Dr. 2102

ORNL/TM-6545

LSFODF: A Generalized Nonlinear Least-Squares Fitting Program for Use with ORELA ODF Files

D. Mark Wharton
David K. Olsen

MASTER

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION · FOR THE DEPARTMENT OF ENERGY

ORNL/TM-6545
Dist. Category UC-79d

Contract No. W-7405-eng-26

Engineering Physics Division

LSFODF: A GENERALIZED NONLINEAR LEAST-SQUARES FITTING PROGRAM FOR USE WITH ORELA ODF FILES

D. Mark Wharton* and David K. Olsen

*ORAU Summer Student from the Physics Department of the University of North Carolina at Asheville. Permanent Address: Route 3, Box 1217B, Asheville, North Carolina 28806.

Date Published - DECEMBER 1978

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

TABLE OF CONTENTS

ABSTRACT

The Fortran-10 program LSFODF has been written on the ORELA PDP-10 in order to perform non-linear least-squares curve fitting with user supplied functions and derivatives on data which can be read directly from ORELA-data-format (ODF) files. LSFODF can be used with any user supplied function and derivatives; has its storage requirements specified in this function; has P-search and η-search capabilities; and can output the input data and fitted curve in an ODF file which then can be manipulated and plotted with the existing ORELA library of ODF programs. A description of the fitting formalism, input instructions, five test cases, and a program listing are given.

I. INTRODUCTION

The need for a generalized nonlinear least-squares curve fitting program which uses as input and output ORELA Data Formated (ODF) files for use on the ORELA PDP-10 has been apparent. LSFODF is an attempt to fulfill this need and will perform, using the so-called method of Guass, linear and nonlinear curve fitting with user supplied functions. For the nonlinear case convergence is aided by various P-search and η-search options which follow closely those used in the computer code MULTI.[1]

To operate LSFODF the user must supply a Fortran-10 FUNCTION FCTN(X,Z) which must return the values of the function and its derivatives (from either analytic or numerical methods) with respect to the variable parameters, all evaluated at X, Z and the current variable parameters. In principle, any function with its derivatives which can be Fortran-10 coded can be used to least-squares fit input data. A user may collect a library of FCTN(X,Z) to be used with LSFODF. The storage requirements of the input data and fitted curve are specified in FCTN(X,Z).

LSFODF is limited to a fitting function of two independent variables which have no uncertainties. The dependent variable is calculated from parameters in FCTN(X,Z) and least-squares fitted by varying these parameter to measured dependent variables whose uncertainties must be uncorrelated. The data to be fitted can be input from either an ASCII or ODF file. The output consist of an ASCII printable file which, if desired, list the covariance matrix of the variable parameters. Also, if desired, an ODF file of the input data and fitted curve may be produced which can be easily plotted and manipulated with the existing code OPRODF.[2]

BLANK PAGE

The next section of this technical memo gives the formalism of the fitting procedure. Section III describes the input and output and the requirements for the FUNCTION FCTN(X,Z). Section IV gives complete descriptions of five test cases or examples. Finally, the Appendix gives a complete listing of LSFODF.

II. FORMALISM AND FITTING PROCEDURE

A. Method of Gauss

LSFODF employs the well-known and commonly used so-called method of Gauss which is reviewed below and contains P-search and η-search options. Assume that we have data in the form of n observations (y_i, x_i z_i) where i = 1, 2, ..., n and that these data can be characterized by the function

$$y = f(x,z;p_1,p_2,\ldots,p_m) \quad , \tag{1}$$

where y is the dependent variable, x and z are independent variables and $p_1,p_2,\ldots p_m$ are parameters which describe the function and form the vector $\vec{p}$. We wish to minimize the quantity ϕ defined by

$$\phi = \sum_{i=1}^{n} w_i[y_i - f(x_i,z_i;\vec{p})]^2 \quad , \tag{2}$$

with respect to the vector $\vec{p}$ where the w_i are weights associated with each y_i. Under normal operation LSFODF follows standard practice and sets the w_i equal to the inverse of the variance of y_i; that is, $w_i = 1/\Delta y_i^{\,2}$ where Δy_i is the uncorrelated standard deviation of y_i.

To minimize ϕ the function f is expanded in a first order Taylor series around $\vec{p}'$ the initial guess for $\vec{p}$:

$$f(x,z;\vec{p}) = f(x,z;\vec{p}') + \sum_{k=1}^{m}[\partial f/\partial p_k](p_k - p'_k) \tag{3}$$

Substituting equation (3) into equation (2) one obtains

$$\phi = \sum_{i=1}^{n} w_i[y_i - f(x_i,z_i;\vec{p}') - \sum_{k=1}^{m}(\partial f_i/\partial p_k)d_k]^2 \quad , \tag{4}$$

where $d_k = (p_k - p'_k)/\eta$ with $\eta = 1$. The symbol $\partial f_i/\partial p_k$ implies the partial derivative of f with respect to p_k evaluated at $x_i, z_i; \vec{p}'$. Setting $\vec{\nabla}\phi = 0$ and rearranging terms one obtains the following m simultaneous linear equations

$$\sum_{i=1}^{n} w_i[y_i - f(x_i,y_i;\vec{p}')](\partial f_i/\partial p_k) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_i(\partial f_i/\partial p_j)(\partial f_i/\partial p_k)d_k \quad (5)$$

which can be written in matrix notation as

$$\vec{V} = M\vec{d} \quad (6a)$$

where

$$V_k = \sum_{i=1}^{n} w_i[y_i - f(x_i,y_i;\vec{p}')](\partial f_i/\partial p_k) \quad (6b)$$

$$M_{jk} = \sum_{i=1}^{n} w_i(\partial f_i/\partial p_j)(\partial f_i/\partial p_k) \quad (6c)$$

Inverting[3] the matrix M the previously defined vector $\vec{d}$ is determined,

$$\vec{d} = M^{-1}\vec{V} \quad , \quad (6d)$$

from which a new parameter vector is obtained

$$\vec{p} = \vec{p}' + \eta d \quad . \quad (6e)$$

For all functions linear in $\vec{p}$, equation (3) is the exact Taylor expansion of f and consequently equation (6) gives the exact solution with $\eta = 1$. For the non-linear problem LSFODF starts with the initial guess $\vec{p}'$ input by the user and iterates through equations (6) replacing $\vec{p}'$ by $\vec{p}$ after each iteration until the convergence criterion

$$\phi_{new} \leq \text{RATIO} \cdot \phi_{old} \quad , \quad (7)$$

is not met, where ϕ_{old} and ϕ_{new} are computed from equation (2) using $\vec{p}'$ and $\vec{p}$

for the old and new iterations, respectively. RATIO can be input; if RATIO is input as zero, a default value of 0.999 is used. Usually for the non-linear case the correction vector is overestimated; consequently η can be input; if η is input as zero, a default value of 0.75 is used. To aid the convergence of non-linear cases LSFODF has various P-search and η-search options which follow closely those contained in the computer code MULTI described in reference 1.

B. P-Search

For non-linear problems the first-order Taylor expansion of equation (3) is not exact and does not guarantee convergence. The P-search aids convergence for some cases by simultaneously decreasing the magnitude and changing the direction of the correction vector $\vec{d}$. This is accomplished by adding a constant term to all diagonal elements of the matrix M; if the added constant is small it has little effect on $\vec{d}$, and if the added constant is large it dominates $\vec{d}$. Therefore, a range of constants, P_ℓ are added using the maximum diagonal element of M, g_{Max}, as a reference magnitude. Specifically P_ℓ is increased following a geometric progression from P_1 to P_{np} in the equation

$$(M + P_\ell I)\vec{d} = \vec{V} \tag{8}$$

where

$$P_{\ell+1} = rP_\ell \tag{8a}$$

$$r = (C_{Max}/C_{Min})^{1/(np-1)} , \tag{8b}$$

$$P_1 = C_{Min} \cdot g_{Max} , \tag{8c}$$

$$P_{np} = C_{Max} \cdot g_{Max} . \tag{8d}$$

Equations (8) are solved for $\vec{d}$ which is inserted into equation (6e) to compute $\vec{p}$, which in turn is used to compute ϕ. The P-search parameters C_{Min}, C_{Max}, and np can be input; if they are input as zero, the default values 0.001, 1.000 and 5 are used, respectively. The switch IP2 gives two P-search options. If IP2 = 0, LSFODF computes ϕ for each of the np values of P_ℓ and continues with the P_ℓ which produced the smallest ϕ. If IP2 = 1, LSFODF computes ϕ sequentially, starting with P_1 and continues when the convergence criterion of equation (7) is met. The switch IPS gives the option to use the P-search only when condition (7) is not met, for each iteration, or not at all. If the P-search fails to produce convergence LSFODF will either stop or proceed with an η-search.

