p$$

is satisfied for all of the parameters for some arbitrary value of ε. To speed convergence, an acceleration factor γ is sometimes used and the new values of the parameters are:

$$a_k = a_{k_o} - \gamma A_k \qquad 0 < \gamma \leq 1. \tag{18}$$

4.3 – Estimation of Uncertainties of the Parameters

It is often of interest to know the uncertainties (or standard deviations) in the values of the parameters found by the least squares fit method. The uncertainties are defined as

$$\sigma_{a_k} \equiv \sqrt{(a_k - \alpha_k)^2_m} \qquad k = 1, \ldots, p \tag{19}$$

where

p = number of parameters

a_k = the calculated value of the k^{th} parameter

α_k = the true value of the k^{th} parameter.

Because the observed values of the variables differ from the true values, the α_k differ from a_k. In the above equation m denotes the mean value of the square of the difference if the experiment is done many times.

The derivation of the expression for σ_{ak} is long and tedious and will not be presented here*. It depends, however, on two assumptions: first, that the second and higher order terms in the Taylor series expansion of

$$f(X_{1i}, \ldots, X_{ni}, a_1, \ldots, a_p) - f(\xi_{1i}, \ldots, \xi_{ni}, \alpha_1, \ldots, \alpha_p)$$

are negligible where ξ_{ki} is the <u>exact</u> value of the k^{th} independent variable at the i^{th} point and α_k is the true value of the k^{th} parameter. This is a reasonable assumption unless the errors are large. The second assumption is that the errors in the variables are uncorrelated, which is usually the case.

The equation for σ_{a_k} finally obtained, is

$$\sigma_{a_k} = \sqrt{\frac{S}{n-p}} \sqrt{C_{kk}^{-1}} \qquad k = 1, \ldots, p \qquad (20)$$

This is an approximation since an actual mean value is not used but only an unbiased estimate, that is the value taken from the current experiment. In equation (20) S is the weighted sum of the squares of the residuals and is equal to

$$S = \sum_{i=1}^{m} \frac{(F_o^i)^2}{L_i} , \qquad (21)$$

m is the number of data points, p is the number of parameters, and C_{kk}^{-1} is the k^{th} element in the diagonal of the inverse of the matrix C. m − p is the number of degrees of freedom of the problem.

5 – IDENTIFICATION OF BASIC CHARACTERS

Some BASIC characters are given a constant value, and remain unchanged throughout the program (e.g. A9 = π). Others are input values, and may be changed by the operator (e.g. K = number of parameters). A number of characters are used as internal flags (e.g. E2 = number of iterations), or as loop indeces (I, J, L). Many characters serve as temporary storage (e.g. B, L1 etc.). The following characters can be correlated with the mathematical expression of section 4.

* For details see Wolberg, pp 54-60

BASIC characters	Mathematical expression	BASIC characters	Mathematical expression
X(I)	X_i	M(J,L)	$C_{k\ell}$
Y(I)	Y_i	A(J)	A_k
Q(I)	$\frac{1}{\sigma_i^2(X)}$	C(J,L)	C^{-1}_{kj}
R(I)	$\frac{1}{\sigma_i^2(Y)}$	N	m
F(I)	F^i	K	p
G(J)	F^i_{ak}	D1	ε
Z	$\frac{\partial F^i}{\partial X_i}$	A8	γ
V(J)	V_k	S2	S

6 – SUGESTIONS AND HINTS

The program GENFIT should be used with the R P T object tape, which permits the input of punched paper data tapes. If the RPT tape has not been read, then error messages will be given for lines 5050 and 5073 when the program is read in. If no punched data tapes will be used, these error messages may be ignored.

The convergence criterion D1 is set to 10^{-2} in line 84. In some cases it may be desired to change this value (D1 is the maximum permitted percentage change in the parameters for convergence).

The acceleration factor A8 is set to 1 in line 86. For cases where convergence cannot be reached due to instability in the iteration process, it may be an advantage to reduce A8 to 0.5 or some other value. Naturally the number of iterations will then increase, and the maximum number permitted (set to 20 in line 461) may have to be modified.