C. η-Search

LSFODF also has the option of keeping the direction of $\vec{d}$ fixed and searching on its magnitude to minimize ϕ by varying η. Three values of η, defined to be η_1, η_2 and η_3 and the corresponding $\phi(\eta_1)$, $\phi(\eta_2)$ and $\phi(\eta_3)$, are required. Initially η_1, η_2, and η_3 are set to zero, the input η and twice the input η, respectively. Without the η-search $\phi(\eta_1)$ and $\phi(\eta_2)$ have been computed. With the η-search $\phi(\eta_3)$ is computed and stored and a parabolic fit of ϕ vs η allows the vertex of the parabola, η_v, to be determined. If either the absolute difference between η_v and η_2 is less than 0.1 η_v, or if the absolute difference between $\phi(\eta_v)$ and $\phi(\eta_2)$ is less than (1 - RATIO) $\phi(\eta_v)$ the search terminates. If neither of these conditions is met, the η-search continues by setting η_1 to the maximum previous η which is less than η_v, η_2 to η_v and η_3 to $2\eta_v - \eta_1$. The new $\phi(\eta_1)$ and $\phi(\eta_2)$ exist, $\phi(\eta_3)$ is calculated, and a new parabolic fit of ϕ vs η allows a new η_v to be computed. This procedure continues in an iterative manner until either η_v or $\phi(\eta_v)$ changes by less than the required amount. The switch IES gives the option to use this method only when condition (7) is not met, for each iteration, or not at all.

D. Parameter Errors

For each iteration, defined to be the calculation of a new matrix M, LSFODF prints $\vec{p}'$ and $\vec{p}$ which are the old and new least-squares estimates of the function parameters. From the inverted matrix, M^{-1}, an estimate of the standard deviation of p'_k is printed and given by

$$\sigma(p_k) = \sqrt{M^{-1}_{kk}\ \chi^2} \tag{9}$$

where $\chi^2 = \phi/(n - m)$, n being the number of data points and m being the number of parameters. The covariance between p_j and p_k is given by

$$\mathrm{Cov}(p_j,p_k) = M^{-1}_{jk}\ \chi^2 \tag{10}$$

From which the correlation matrix element is

$$\mathrm{Cov}(p_j,p_k) = \frac{M^{-1}_{jk}\ \chi^2}{\sigma(p_k)\sigma(p_j)} \tag{11}$$

The switch IF3 gives the option to have the correlation matrix printed for the last iteration, for each iteration, or not at all.

III. PROGRAM OPERATION

LSFODF requires three additional PDP-10 files for execution: a file containing FUNCTION FCTN(X,Z) which gives the user supplied function and derivatives which must be written in FORTRAN 10; a file named LSFODF.INP which contains control input and the initial guesses for the variable parameters; and a file containing the input data to be fitted which can be either a data file in ODF format with any name or a specially prepared ASCII data file named LSFODF.DAT.

If the file containing FUNCTION FCTN(X,Z) is named FCTN.F4, LSFODF is executed by EX FCTN, LSFODF where FCTN must precede LSFODF since the storage requirements for the input data are given in FUNCTION FCTN(X,Z). LSFODF produces the output file LSFODF.PRT and if desired, by setting the IF2 switch, ODF file(s) which can be plotted with either of the existing programs OPRODF[2] or PLTODF.

A. FUNCTION FCTN(X,Z)

The user must supply a FORTRAN 10 file containing the FUNCTION FCTN(X,Z) which contains the fitting function and its derivatives with respect to the variable parameters. This function should be of the form:

```
FUNCTION FCTN(X,Z)
COMMON/DAT/F5*N)
COMMON/FCN/N,M,INTIAL,P(18)
COMMON/DER/DFDP(18)
IF(INTIAL.EQ.1) Go to 999
.
.
.
INTIAL = 1
```

```
999  DFDP(1)   = ∂f₁[X,Z;P(1),P(2),...,P(M)]/∂P(1)
     DFDP(2)   = ∂f₂[X,Z;P(1),P(2),...,P(M)]/∂P(2)
     .
     .
     .
     DFDP(M)   = ∂f[X,Z;P(1),P(2),...,P(M)/∂P(M)
     FCTN      = f(X,Z;P(1),P(2).,,,.P(M)]
     RETURN
     END
```

In FCTN(X,Z), FCTN is returned as the dependent variable corresponding to the data points y_i with errors Δy_i, x is the independent variable (time, neutron energy, etc.) and Z is a second independent variable (sample thickness, gamma-ray energy, etc.) which may or may not be used. The array F of labeled common DAT contains the input data and fitted curve and must be dimensioned greater than five times the number of data points. The variables M and N of labeled common FCN are the number of variables and the number of data points respectively which correspond to the m and n of Section II.

In LSFODF INTIAL is set to zero and can be used to initialize as shown. Data stored in the array F can be accessed as follows: F(ID) = x coordinate, F(ID+1) = measured y coordinate, F(ID+2) = standard deviation of measured y coordinate, F(ID+3) = calculated y coordinate, and F(ID+4) = z coordinate if used, where ID runs from ID = 1 to ID = N*5 in steps of 5. In many cases INTIAL is not needed and the statements containing INTIAL are not needed.

For each iteration and each value X (or X and Z), LSFODF calls FUNCTION FCTN(X,Z). The value of the function evaluated at X and Z with the parameters in the array P(18) of labeled common FCN must be returned as FCTN as shown. Likewise the derivative of the function with respect to the variable parameters P(1), P(2),...,P(M) must be returned in the array DFDP(18) of labeled common DER as shown. The derivatives are evaluated at X and Y with the parameters contained in P(18).

B. FILE LSFODF.INP

CARD #1	FORMAT(16A5) TITLE
CARD #2	FORMAT(2I3,I6,7I2,I3,4F10.6) M,IT2,N,IFDM,IF1, IF2,IF3,IPS,IP2,IES,NPSM,RATIO,ETA,CMIN,CMAX
OPTIONAL CARD #2A	FORMAT(A10,2I6,4I3) FILE,ISC,IEC,IDS(1),IDS(2), IDS(3),IDS(4)
CARDS #3 → #3+M	FORMAT(E12.6) Guesses for initial parameters, one per card.

TITLE	Any alphanumeric characters written on this card will appear on the top of LSFODF.PRT.
M	Number of variable parameters.
IT2	Maximum number of iterations whose default value is 10, that is if columns 4 to 6 are left blank, LSFODF will set IT2 =10
N	Number of data points. If N>0 LSFODF assumes the input data to be fitted will be read from the file LSFODF.DAT. In particular, LSFODF.DAT will contain N data points, one per card with FORMAT(4E12.6). On each card will be in order x, y, Δy and z (optional). For this case LSFODF.INP LSFODF.INP cannot contain CARD #2A. If N=0 LSFODF assumes the input data to be fitted will be read directly from an existing ODF file which must be specified on card #2A.
IFDM	= 0 or 1, only the independent variable x will be used. = 2, both independent variables x and y will be used.
IF1	= -1, input data and curve will be printed for the zeroth and last iteration. = 0, print above for last iteration only. = 1, print above for each iteration. = 2, never print above data.

IF2	= -1, create ODF file for plots of the zeroth and last iterations. = 0, create plot file for the last iteration. = 1, create plot files for each iteration. = 2, create no plot files.
IF3	= 0, print correlation matrix for last iteration only. = 1, print matrix for each iteration. = 2, never print matrix.
IPS	= -1, no P-search. = 0, P-search only when χ^2 does not decrease. = 1, P-search each iteration.
IP2	= 0, LSFODF calculates a table of χ^2 for each value of P_χ specified and continues with the value of P_χ which gave the lowest χ^2. = 1, LSFODF continues with the first value of P_χ which causes χ^2 to decrease.
IES	= -1, No η-search. = 0, η-search only when χ^2 does not decrease. = 1, η-search every iteration.
NPSM	= Number of P-searches per iteration. The np of eq. (8b). (Default value is 5.)
RATIO	Convergences criterion of eq. (7). (Default value is 0.999.)
CMIN	Determines the minimum value of P_χ from eq. (8c) for the P-search. (Default value is 0.001.)
CMAX	Determines the maximum value of P_χ from eq. (8d) for the P-search. (Default value is 1.0.)

OPTIONAL INPUT TO READ DIRECTLY FROM AN ODF FILE.

FILE	Name of ODF file from which data is read.
ISC	Starting channel of data in ODF file.
IEC	Ending channel of data in ODF file.
IDS(1)	Data set to read x from.
IDS(2)	Data set to read y from.
IDS(3)	Data set to read Δy from.
IDS(4)	Data set to read z from if needed, i.e., IFDM = 2.

C. Output

Output files generated by LSFODF are LSFODF.PRT examples of which are shown for the test cases in the following section and ODF plot files as determined by the switch IF2. At both the start (OLD VALUE) and end (NEW VALUE) for each iteration, defined to be the calculation of a new matrix M, LSFODF.PRT has listed the χ^2 which is the ϕ of equation (2) divided by n-m, the values of the parameters, and their corresponding errors of equation (9). In addition, the switch IF1 allows the input data and curve to be printed, and the switch IF3 allows the correlation matrix of equation (11) to be printed.

The ODF plot files are created by the SUBROUTINE ODFGEN and have the name FORXX.DAT where XX is the iteration number which occurred before the file was created. These ODF files contain 5 data sets if two independent variables are used and 4 data sets if one independent variable is used and contain a "number of channels" equal to the number of data points. Data set 1 contains the input x-coordinates, data set 2 contains the corresponding measured y coordinate, data set 3 contains the absolute standard deviation of these measured y coordinates, and data set 4 contains the calculated y coordinate. Data set 5 contains the corresponding z coordinate if used. These files were designed to be plotted with the existing programs OPRODF[2] or PLTODF.