RESUMO

GENFIT é um programa básico de processamento de dados, apropriado para pequenos computadores atuais. Apesar de ter sido programado inicialmente para um computador HP2116C com memória magnética de 16 k, entrada por fita perfurada, teletipo e com "plotter" HP, o programa pode ser utilizado numa extensa gama de mini-computadores similares.

Em essência, o programa resolve o problema de ajuste de curvas, utilizando o método de mínimos quadrados não-linear, sendo a teoria apresentada com algum detalhe. Um conjunto de dados, que consiste numa série de pontos no plano X-Y, é ajustado por uma função selecionada, cujos parâmetros fornecem o melhor ajuste possível no critério de mínimos quadrados. A biblioteca de funções do programa pode ser estendida facilmente.

São especificados erros apropriados para as direções X e Y e cada parâmetro na função de ajuste pode ser tomado como variável ou constante. A convergência pode ser acelerada modificando (ou intercambiando) os valores das constantes, de acordo com resultados de cálculos prévios.

```
10  REM     GENFIT     A GENERAL LEAST SQUARES FITTING PROGRAM
12  REM     USE WITH RPT OBJECT TAPE
15  PRINT
20  PRINT "GENFIT       CURVE FITTING BY LEAST SQUARES"
21  PRINT
60  DIM X[150],Y[150],Q[150],R[150],F[150],T[150]
61  DIM S[150],Z[150]
70  DIM A[10],G[10],P[10],V[10],W[10],C[10,10],M[10,10]
84  LET D1=1.00000E-02
85  REM   D1 IS THE CONVERGENCE CRITERION
86  LET A8=1
87  REM A8 = ACCELERATION FACTOR
90  LET A9=3.14159
91  LET L9=0
92  REM L9 CYCLES DATA REJECTION (SUB 5000)
93  LET J9=1
94  REM   J9=0 SUPRESSES INPUT FOR DATA REJECTION CYCLE
100  LET E2=1
101  LET E9=0
102  REM E9 INDICATES CONVERGENCE
103  IF L9>0 THEN 108
104  PRINT "DATA FROM TELETYPE(1), PUNCHED TAPE(2) OR READ(3)";
105  INPUT L3
106  IF L3=1 THEN 110
107  IF L3>3 THEN 104
108  GOSUB 5000
109  GOTO 170
110  LET J9=1
111  PRINT "NUMBER OF DATA POINTS";
112  INPUT N
113  IF N<255 AND N>1 THEN 120
114  IF N<2 THEN 111
115  PRINT "MAXIMUM NUMBER OF POINTS IS 255"
116  GOTO 111
117  STOP
120  PRINT "FOR EACH DATA POINT GIVE X,XVAR,Y,YVAR"
122  PRINT
125  FOR I=1 TO N
127  PRINT I;
130  INPUT X[I],Q[I],Y[I],R[I]
132  IF Q[I] OR R[I] THEN 139
134  PRINT "ZERO ERROR ON BOTH X AND Y NOT PERMITTED"
136  GOTO 127
139  NEXT I
170  IF J9=0 THEN 319
180  PRINT
182  PRINT "FUNCTION NO.";
184  INPUT H
186  IF H<1 THEN 182
187  IF H>9 THEN 182
192  PRINT "NUMBER OF PARAMETERS =";
194  INPUT K
195  IF K<1 THEN 192
196  IF K<11 THEN 199
```

```
197  PRINT " MAXIMUM NUMBER OF PARAMETERS IS 10"
198  GOTO 192
199  REM
200  PRINT "PARAMETERS SET TO ZERO(0) OR INPUT(1) OR HELD OVER(2)";
202  INPUT H1
205  IF H1=1 THEN 230
206  IF H1=2 THEN 255
207  IF H1#0 THEN 200
210  LET E8=0
211  FOR I=1 TO N
212  IF R[I]#0 THEN 216
214  LET E8=1
216  NEXT I
217  IF E8=0 THEN 250
220  PRINT
221  PRINT "ZERO Y ERRORS NOT PERMITTED WITH THIS OPTION"
223  GOTO 200
230  FOR J=1 TO K
232  PRINT J;
234  INPUT P[J]
236  NEXT J
240  GOTO 255
250  FOR J=1 TO K
251  LET P[J]=0
252  NEXT J
255  FOR J=1 TO K
256  LET V[J]=0
257  NEXT J
262  PRINT "HOW MANY PARAMETERS TO BE HELD CONSTANT  ";
263  INPUT W1
264  IF W1=0 THEN 300
265  IF K-W1>0 THEN 268
266  PRINT "ALL PARAMETERS MAY NOT BE CONSTANT"
267  GOTO 262
268  PRINT "FOR EACH CONSTANT PARAMETER GIVE ITS INDEX AND VALUE"
270  FOR J=1 TO W1
272  INPUT L,B
273  IF L>0 AND L<K+1 THEN 276
274  PRINT "INVALID INDEX"
275  GOTO 272
276  LET V[L]=1
277  LET P[L]=B
278  NEXT J
300  PRINT "PRINT EACH ITERATION ?    NO=0   YES=1   ";
302  INPUT B2
304  IF B2=1 THEN 310
306  IF B2#0 THEN 300
308  LET B1=0
309  GOTO 316
310  PRINT "PRINT THE MATRICES   ?    NO=0   YES=1   ";
312  INPUT B1
314  IF B1<0 OR B1>1 THEN 310
316  IF J9=0 THEN 319
317  GOSUB 3900
```

```
319  LET L=K-W1
320  MAT A=ZER[L]
321  MAT V=ZER[L]
322  MAT C=ZER[L,L]
323  MAT M=ZER[L,L]
324  LET S2=0
330  GOSUB 1000
331  REM      SUB 1000 PREPARES MATRICES M,V
338  IF B1=0 THEN 350
339  PRINT
340  PRINT "MATRIX M"
341  MAT  PRINT M
350  MAT C=INV(M)
352  MAT A=C*V
353  LET E2=E2+1
359  IF B1=0 THEN 370
364  PRINT "MATRIX C"
365  MAT  PRINT C
367  PRINT "VECTOR A"
368  MAT  PRINT A
370  LET S2=0
380  GOSUB 1000
381  REM     SUB 1000 CALCULATES T(I)
400  IF E2>2 THEN 415
404  PRINT
405  PRINT "PARAMETER      OLD VALUE          NEW";
408  PRINT "         ERROR (ST.DEV)"
410  PRINT
415  LET E=0
417  LET S1=S2/(N-K+W1)
420  LET B=0
421  LET J=1
422  FOR L=1 TO K
425  IF W[L]=1 THEN 455
426  LET B=P[L]-A[J]*A8
428  IF P[L]=0 THEN 437
430  LET D=A[J]*100/P[L]
431  LET D=ABS(D)