IV. EXAMPLES

Five examples or test cases using LSFODF are presented. The first example is a three parameter fit of 9 exponential time decay and a constant background to the measured ORELA flux transmitted through 10.2 cm of polyethylene.[4] The second example is a fit to the data of example one with two exponential time decays assuming the decay constants are known. In this case the function is linear with respect to the parameters, and with $\eta = 1$, the fit converges in one iteration as expected. The third example is a three-parameter, two-dimensional test case from Ref. 5 for which LSFODF gives identical parameters, errors, and correlations. The fourth example is a three-resonance fit with eleven parameters to the measured transmission of 0.5 to 55.0 eV neutrons through 1.475 inches of ^{238}U.[6] The resulting parameters, errors, and correlation matrix are identical to those obtained from an equivalent least-squares fit with the computer code SIOB[7] to the same data. The fifth example is an eleven parameter fit of a linear background and three unresolved Gaussians integrated over finite channel widths to a measured[8] gamma-ray spectrum. For each example, the file LSFODF.INP, the FUNCTION FCTN(X,Z), the file LSFODF.PRT and a figure of the fitted data with curves from the initial and converged parameters are given.

A. Test Case One

The transmission of the ORELA flux through 10.2 cm of polyethylene was measured at 150 m as a function flight time[4] as shown in Fig. 1 and is fitted with a three parameter function of a time decaying exponential. The fitting function is given by the equation

$$y = p_1 e^{-p_2 x} + p_3$$

where

y = the transmitted flux in arbitrary units ,

x = the flight time in microseconds ,

p_1, p_2, p_3 = parameters to be varied .

The derivatives of y with respect to the variable parameters are:

$$\partial y/\partial p_1 = e^{-p_2 x} \quad ,$$

$$\partial y/\partial p_2 = -x p_1 e^{-p_2 x} \quad ,$$

$$\partial y/\partial p_3 = \frac{\partial y}{\partial p_3} = 1 \quad .$$

The smooth curve in Fig. 1 is the resulting least-squares fit to the measured transmission and the histogram is the calculated function from the initial parameters. The next page gives the input file LSFODF.INP and the FUNCTION FCTN(X,Z) followed by a listing of the output file LSFODF.PRT. The decay constant of 0.03974 gives a half life of 17.4 μsec.

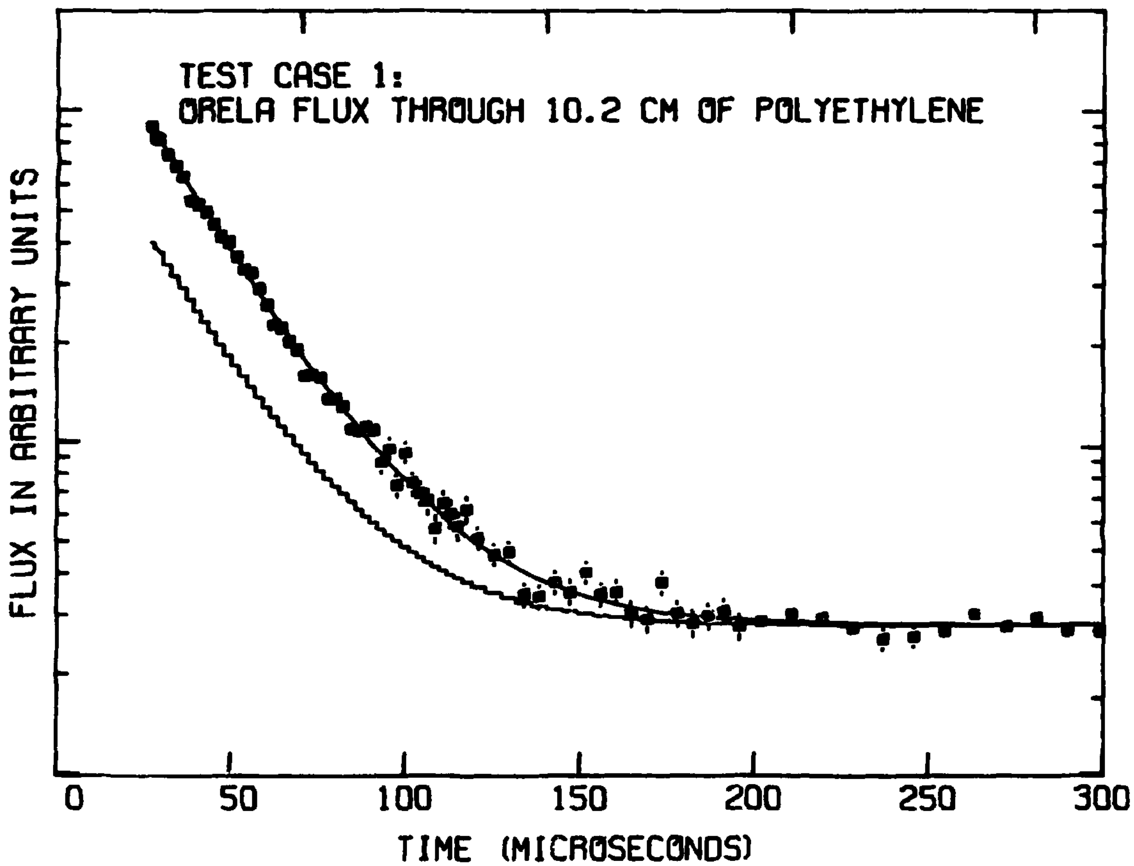

Fig. 1. LSFODF least-squares fit of an exponential decay and a constant background to the measured ORELA flux through 10.2 cm of polyetheylene. The histogram is from the initial-guess parameters and the smooth curve is from the converged parameters.

```
TEST CASE 1: ORELA FLUX THROUGH 10.2 CM CF POLYETHYIENE
  3        77     -1
 .113000E+05
 .404000E-01
 .278000E+03
```

```
      FUNCTION FCTN(X,Z)
      COMMON/DAT/F(500)
      COMMCN/FCN/M,N,INTIAL,P(18)
      COMMON/DER/DFDP(18)
      T=EXP(-P(2)*X)
      DFDP(1)=T
      DFDP(2)=-X*T*P(1)
      DFDP(3)=1.0
      FCTN=P(1)*T+P(3)
      RETURN
      END
```

```
TEST CASE 1: ORELA FLUX THROUGH 10.2 CM OF POLYETHYLENE

CONTROLS =
  3 10    77 0 0-1 0 0 0 0  5  0.999000  0.750000  0.001000  1.000000

NUMBER OF DATA POINTS =  77
NUMBER OF PARAMETERS = 3


ITERATION NUMBER   1


OLD CHI SQUARE = 1.791E+02    NEW CHI SQUARE = 9.634E+00
PARAMETERS
   NUMBER      OLD VALUE        NEW VALUE         ERROR
     1          11300.0          21801.4          1.017E+03
     2          .404000E-01      .392714E-01      2.367E-03
     3          278.000          279.659          1.350E+01


ITERATION NUMBER   2


OLD CHI SQUARE = 9.634E+00    NEW CHI SQUARE = 1.652E+00
PARAMETERS
   NUMBER      OLD VALUE        NEW VALUE         ERROR
     1          21801.4          24417.5          3.947E+02
     2          .392714E-01      .396607E-01      4.678E-04
     3          279.659          279.974          5.631E+00


ITERATION NUMBER   3


OLD CHI SQUARE = 1.652E+00    NEW CHI SQUARE = 1.105E+00
PARAMETERS
   NUMBER      OLD VALUE        NEW VALUE         ERROR
     1          24417.5          25081.1          3.302E+02
     2          .396607E-01      .397201E-01      3.515E-04
     3          279.974          280.064          4.594E+00


ITERATION NUMBER   4


OLD CHI SQUARE = 1.105E+00    NEW CHI SQUARE = 1.070E+00
PARAMETERS
   NUMBER      OLD VALUE        NEW VALUE         ERROR
     1          25081.1          25247.9          3.261E+02
     2          .397201E-01      .397340E-01      3.383E-04
     3          280.064          280.091          4.520E+00


ITERATION NUMBER   5


OLD CHI SQUARE = 1.070E+00    NEW CHI SQUARE = 1.068E+00
PARAMETERS
   NUMBER      OLD VALUE        NEW VALUE         ERROR
     1          25247.9          25289.7          3.260E+02
     2          .397340E-01      .397376E-01      3.361E-04
     3          280.091          280.098          4.515E+00
```

ITERATION NUMBER 6

P SEARCH

NPS	P	CHI SQUARE
0	0.000E-01	1.0681E+00
1	6.628E+04	1.0682E+00
2	3.727E+05	1.0682E+00
3	2.096E+06	1.0682E+00
4	1.179E+07	1.0682E+00
5	6.628E+07	1.0682E+00
0	0.000E-01	1.0681E+00