432  IF D<D1 THEN 439
436  GOTO 438
437  LET D=100
438  LET E=1
439  LET S3=SQR(S1*C[J,J])
446  IF E9=0 AND B2=0 THEN 452
450  PRINT L,P[L],B,S3
452  LET P[L]=B
453  LET J=J+1
454  GOTO 457
455  IF E9=0 AND B2=0 THEN 457
456  PRINT L,P[L],"CONSTANT"
457  NEXT L
458  IF B2=0 THEN 460
459  PRINT
460  IF E=0 THEN 500
```

```
461  IF E2<20 THEN 465
462  PRINT
463  PRINT "DOES NOT CONVERGE AFTER 20 ITERATIONS"
464  STOP
465  REM   RE-ITERATE
466  GOTO 319
500  REM CONVERGED
505  IF E9>0 THEN 510
506  LET E9=1
507  GOTO 319
510  PRINT
512  PRINT "CONVERGED IN",E2-1,"ITERATIONS"
513  PRINT
515  IF B3=0 THEN 602
520  PRINT
522  PRINT "  X              Y                 YFITTED";
523  PRINT "       Y-YFITTED        FIT ERROR"
530  PRINT
540  FOR I=1 TO N
545  LET B=SQR(S1*T[I])
550  PRINT X[I],Y[I],F[I],Y[I]-F[I],B
560  NEXT I
600  PRINT
602  PRINT "SUM OF DEVS.SQ =",S2
605  PRINT "SIGMA  SQ.     =",S1
610  PRINT
710  IF E5=0 THEN 990
720  GOSUB 3000
990  PRINT
991  PRINT
992  LET J9=0
995  GOTO 100
1000  FOR I=1 TO N
1006  LET F[I]=T[I]=0
1007  LET Z=0
1011  IF H=1 THEN 1100
1012  IF H=2 THEN 1200
1013  IF H=3 THEN 1300
1014  IF H=4 THEN 1400
1015  IF H=5 THEN 1500
1016  IF H=6 THEN 1600
1017  IF H=7 THEN 1700
1018  IF H=8 THEN 1800
1019  IF H=9 THEN 1900
1020  LET J=1
1022  FOR L=1 TO K
1024  IF V[L]=1 THEN 1027
1025  LET G[J]=G[L]
1026  LET J=J+1
1027  NEXT L
1029  LET O=F[I]-Y[I]
1030  LET L1=Z*Z*Q[I]
1031  LET L2=R[I]
1035  LET B=1/(L1+L2)
```

```
1040  FOR J=1 TO K-W1
1045  FOR L=1 TO K-W1
1050  LET M[J,L]=M[J,L]+B*G[J]*G[L]
1052  LET T[I]=T[I]+G[J]*G[L]*C[J,L]
1055  NEXT L
1060  LET V[J]=V[J]+B*O*G[J]
1070  NEXT J
1075  LET S2=S2+O*O*B
1090  NEXT I
1099  RETURN
1100  IF E2>1 OR I>1 THEN 1130
1101  IF J9=0 THEN 1130
1104  PRINT
1105  PRINT "FUNCTION IS Y = P1 + P2.X + P3.X↑2 + ..."
1106  PRINT
1130  FOR J=1 TO K
1135  LET G[J]=0
1140  IF X[I]=0 THEN 1150
1145  LET G[J]=X[I]↑(J-1)
1150  LET G[1]=1
1155  LET F[I]=F[I]+P[J]*G[J]
1160  NEXT J
1165  FOR J=1 TO K-1
1167  LET Z=Z+J*P[J+1]*G[J]
1170  NEXT J
1199  GOTO 1020
1200  IF E2>1 OR I>1 THEN 1225
1201  IF J9=0 THEN 1225
1205  PRINT
1210  PRINT "FUNCTION IS Y = P1 + P2.EXP(P3.X) + P4.EXP(P5.X) + ..."
1220  PRINT
1225  LET G[1]=1
1226  LET Z=0
1227  LET F[I]=P[1]
1230  FOR J=1 TO (K-1)/2
1240  LET G[2*J]=EXP(P[2*J+1]*X[I])
1255  LET G[2*J+1]=P[2*J]*X[I]*EXP(P[2*J+1]*X[I])
1260  LET Z=Z+P[2*J]*P[2*J+1]*EXP(P[2*J+1]*X[I])
1265  LET F[I]=F[I]+P[2*J]*EXP(P[2*J+1]*X[I])
1270  NEXT J
1299  GOTO 1020
1300  GOTO 182
1400  IF E2>1 OR I>1 THEN 1430
1402  IF J9=0 THEN 1430
1404  PRINT
1405  PRINT "FUNCTION IS Y = P2.SIN(#.X/P1) + P3.SIN(2#.X/P1) + ..."
1406  PRINT "       WHERE # = 3.14159"
1407  PRINT
1430  LET G[1]=0
1431  FOR J=2 TO K
1435  LET B=COS((J-1)*A9*X[I]/P[1])
1436  LET C=(J-1)*A9*X[I]*P[J]/(P[1]↑2)
1437  LET G[1]=G[1]-B*C
1440  LET G[J]=SIN((J-1)*A9*X[I]/P[1])
```

```
1445  LET F[I]=F[I]+P[J]*G[J]
1450  LET Z=Z+(J-1)*P[J]*(A9/P[1])*COS((J-1)*A9*X[I]/P[1])
1470  NEXT J
1499  GOTO 1020
1500  IF E2>1 OR I>1 THEN 1530
1502  IF J9=0 THEN 1530
1504  PRINT
1505  PRINT "FUNCTION IS Y = P2.SIN(Ø.X/P1) + P3"
1506  PRINT "       WHERE Ø = 3.14159"
1507  PRINT
1530  LET G[1]=0
1531  LET J=2
1535  LET B=COS((J-1)*A9*X[I]/P[1])
1536  LET C=(J-1)*A9*X[I]*P[J]/(P[1]↑2)
1537  LET G[1]=G[1]-B*C
1540  LET G[J]=SIN((J-1)*A9*X[I]/P[1])
1545  LET F[I]=F[I]+P[J]*G[J]+P[3]
1550  LET Z=Z+(J-1)*P[J]*(A9/P[1])*COS((J-1)*A9*X[I]/P[1])
1560  LET G[3]=1
1599  GOTO 1020
1600  GOTO 182
1700  GOTO 182
1800  GOTO 182
1900  GOTO 182
3000  REM    PLOTTING SUBROUTINE
3002  LET U6=200
3003  LET U5=3
3005  LET M1=M2=M3=M4=1
3010  FOR I=1 TO N
3015  LET S[I]=Y[I]+SQR(R[I])
3020  LET Z[I]=Y[I]-SQR(R[I])
3025  IF E5=1 THEN 3040
3030  LET Y[I]=LOG(Y[I])
3032  LET S[I]=LOG(S[I])
3034  LET Z[I]=LOG(Z[I])
3036  LET F[I]=LOG(F[I])
3040  IF Y[I]>Y[M1] THEN 3060
3050  LET M1=I
3060  IF Y[I]<Y[M2] THEN 3080
3070  LET M2=I
3080  IF F[I]>F[M3] THEN 3100
3090  LET M3=I
3100  IF F[I]<F[M4] THEN 3120
3110  LET M4=I
3120  NEXT I
3130  LET N1=Z[M1]
3140  LET N2=S[M2]
3142  IF N1<F[M3] THEN 3146
3144  LET N1=F[M3]
3146  IF N2>F[M4] THEN 3150
3148  LET N2=F[M4]
3150  LET M1=M2=1
3160  FOR I=1 TO N
3165  IF X[I]>X[M1] THEN 3175
```

```
3170 LET M1=I
3175 IF X[I]<X[M2] THEN 3185
3180 LET M2=I
3185 NEXT I
3200 LET N5=X[M1]-U5*SQR(Q[M1])
3210 LET N6=X[M2]+U5*SQR(Q[M2])
3211 LET N7=N6-N5
3212 IF Q[M1]>0 THEN 3216
3214 LET N5=N5-N7/20
3216 IF Q[M2]>0 THEN 3220
3218 LET N6=N6+N7/20
3220 PRINT "XMAX=",N6
3225 PRINT "XMIN=",N5
3230 PRINT "YMAX=",N2
3235 PRINT "YMIN=",N1