ETA SEARCH

ETA	CHI SQUARE
0.000E-01	1.0683E+00
7.500E-01	1.0681E+00
1.000E+00	1.0681E+00

OLD CHI SQUARE = 1.068E+00 NEW CHI SQUARE = 1.068E+00

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	25289.7	25303.8	3.260E+02
2	.397376E-01	.397389E-01	3.356E-04
3	280.098	280.101	4.515E+00

CORRELATION MATRIX

(1)	1.00		
(2)	0.92	1.00	
(3)	0.28	0.39	1.00
	(1)	(2)	(3) (

X AXIS	Y EXP	Y ERROR	Y CALC
2.7150E+01	9.0135E+03	6.7132E+01	8.8825E+03
2.8250E+01	8.3350E+03	2.0415E+02	8.5146E+03
2.8450E+01	8.3150E+03	2.0390E+02	8.4494E+03
2.8650E+01	8.3000E+03	2.0372E+02	8.3847E+03
2.8850E+01	8.2500E+03	1.9628E+02	8.3206E+03
2.9050E+01	8.2200E+03	2.0273E+02	8.2569E+03
2.9250E+01	8.1950E+03	2.0242E+02	8.1938E+03
3.1450E+01	7.3950E+03	1.9229E+02	7.5313E+03
3.3650E+01	6.8250E+03	1.8473E+02	6.9243E+03
3.5850E+01	6.3650E+03	1.7840E+02	6.3681E+03
3.8050E+01	5.3950E+03	1.6424E+02	5.8584E+03
4.0250E+01	5.2150E+03	1.6148E+02	5.3914E+03
4.2450E+01	4.9400E+03	1.5716E+02	4.9636E+03
4.4650E+01	4.5400E+03	1.5067E+02	4.5715E+03
4.6850E+01	4.1850E+03	1.4466E+02	4.2123E+03
4.9050E+01	4.0300E+03	1.4195E+02	3.8831E+03
5.1250E+01	3.6400E+03	1.3491E+02	3.5815E+03
5.3450E+01	3.3300E+03	1.2903E+02	3.3051E+03
5.5650E+01	3.2400E+03	1.2728E+02	3.0519E+03
5.7850E+01	2.9100E+03	1.2062E+02	2.8198E+03
6.0050E+01	2.6100E+03	1.1424E+02	2.6072E+03
6.2250E+01	2.2700E+03	1.0654E+02	2.4124E+03
6.4450E+01	2.2150E+03	1.0524E+02	2.2339E+03
6.6650E+01	2.0250E+03	1.0062E+02	2.0704E+03

6.8850E+01	1.9050E+03	9.7596E+01	1.9205E+03
7.1050E+01	1.5900E+03	8.9163E+01	1.7832E+03
7.3250E+01	1.6000E+03	8.9443E+01	1.6573E+03
7.5450E+01	1.5650E+03	8.8459E+01	1.5420E+03
7.7650E+01	1.3600E+03	8.2462E+01	1.4364E+03
7.9850E+01	1.3550E+03	8.2310E+01	1.3396E+03
8.2050E+01	1.2800E+03	8.0000E+01	1.2509E+03
8.4250E+01	1.0950E+03	7.3953E+01	1.1696E+03
8.6450E+01	1.0750E+03	7.3314E+01	1.0952E+03
8.8650E+01	1.1150E+03	7.4666E+01	1.0269E+03
9.0850E+01	1.0850E+03	7.3655E+01	9.6443E+02
9.3050E+01	8.6500E+02	6.5765E+01	9.0714E+02
9.5250E+01	9.5500E+02	6.9101E+01	8.5465E+02
9.7450E+01	7.3500E+02	6.0622E+01	8.0655E+02
9.9650E+01	9.3000E+02	6.8191E+01	7.6248E+02
1.0185E+02	7.4500E+02	6.1033E+01	7.2210E+02
1.0405E+02	6.9500E+02	5.8949E+01	6.8510E+02
1.0625E+02	6.6000E+02	5.7446E+01	6.5119E+02
1.0845E+02	5.4500E+02	5.2202E+01	6.2013E+02
1.1065E+02	6.5000E+02	5.7009E+01	5.9166E+02
1.1285E+02	6.0500E+02	5.5000E+01	5.6558E+02
1.1505E+02	5.5500E+02	5.2678E+01	5.4166E+02
1.1725E+02	6.2500E+02	5.5902E+01	5.1979E+02
1.2075E+02	5.1000E+02	3.5707E+01	4.8866E+02
1.2515E+02	4.5750E+02	3.3819E+01	4.5521E+02
1.2955E+02	4.6500E+02	3.4095E+01	4.2712E+02
1.3395E+02	3.4750E+02	2.9475E+01	4.0353E+02
1.3835E+02	3.4250E+02	2.9262E+01	3.8373E+02
1.4275E+02	3.7750E+02	3.0721E+01	3.6711E+02
1.4715E+02	3.5250E+02	2.9686E+01	3.5315E+02
1.5155E+02	4.0500E+02	3.1820E+01	3.4143E+02
1.5595E+02	3.4750E+02	2.9475E+01	3.3159E+02
1.6035E+02	3.5250E+02	2.9686E+01	3.2333E+02
1.6475E+02	3.0500E+02	2.7613E+01	3.1640E+02
1.6915E+02	2.9250E+02	2.7042E+01	3.1056E+02
1.7355E+02	3.7750E+02	3.0721E+01	3.0569E+02
1.7795E+02	3.0500E+02	2.7613E+01	3.0156E+02
1.8235E+02	2.8500E+02	2.6693E+01	2.9814E+02
1.8675E+02	3.0000E+02	2.7386E+01	2.9524E+02
1.9115E+02	3.1000E+02	2.7839E+01	2.9281E+02
1.9555E+02	2.7750E+02	2.6339E+01	2.9077E+02
2.0175E+02	2.8750E+02	1.8957E+01	2.8844E+02
2.1055E+02	3.0375E+02	1.9486E+01	2.8596E+02
2.1935E+02	2.9625E+02	1.9244E+01	2.8425E+02
2.2815E+02	2.7500E+02	1.8540E+01	2.8302E+02
2.3695E+02	2.5500E+02	1.7854E+01	2.8216E+02
2.4575E+02	2.5875E+02	1.7984E+01	2.8155E+02
2.5455E+02	2.6875E+02	1.8329E+01	2.8112E+02
2.6335E+02	3.0375E+02	1.9486E+01	2.8082E+02
2.7215E+02	2.7750E+02	1.8625E+01	2.8061E+02
2.8095E+02	2.9625E+02	1.9244E+01	2.8046E+02
2.8975E+02	2.6875E+02	1.8329E+01	2.8035E+02
2.9895E+02	2.6750E+02	1.2930E+01	2.8028E+02

B. Test Case Two

Example two shows the operation of LSFODF for a linear function. The data from example one are fitted with two exponentials whose time constants are assumed known. In this case the function can be written as

$$y = p_1 e^{-\lambda_1 x} + p_2 e^{-\lambda_2 x} + p_3 ,$$

where λ_1 = .03735 and λ_2 = .2005 are assumed known. The derivatives of the function with respect to the variable parameters are:

$$\partial y/\partial p_1 = e^{-\lambda_1 x}$$

$$\partial y/\partial p_2 = e^{-\lambda_2 x}$$

$$\partial y/\partial p_3 = 1.0$$

The initial guesses for the variable parameters are all zero. Because equation (3) is exact for this example; that is, the second order derivatives are all zero, with $\eta = 1$, the variable parameters converge in one iteration to their corresponding values obtained in the previous example. The next page gives the input file LSFODF.INP and the FUNCTION FCTN(X,Z) followed by a listing of the output file LSFODF.PRT. The second iteration is not required, but demonstrates convergence in one iteration.

```
TEST CASE 2: LINEAR FUNCTION
  3       81   2 2                        1.
0.0
0.0
0.0
```

```
      FUNCTION FCTN(X,Z)
      COMMON/DAT/F(410)
      COMMON/FCN/M,N,INTIAL,P(18)
      COMMON/DER/DFDP(18)
      IF (INTIAL.EQ.1) GOTO 5
      INTIAL=1
      A1=-3.73483E-2
      A2=-2.00494E-1
      DFDP(3)=1.
    5 DFDP(1)=EXP(A1*X)
      DFDP(2)=EXP(A2*X)
      FCTN=P(1)*DFDP(1)+P(2)*DFDP(2)+P(3)
      RETURN
      END
```

```
TEST CASE 2: LINEAR FUNCTION

CONTROLS =
  3 10    81 0 2 2 0 0 0 0  5  0.999000  1.000000  0.001000  1.000000

NUMBER OF DATA POINTS =  81
NUMBER OF PARAMETERS = 3


ITERATION NUMBER   1


OLD CHI SQUARE = 2.043E+03    NEW CHI SQUARE = 1.945E+00
PARAMETERS
   NUMBER     OLD VALUE      NEW VALUE         ERROR
     1         .000000        21414.9        1.838E+02
     2         .000000        278781.        7.429E+03
     3         .000000        274.176        5.640E+00


ITERATION NUMBER   2


            P  SEARCH
    NPS       P            CHI SQUARE
     0      0.000E-01      1.9449E+00
     1      6.530E-05      1.9449E+00
     2      3.672E-04      1.9449E+00
     3      2.065E-03      1.9449E+00
     4      1.161E-02      1.9449E+00
     5      6.530E-02      1.9449E+00
     0      0.000E-01      1.9449E+00

          ETA SEARCH
        ETA          CHI SQUARE
     0.000E-01       1.9449E+00
     1.000E+00       1.9449E+00
     5.000E-01       1.9449E+00

OLD CHI SQUARE = 1.945E+00    NEW CHI SQUARE = 1.945E+00
PARAMETERS
   NUMBER     OLD VALUE      NEW VALUE         ERROR
     1         21414.9        21414.9        1.838E+02
     2         278781.        278781.        7.429E+03
     3         274.176        274.176        5.640E+00

CORRELATION MATRIX

( 1)     1.00

( 2)    -0.83   1.00

( 3)    -0.23   0.13   1.00

        ( 1)   ( 2)   ( 3)   (
```