3240 IF E5=1 THEN 3255
3250 PRINT "GRAPH HAS LOG SCALE ON Y-AXIS"
3255 LET N7=N6-N5
3256 LET N9=X[M2] X[M1]
3258 LET N8=N2-N1
3260 LET U1=0
3262 LET U2=9999
3300 REM    DRAW FRAME
3305 PRINT "PLTL"
3310 PRINT INT(U1+1);INT(U1+1)
3330 PRINT INT(U1+1);INT(U2)
3340 PRINT INT(U2);INT(U2)
3350 PRINT INT(U2);INT(U1)
3360 PRINT INT(U1);INT(U1)
3370 PRINT "PLTP"
3400 REM     PLOT POINTS AND ERROR BARS
3405 FOR I=1 TO N
3410 LET U=(X[I]-N5)*U2/N7
3415 LET U3=SQR(Q[I])*U2/N7
3420 LET Y=Y[I]-N1
3425 LET Y1=S[I]-N1
3430 LET Y2=Z[I]-N1
3465 LET Y=Y*U2/N8
3470 LET Y1=Y1*U2/N8
3475 LET Y2=Y2*U2/N8
3480 IF U3#0 THEN 3488
3481 REM   PLOT SQUARE IF XERR=0
3484 GOSUB 3850
3486 GOTO 3500
3487 REM   PLOT BAR WITH WIDTH = XERR(ST.DEV)
3488 PRINT INT(U-U3);INT(Y)
3490 PRINT "PLTL"
3495 PRINT INT(U+U3);INT(Y)
3500 PRINT "PLTP"
3502 REM   PLOT BAR WITH HEIGHT = YERR(ST.DEV)
3505 PRINT INT(U);INT(Y1)
3510 PRINT "PLTL"
3515 PRINT INT(U);INT(Y2)
3520 PRINT "PLTP"
```

```
3545  NEXT I
3550  REM     PLOT FITTED FUNCTION
3551  IF B4=1 THEN 3699
3552  LET N3=40
3553  REM   N3 IS NUMBER OF STEPS
3555  FOR I=1 TO N3
3560  LET X[I]=X[M1]+((I-1)*N9/(N3-1))
3561  LET Q[I]=R[I]=1
3562  LET Y[I]=1
3565  NEXT I
3566  LET N=N3
3568  GOSUB 1000
3571  PRINT "PLTL"
3572  FOR I=1 TO N3
3573  LET T[I]=SQR(S1*T[I])
3574  LET A=F[I]
3575  IF E5=1 THEN 3578
3576  LET A=LOG(A)
3578  LET Y=A-N1
3580  LET U=(X[I]-N5)*U2/N7
3582  LET Y=Y*U2/N8
3584  PRINT INT(U);INT(Y)
3585  NEXT I
3599  PRINT "PLTT"
3600  REM      PLOT ERROR CURVES
3601  IF B4#3 THEN 3699
3605  PRINT
3606  PRINT "CHANGE PEN COLOUR.  TYPE ANY NUMBER TO START"
3607  INPUT A
3608  PRINT "PLTL"
3610  FOR I=1 TO N3
3611  LET Y=F[I]+T[I]
3613  IF E5=1 THEN 3616
3614  LET Y=LOG(Y)
3616  LET Y=Y-N1
3617  LET U=(X[I]-N5)*U2/N7
3626  LET Y=Y*U2/N8
3628  PRINT INT(U);INT(Y)
3630  NEXT I
3635  PRINT "PLTT"
3636  PRINT "PLTL"
3638  FOR I=1 TO N3
3639  LET Y=F[I]-T[I]
3640  IF E5=1 THEN 3642
3641  LET Y=LOG(Y)
3642  LET Y=Y-N1
3644  LET U=(X[I]-N5)*U2/N7
3650  LET Y=Y*U2/N8
3652  PRINT INT(U);INT(Y)
3655  NEXT I
3660  PRINT "PLTT"
3680  PRINT "CHANGE PEN COLOUR. TYPE ANY NUMBER TO START"
3682  INPUT A
3699  RETURN
```

```
3850 REM    PLOT SQUARE WITH SIDE M3
3860 LET M3=50
3865 PRINT "PLTL"
3870 PRINT INT(U-M3);INT(Y+M3)