C. Test Case Three

Example three is a two dimensional test case taken from Ref. 5 where statistical fluctuations with a standard deviation of 0.1 were generated for the function

$$y = 3x + 3e^{-z/2} \quad .$$

The constants 3, 3 and 2 were then treated as parameters to be obtained by a least-squares fit. Consequently a function

$$y = p_1 x + p_2 e^{-p_3 z}$$

with derivatives with respect to the variable parameters

$$\partial y/\partial p_1 = x$$

$$\partial y/\partial p_2 = e^{p_3 z}$$

$$\partial y/\partial p_3 = -z p_2 e^{-p_3 z}$$

are used. Using $\eta = 1$ and the input data and initial parameter guesses of Ref. 5, LSFODF follows the fit of Ref. 5 and gives identical converged parameters and covariance matrix. The next page gives the input file LSFODF.INP and the FUNCTION FCTN(X,Z) followed by a listing of the output file LSFODF.PRT. Note that the switch IFDM is set equal to 2 for two independent variables.

```
TEST CASE 3: TWO DIMENSIONS
  3       13 2   2                        1.
2.97
1.93
-.41
```

```
      FUNCTION FCTN(X,Z)
      COMMCN/DAT/F(3000)
      COMMON/FCN/M,N,INTIAL,P(18)
      COMMON/DER/DFDP(18)
      DFDP(1)=X
      DFDP(2)=EXP(P(3)*Z)
      DFDP(3)=Z*DFDP(2)*P(2)
      FCTN=P(1)*X+P(2)*DFDP(2)
      RETURN
      END
```

```
TEST CASE 3: TWO DIMENSIONS

CONTROLS =
  3 10    13 1 0 2 0 0 0 0  5  0.999000  1.000000  0.001000  1.000000

NUMBER OF DATA POINTS =  13
NUMBER OF PARAMETERS = 3


ITERATION NUMBER   1


OLD CHI SQUARE = 5.033E-02    NEW CHI SQUARE = 1.608E-02
PARAMETERS
   NUMBER      OLD VALUE        NEW VALUE          ERROR
     1          2.97000          3.01799         3.676E-02
     2          2.93000          2.95693         7.862E-02
     3         -.410000         -.513101         2.477E-02


ITERATION NUMBER   2


OLD CHI SQUARE = 1.608E-02    NEW CHI SQUARE = 1.574E-02
PARAMETERS
   NUMBER      OLD VALUE        NEW VALUE          ERROR
     1          3.01799          3.01738         3.654E-02
     2          2.95693          2.95830         7.809E-02
     3         -.513101         -.522197         2.920E-02


ITERATION NUMBER   3


           P  SEARCH
    NPS       P             CHI SQUARE
     0     0.000E-01        1.5744E-02
     1     3.867E-02        1.5744E-02
     2     2.175E-01        1.5744E-02
     3     1.223E+00        1.5744E-02
     4     6.877E+00        1.5744E-02
     5     3.867E+01        1.5744E-02
     3     1.223E+00        1.5744E-02

         ETA SEARCH
       ETA          CHI SQUARE
     0.000E-01      1.5744E-02
     1.000E+00      1.5744E-02

OLD CHI SQUARE = 1.574E-02    NEW CHI SQUARE = 1.574E-02
PARAMETERS
   NUMBER      OLD VALUE        NEW VALUE          ERROR
     1          3.01738          3.01713         3.655E-02
     2          2.95830          2.95816         7.811E-02
     3         -.522197         -.521958         2.967E-02
```

CORRELATION MATRIX

(1)	1.00		
(2)	-0.45	1.00	
(3)	-0.55	-0.19	1.00
	(1)	(2)	(3) (

X AXIS	Y EXP	Y ERROR	Y CALC	Z AXIS
0.0000E-01	2.9300E+00	1.0000E+00	2.9582E+00	0.0000E-01
0.0000E-01	1.9500E+00	1.0000E+00	1.7552E+00	1.0000E+00
0.0000E-01	8.1000E-01	1.0000E+00	1.0415E+00	2.0000E+00
0.0000E-01	5.8000E-01	1.0000E+00	6.1798E-01	3.0000E+00
1.0000E+00	5.9000E+00	1.0000E+00	5.9753E+00	0.0000E-01
1.0000E+00	4.7400E+00	1.0000E+00	4.7724E+00	1.0000E+00
1.0000E+00	4.1800E+00	1.0000E+00	4.0586E+00	2.0000E+00
1.0000E+00	4.0500E+00	1.0000E+00	4.0586E+00	2.0000E+00
2.0000E+00	9.0300E+00	1.0000E+00	8.9924E+00	0.0000E-01
2.0000E+00	7.8500E+00	1.0000E+00	7.7895E+00	1.0000E+00
2.0000E+00	7.2200E+00	1.0000E+00	7.0757E+00	2.0000E+00
2.5000E+00	8.5000E+00	1.0000E+00	8.5843E+00	2.0000E+00
2.9000E+00	9.8100E+00	1.0000E+00	9.9058E+00	1.8000E+00

D. Test Case Four

Example 4 is a three resonance fit to the measured[6] neutron transmission of 0.5 to 55.0 eV neutrons through a n = 0.1754 atom/barn thick sample of ^{238}U. Assuming no resolution broadening the transmission can be expressed in terms of the total cross section as:

$$T(E) = e^{-n\sigma(E)} \; ,$$

where assuming no Doppler broadening the total cross section $\sigma(E)$ can be expressed as:

$$\sigma(E) = 4\pi R^2 + P\ell n\left[\frac{75.0 - E}{E + 20.0}\right] + \frac{\pi}{k^2} \sum_{i=1}^{3} \Gamma_{ni}[\Gamma_i - 4kRD_i]/B_i$$

where

$$\Gamma_{ni} = \sqrt{E}\, \Gamma^{\circ}_{ni}$$

$$\Gamma \;\; = \Gamma_{ni} + \Gamma_{\gamma i}$$

$$B_i = (E_i - E)^2 + \Gamma_i^{\,2}/4$$

$$k = 0.0021874\sqrt{E}$$

$$D_i = E_i - E \quad .$$

We desire to fit the data shown in Fig. 2, allowing the effective radius R, the strength function parameter P, and the three sets of Γ°_{ni}, $\Gamma_{\gamma i}$ and E_i all to vary. The derivatives of the transmission with respect to the variable parameters are

$$\frac{\partial T}{\partial R} = \left(8\pi R + \frac{4\pi}{k} \sum_{i=1}^{3} \Gamma_{ni} D_i/B_i\right) nT$$

$$\frac{\partial T}{\partial P} = \ln\left[\frac{75.0 - E}{E + 20.0}\right] nT$$

$$\frac{\partial T}{\partial E_i} = \frac{\pi}{k^2}\,\Gamma_{ni}[4Rk(2D_i{}^2 - B_i) - 2D_i\Gamma_i]nT/B_i{}^2$$

$$\frac{\partial T}{\partial \Gamma_{ni}} = \frac{\pi\sqrt{E}}{k^2}[B_i(\Gamma_i + \Gamma_{ni} - 4RkD_i) + \Gamma_{ni}\Gamma_i(2RkD_i - \Gamma_i/2)]nT/B_i$$

$$\frac{\partial T}{\partial \Gamma\gamma_i} = \frac{\pi}{k^2}\,\Gamma_{ni}[B_i + \Gamma_i(2RkD_i - \Gamma_i/2)]nT/B_i$$

The input file LSFODF.INP and the FUNCTION FCTN(X,Z) are given on the next page followed by a listing of the file LSFODF.PRT. The switch IF1 was set to zero so no printouts of the 500 data points and fit occurred. In FUNCTION FCTN(X,Z) and LSFODF.PRT, the variable parameters R,P, E_1,$\Gamma^\circ_{n_1}$, Γ_{γ_1},E_2 $\Gamma^\circ_{n_2}$,Γ_{γ_2},E_3,$\Gamma^\circ_{n_2}$,Γ_{γ_3} are ordered sequentially from one to eleven. The dependent variable y is T and the independent variable x is E. LSFODF gave identical converged parameters and covariance matrix as the least-squares-fit computer code SIOB[7] using the same cross section formula to fit the same input data.