3875 PRINT INT(U+M3);INT(Y+M3)
3880 PRINT INT(U+M3);INT(Y-M3)
3885 PRINT INT(U-M3);INT(Y-M3)
3890 PRINT INT(U-M3);INT(Y+M3)
3895 RETURN
3900 REM
3901 PRINT "PRINT FITTED DATA    ?    NO=0   YES=1   ";
3902 INPUT B3
3903 IF B3<0 OR B3>1 THEN 3902
3905 PRINT "DO YOU WANT A GRAPH ?  NO=0  LINEAR=1  LOG=2";
3910 INPUT E5
3911 IF E5<0 OR E5>2 THEN 3910
3915 IF E5=0 THEN 3999
3920 PRINT "POINTS ONLY(1), FITTED FUNCTION(2), ERROR CURVES(3)";
3925 INPUT B4
3926 IF B4<1 OR B4>3 THEN 3925
3999 RETURN
5000 REM   SUBROUTINE FOR READING DATA FROM TAPE OR DATA INSTRUCTIONS
5001 REM L3=2   DATA FROM PUNCHED TAPE
5002 IF L9>0 THEN 5255
5010 PRINT "NUMBER OF DATA POINTS";
5011 INPUT N1
5012 IF N1>1 THEN 5014
5013 GOTO 5010
5014 PRINT
5015 PRINT "FIRST X VALUE      ";
5017 INPUT X[1]
5020 PRINT "X STEP             ";
5022 INPUT L4
5025 PRINT "ERROR IN X (XVAR) ";
5027 INPUT L5
5030 PRINT "DATA POINTS TO BE COMBINED IN GROUPS OF :";
5031 INPUT L1
5032 IF L1>0 THEN 5035
5033 PRINT "MINIMIM GROUP SIZE = 1"
5034 GOTO 5030
5035 LET L8=INT(N1/L1)
5038 IF L8<150 THEN 5045
5040 PRINT "MAX. NO. OF GROUPED POINTS IS 150.  INCREASE GROUP SIZE"
5042 GOTO 5030
5045 IF L3=2 THEN 5050
5046 READ A
5047 GOTO 5052
5050 CALL (1,A)
5051 REM   FIRST NUMBER ON TAPE IS CHANNEL ZERO
5052 PRINT
5055 PRINT "CHANNEL ZERO = ",A
5060 FOR I=1 TO L8
5063 LET Y[I]=0
5065 FOR J=1 TO L1
```

```
5067 IF L3=2 THEN 5073
5070 READ A
5071 GOTO 5075
5073 CALL (1,A)
5075 LET Y[I]=Y[I]+A
5080 NEXT J
5081 NEXT I
5083 IF L3#3 THEN 5090
5084 LET N2=N1-L1*L8
5085 IF N2<1 THEN 5090
5086 FOR I=1 TO N2
5087 READ A
5088 NEXT I
5090 LET X[1]=X[1]+(L1-1)*L4/2
5100 FOR I=2 TO L8
5110 LET X[I]=X[I-1]+L1*L4
5115 LET Q[I]=L5
5120 NEXT I
5125 LET Q[1]=L5
5200 PRINT
5205 PRINT "NUMBER OF POINTS TO BE REJECTED ";
5210 INPUT L6
5212 IF L6<0 OR L6>L8-2 THEN 5210
5220 IF L6#0 THEN 5240
5230 LET L9=0
5235 GOTO 5255
5240 PRINT "NUMBER OF REJECTION STEPS  (>0) ";
5245 INPUT L9
5246 IF L9>0 THEN 5255
5247 GOTO 5240
5255 LET L8=L8-L6
5256 LET N=L8
5260 FOR I=1 TO L8
5265 LET J=I+L6
5270 LET X[I]=X[J]
5275 LET Q[I]=Q[J]
5280 LET Y[I]=Y[J]
5290 LET R[I]=Y[I]
5293 LET F[I]=T[I]=0
5295 NEXT I
5300 LET L9=L9-1
5310 PRINT "NUMBER 0F POINTS=",L8
5400 RETURN
6000 REM   DATA STORAGE FOR INPUT ROUTINE 5000 (L3=3)
9000 END
```