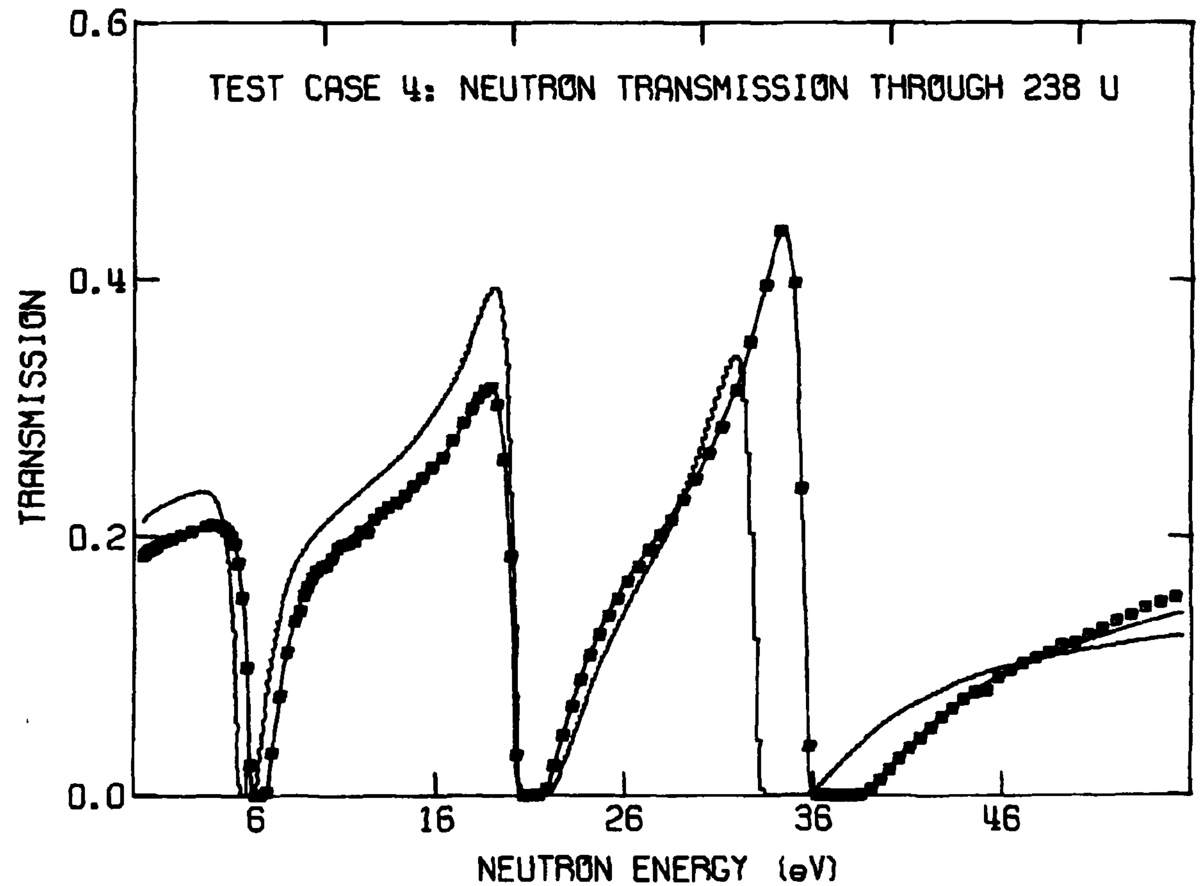

Fig. 2. LSFODF least-squares fit of three resonances with eleven parameters to the measured neutron transmission of 0.5 to 55.0 eV neutrons through a 0.1754 atom/barn of ^{238}U. The calculated transmission from the initial parameter guesses and converged parameters are shown.

```
C  FILE LSPODF.INP FOR TEST CASE 4
TEST CASE 4: NEUTRON TRANSMISSION THROUGH 238 U
 11 25    500    2-1           7                           .00001    .1
3.000000E-01
0.000000E-01
6.000000E+00
5.700000E-04
2.300000E-02
2.100000E+01
2.800000E-03
2.300000E-02
3.400000E+01
5.000000E-03
2.300000E-02
```

```
      FUNCTION FCTN(X,Z)
      COMMON/DAT/F(3000)
      COMMON/FCN/M,N,INTIAL,P(18)
      COMMON/DER/DFDP(18)
      DIMENSION G(3),GN(3),D(3),B(3),C(3)
      SE=SQRT(X)
      FK=2.1874E-3*SE
      RHO=P(1)*FK
      FK=1./FK
      POK2=3.14159*FK
      T=0
      DFDP(1)=0
      DO 10 L=1,3
      GN(L)=SE*P(L*3+1)
      D(L)=P(L*3)-X
      G(L)=GN(L)+P(L*3+2)
      C(L)=(G(L)-4*RHO*D(L))*FK
      B(L)=D(L)*D(L)+G(L)*G(L)/4
      T=T+GN(L)*C(L)/B(L)
      C(L)=POK2*FK/(B(L)*B(L))
   10 DFDP(1)=DFDP(1)-GN(L)*D(L)/B(L)
      T=T*POK2
      DFDP(1)=DFDP(1)*POK2*4+25.13274*P(1)
      DFDP(2)=ALOG((75.-X)/(X+20.))
      T=T+4*POK2*RHO*P(1)+P(2)*DFDP(2)
      POK2=POK2*FK
      T=EXP(-.1754*T)
      FCTN=T
      T=-.1754*T
      DFDP(1)=DFDP(1)*T
      DFDP(2)=DFDP(2)*T
      DO 20 L=1,3
      C(L)=C(L)*T
      DFDP(L*3)=C(L)*GN(L)*(4*RHO*(2*D(L)*D(L)-B(L))-2*D(L)*G(L))
      D(L)=D(L)*RHO
      G1=G(L)*(D(L)+D(L)-G(L)/2)
      DFDP(L*3+1)=C(L)*SE*(B(L)*(G(L)+GN(L)-4*D(L))+GN(L)*G1)
   20 DFDP(L*3+2)=C(L)*GN(L)*(B(L)+G1)
      RETURN
      END
```

```
TEST CASE 4: NEUTRON TRANSMISSION THROUGH 238 U

CONTROLS =
 11 25    500 0 2-1 0 0 0 0  7  0.999000  0.750000  0.000010  0.100000

NUMBER OF DATA POINTS = 500
NUMBER OF PARAMETERS =11


ITERATION NUMBER    1


OLD CHI SQUARE = 3.296E+04    NEW CHI SQUARE = 1.516E+04
PARAMETERS
   NUMBER       OLD VALUE       NEW VALUE          ERROR
     1           .900000         .915491          1.266E-02
     2           .000000         .657188          2.653E-01
     3          6.00000         6.03363           3.640E-02
     4           .570000E-03     .976981E-03      6.642E-05
     5           .230000E-01     .708754E-02      3.543E-03
     6          21.0000         20.9096           8.355E-02
     7           .280000E-02     .247447E-02      2.752E-04
     8           .230000E-01     .234216E-01      4.023E-03
     9          34.0000         34.5847           2.268E-01
    10           .500000E-02     .612594E-02      3.892E-04
    11           .230000E-01     .288834E-01      7.229E-03


ITERATION NUMBER    2


OLD CHI SQUARE = 1.516E+04    NEW CHI SQUARE = 4.977E+03
PARAMETERS
   NUMBER       OLD VALUE       NEW VALUE          ERROR
     1           .915491         .908523          8.206E-03
     2           .657188        1.00460           1.632E-01
     3          6.03363         6.18566           1.752E-02
     4           .976981E-03     .105652E-02      3.431E-05
     5           .708754E-02     .994510E-02      4.674E-04
     6          20.9096         20.8618           5.269E-02
     7           .247447E-02     .225978E-02      1.459E-04
     8           .234216E-01     .236068E-01      2.655E-03
     9          34.5847         35.3839           1.551E-01
    10           .612594E-02     .648994E-02      2.893E-04
    11           .288834E-01     .379271E-01      4.860E-03
```

ITERATION NUMBER 3

OLD CHI SQUARE = 4.977E+03   NEW CHI SQUARE = 1.770E+03

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	.908523	.892322	5.119E-03
2	1.00460	.966786	1.019E-01
3	6.18566	6.39734	1.343E-02
4	.105652E-02	.908573E-03	2.985E-05
5	.994510E-02	.144486E-01	4.354E-04
6	20.8618	20.8771	3.172E-02
7	.225978E-02	.216889E-02	8.168E-05
8	.236068E-01	.255762E-01	1.690E-03
9	35.3839	36.2078	1.152E-01
10	.648994E-02	.617251E-02	2.123E-04
11	.379271E-01	.420932E-01	3.809E-03

ITERATION NUMBER 4

OLD CHI SQUARE = 1.770E+03   NEW CHI SQUARE = 3.028E+02

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	.892322	.888179	2.101E-03
2	.966786	.769941	4.177E-02
3	6.39734	6.56822	6.232E-03
4	.908573E-03	.687370E-03	1.456E-05
5	.144486E-01	.191425E-01	3.021E-04
6	20.8771	20.8852	1.320E-02
7	.216889E-02	.208678E-02	3.351E-05
8	.255762E-01	.268813E-01	7.729E-04
9	36.2078	36.8256	5.350E-02
10	.617251E-02	.559707E-02	9.270E-05
11	.420932E-01	.338657E-01	1.908E-03

ITERATION NUMBER 5

OLD CHI SQUARE = 3.028E+02   NEW CHI SQUARE = 4.040E+01

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	.888179	.893022	7.040E-04
2	.769941	.663771	1.469E-02
3	6.56822	6.64469	2.236E-03
4	.687370E-03	.590952E-03	5.149E-06
5	.191425E-01	.222269E-01	1.763E-04
6	20.8852	20.8901	4.798E-03
7	.208678E-02	.206737E-02	1.208E-05
8	.268813E-01	.276800E-01	2.996E-04
9	36.8256	36.7152	1.060E-02
10	.559707E-02	.563964E-02	3.033E-05
11	.338657E-01	.226521E-01	5.855E-04

ITERATION NUMBER 6

OLD CHI SQUARE = 4.040E+01   NEW CHI SQUARE = 2.718E+01

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	.893022	.893908	5.570E-04
2	.663771	.645932	1.187E-02
3	6.64469	6.66289	1.745E-03
4	.590952E-03	.571380E-03	4.055E-06
5	.222269E-01	.233053E-01	1.859E-04
6	20.8901	20.8910	3.954E-03
7	.206737E-02	.206270E-02	9.846E-06
8	.276800E-01	.278520E-01	2.515E-04
9	36.7152	36.6947	7.028E-03
10	.563964E-02	.565334E-02	2.334E-05
11	.226521E-01	.217190E-01	3.258E-04

ITERATION NUMBER 7

OLD CHI SQUARE = 2.718E+01   NEW CHI SQUARE = 2.650E+01

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	.893908	.894147	5.505E-04
2	.645932	.642348	1.174E-02
3	6.66289	6.66683	1.717E-03
4	.571380E-03	.567306E-03	3.998E-06
5	.233053E-01	.235700E-01	1.982E-04
6	20.8910	20.8911	3.913E-03
7	.206270E-02	.206190E-02	9.720E-06
8	.278520E-01	.278841E-01	2.499E-04
9	36.6947	36.6899	6.946E-03
10	.565334E-02	.565747E-02	2.296E-05
11	.217190E-01	.214928E-01	3.120E-04

ITERATION NUMBER 8

OLD CHI SQUARE = 2.650E+01   NEW CHI SQUARE = 2.646E+01

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	.894147	.894211	5.502E-04
2	.642348	.641618	1.174E-02
3	6.66683	6.66769	1.717E-03
4	.567306E-03	.566443E-03	3.997E-06
5	.235700E-01	.236277E-01	2.017E-04
6	20.8911	20.8911	3.912E-03
7	.206190E-02	.206177E-02	9.714E-06
8	.278841E-01	.278902E-01	2.500E-04
9	36.6899	36.6888	6.942E-03
10	.565747E-02	.565869E-02	2.292E-05
11	.214928E-01	.214357E-01	3.095E-04

ITERATION NUMBER 9

P SEARCH

NPS	P	CHI SQUARE
0	0.000E-01	2.6462E+01
1	6.405E+07	2.6463E+01
2	2.973E+08	2.6464E+01
3	1.380E+09	2.6464E+01
4	6.405E+09	2.6464E+01
5	2.973E+10	2.6464E+01
6	1.380E+11	2.6464E+01
7	6.405E+11	2.6464E+01
0	0.000E-01	2.6462E+01

ETA SEARCH

ETA	CHI SQUARE
0.000E-01	2.6464E+01
7.500E-01	2.6462E+01
9.769E-01	2.6462E+01

OLD CHI SQUARE = 2.646E+01   NEW CHI SQUARE = 2.646E+01

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	.894211	.894234	5.503E-04
2	.641618	.641424	1.174E-02
3	6.66769	6.66793	1.717E-03
4	.566443E-03	.566205E-03	3.998E-06
5	.236277E-01	.236434E-01	2.025E-04
6	20.8911	20.8911	3.913E-03
7	.206177E-02	.206174E-02	9.714E-06
8	.278902E-01	.278916E-01	2.501E-04
9	36.6888	36.6895	6.941E-03
10	.565869E-02	.565914E-02	2.292E-05
11	.214357E-01	.214166E-01	3.088E-04

CORRELATION MATRIX

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
(1)	1.00										
(2)	-0.79	1.00									
(3)	-0.01	-0.25	1.00								
(4)	-0.19	0.50	-0.56	1.00							
(5)	0.08	-0.45	0.73	-0.84	1.00						
(6)	-0.28	0.15	0.06	-0.05	0.07	1.00					
(7)	-0.05	0.33	-0.09	0.21	-0.15	-0.12	1.00				
(8)	-0.27	0.06	0.10	-0.12	0.13	0.76	-0.36	1.00			
(9)	-0.40	0.37	-0.02	0.11	-0.07	0.13	-0.04	0.15	1.00		
(10)	-0.06	0.34	-0.08	0.17	-0.13	0.16	0.11	0.18	0.38	1.00	
(11)	-0.44	0.29	0.01	0.06	-0.03	0.09	-0.11	0.11	0.78	0.05	1.00

E. Test Case Five

Example five is a shape fit to three unresolved gamma-ray groups in a Li(Ge) spectrum obtained from an ORELA $^{238}U(n,n'\gamma)$ measurement.[8] The data to be fitted is shown in Fig. 3 where the total number of counts for each gamma-ray group is desired. We assume the γ-ray groups to have Gaussian line shapes and, because the FWHM of the Gaussians are only about two channels wide, the counts for each channel must be taken as an integral over the line shape. These integrations can be accomplished with the error functions ERF.[9] In particular, the number of counts y in a channel, spanning energies from $E = E^-$ to $E = E^+$, produced from a gamma-ray group with total area = A, centroid energy = ε, and FWHM = W is

$$y = A[\mathrm{ERF}(\S^+) - \mathrm{ERF}(\S^-)]/2 \ ,$$

where

$$\S = 2\sqrt{\ell n 2}\ (E - \varepsilon)/W \ .$$

Consequently, assuming a linear background of the form ME + B, we chose a fitting function

$$y = ME + B + 1/2 \sum_{i=1}^{3} A_i[\mathrm{ERF}(\S_i^+) - \mathrm{ERF}(\S_i^-)]$$

and allow the parameters M, E, and the three sets of A_i, ε_i and W_i to vary. The derivatives of y with respect to the variable parameters are

$$\partial y/\partial M = E$$

$$\partial y/\partial B = 1$$

$$\partial y/\partial A_i = [\mathrm{ERF}(\S_i^+) - \mathrm{ERF}(\S_i^-)]/2$$

$$\partial y/\partial \varepsilon_i = 2\sqrt{\ell n2}\; A_i[\mathrm{EXP}^2(-\S_i^+) - \mathrm{EXP}^2(-\S_i^-)]/\sqrt{\pi}\, W_i$$

$$\partial y/\partial W_i = -A_i[\S_i^- \;\mathrm{EXP}(-\S_i^-) - \S_i^+ \;\mathrm{EXP}(-\S_i^+)]/\sqrt{\pi}\, W_i \;.$$

The next page gives the input file LSFODF.INP and the FUNCTION FCTN(X,Z) followed by a listing of the output file LSFODF.PRT. Figure 3 shows the beginning and final fit to the measured spectrum. In this example N = 0 and the data is read directly as channels 927 to 952 from sections 1, 28 and 28 of the ODF file UCSPEC.ODF. Note that the spectrum is also inputed into the error locations of the array F and that INTIAL in FUNCTION FCTN(X,Z) is used to take the square root of the contents of these locations so the spectrum has the correct errors prior to the least squares fit. Also note the use of INTIAL to calculate an initial table of $\S^+$ and $\S^-$. The parameters are ordered sequentially from one to eleven as ε_1, W_1, A_1, ε_2, W_2, A_2, ε_3, W_3, A_3, M and B.

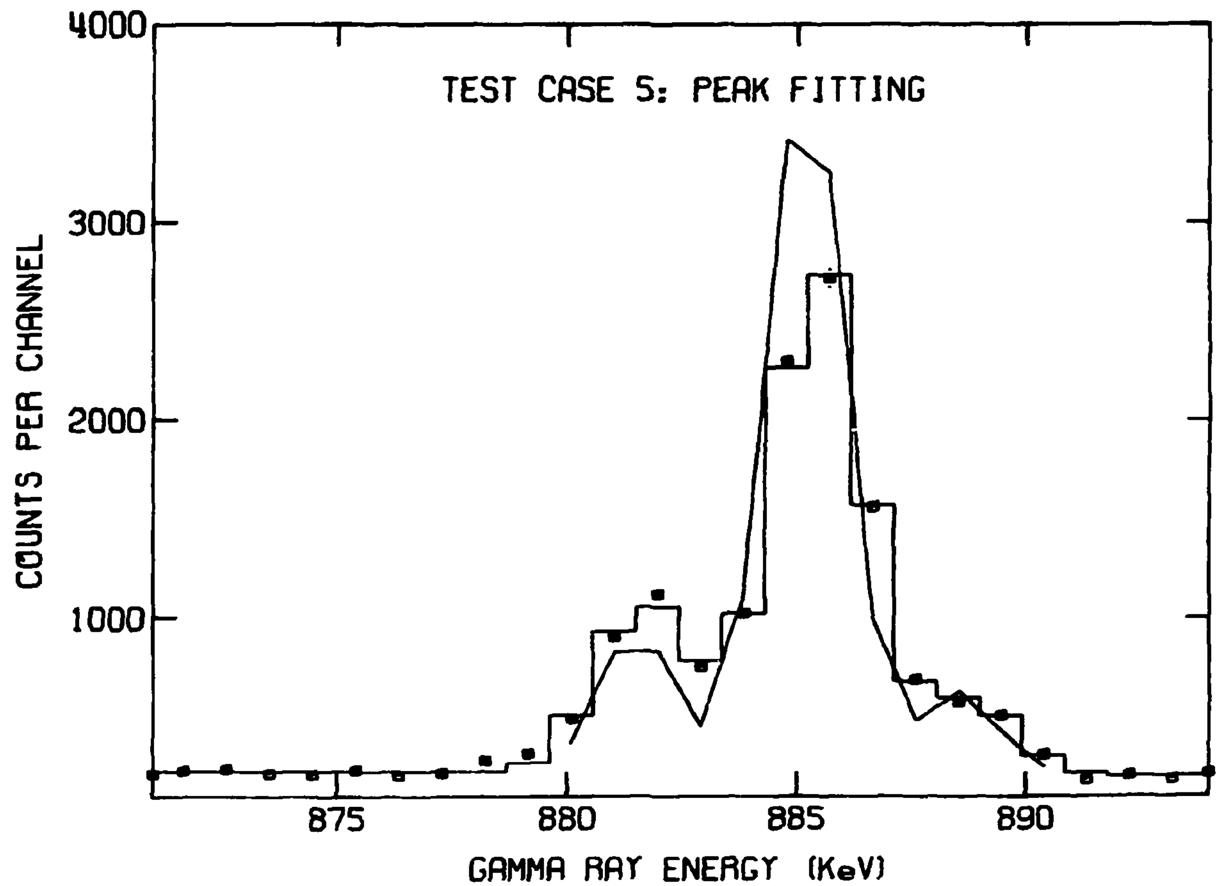

Fig. 3. LSFODF least-squares fit of three exponentials integrated over channel widths and a linear background to a measured Ge(Li) spectrum from a $^{238}U(n,n'\gamma)$ measurement. The histogram curve is the final fit and the other curve is from the initial guesses.

```
TEST CASE 5: PEAK FITTING
 11                    -1
UCSPEC.ODF    927    952   1 28 28
 .881500E+03
 .180000E+01
 .160000E+04
 .885200E+03
 .180000E+01
 .800000E+04
 .888500E+03
 .180000E+01
 .900000E+03
 .000000E+00
 .210000E+03
```

```
      FUNCTION FCTN(X,Z)
      COMMON/DAT/F(3000)
      COMMON/DTA/CHIN,NV2,DOF,PO(18)
      COMMON/FCN/M,N,INTIAL,P(18)
      COMMON/DER/DFDP(18)
      DIMENSION XU(500),XL(500)
      IF (INTIAL.EQ.1) GOTO 5
      DO 7 ID=1,N*5,5
    7 F(ID+2)=SQRT(F(ID+2))
      DO 6 ID=1,N*5-5,5
      J=(ID+4)/5
      XU(J)=(F(ID)+F(ID+5))*.5
      F(ID+4)=FLOAT(J)
    6 XL(J+1)=XU(J)
      F(N*5)=FLOAT(N)
      XU(N)=2.*F(N*5-4)-XL(N)
      XL(1)=F(1)+F(1)-XU(1)
      Z=1.
      L=3
      INTIAL=1
    5 IZ=IFIX(Z)
      Y=0
      TL2=1.66511
      DO 10 I=1,L
      ZP=TL2*(XU(IZ)-P(I*3-2))/P(I*3-1)
      ZM=TL2*(XL(IZ)-P(I*3-2))/P(I*3-1)
      DFDP(I*3)=ERF(ZP)-ERF(ZM)
      Y=Y+P(I*3)*DFDP(I*3)
      S=P(I*3)*.5641896
      EP=EXP(-ZP*ZP)
      EM=EXP(-ZM*ZM)
      DFDP(I*3)=DFDP(I*3)*.5
      DFDP(I*3-1)=S*(EM*ZM-EP*ZP)/P(I*3-1)
   10 DFDP(I*3-2)=S*TL2*(EM-EP)/P(I*3-1)
      Y=Y*.5
      DFDP(L*3+1)=X
      DFDP(L*3+2)=1.
      FCTN=Y+P(L*3+1)*X+P(L*3+2)
      RETURN
      END
```

TEST CASE 5: PEAK FITTING

CONTROLS =
11 10 26 0 0-1 0 0 0 0 5 0.999000 0.750000 0.001000 1.000000

ODF CONTROL = UCSPEC.ODF 927 952 1 28 28

NUMBER OF DATA POINTS = 26
NUMBER OF PARAMETERS =11

ITERATION NUMBER 1

OLD CHI SQUARE = 8.264E+01 NEW CHI SQUARE = 1.261E+01
PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	881.500	881.665	1.285E-01
2	1.80000	2.33564	3.078E-01
3	1600.00	2210.67	2.268E+02
4	885.200	885.305	3.939E-02
5	1.80000	2.06726	9.338E-02
6	8000.00	7103.85	3.399E+02
7	888.500	888.446	2.076E-01
8	1.80000	2.61213	5.044E-01
9	900.000	1169.14	2.116E+02
10	.000000	-1.11429	1.594E+00
11	210.000	1193.12	1.403E+03

ITERATION NUMBER 2

OLD CHI SQUARE = 1.261E+01 NEW CHI SQUARE = 3.593E+00
PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	881.665	881.688	7.772E-02
2	2.33564	2.43000	1.885E-01
3	2210.67	2441.40	1.487E+02
4	885.305	885.402	3.634E-02
5	2.06726	2.20509	9.095E-02
6	7103.85	6957.90	2.607E+02
7	888.446	888.592	2.314E-01
8	2.61213	2.50064	4.916E-01
9	1169.14	1108.71	2.150E+02
10	-1.11429	-1.48151	9.053E-01
11	1193.12	1516.39	7.965E+02

ITERATION NUMBER 3

OLD CHI SQUARE = 3.593E+00   NEW CHI SQUARE = 2.752E+00

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	881.688	881.691	6.830E-02
2	2.43000	2.45842	1.638E-01
3	2441.40	2507.81	1.380E+02
4	885.402	885.450	3.523E-02
5	2.20509	2.26242	9.377E-02
6	6957.90	7001.85	2.375E+02
7	888.592	888.727	2.018E-01
8	2.50064	2.30322	4.195E-01
9	1108.71	1013.56	1.855E+02
10	-1.48151	-1.50209	7.931E-01
11	1516.39	1533.83	6.978E+02

ITERATION NUMBER 4

OLD CHI SQUARE = 2.752E+00   NEW CHI SQUARE = 2.681E+00

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	881.691	881.692	6.727E-02
2	2.45842	2.47074	1.608E-01
3	2507.81	2527.69	1.378E+02
4	885.450	885.464	3.320E-02
5	2.26242	2.27705	9.226E-02
6	7001.85	7010.09	2.185E+02
7	888.727	888.767	1.712E-01
8	2.30322	2.25091	3.712E-01
9	1013.56	993.741	1.566E+02
10	-1.50209	-1.51830	7.754E-01
11	1533.83	1547.79	6.823E+02

ITERATION NUMBER 5

OLD CHI SQUARE = 2.681E+00   NEW CHI SQUARE = 2.675E+00

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	881.692	881.692	6.729E-02
2	2.47074	2.47541	1.606E-01
3	2527.69	2533.19	1.382E+02
4	885.464	885.468	3.278E-02
5	2.27705	2.28175	9.178E-02
6	7010.09	7013.72	2.145E+02
7	888.767	888.780	1.631E-01
8	2.25091	2.23491	3.587E-01
9	993.741	987.784	1.499E+02
10	-1.51830	-1.52428	7.729E-01
11	1547.79	1552.96	6.802E+02

ITERATION NUMBER 6

P SEARCH

NPS	P	CHI SQUARE
0	0.000E-01	2.6743E+00
1	6.326E+01	2.6746E+00
2	3.557E+02	2.6747E+00
3	2.000E+03	2.6748E+00
4	1.125E+04	2.6749E+00
5	6.326E+04	2.6749E+00
0	0.000E-01	2.6743E+00

ETA SEARCH

ETA	CHI SQUARE
0.000E-01	2.6750E+00
7.500E-01	2.6743E+00
1.180E+00	2.6742E+00

OLD CHI SQUARE = 2.675E+00   NEW CHI SQUARE = 2.674E+00

PARAMETERS

NUMBER	OLD VALUE	NEW VALUE	ERROR
1	881.692	881.692	6.736E-02
2	2.47541	2.47841	1.608E-01
3	2533.19	2535.69	1.384E+02
4	885.468	885.469	3.266E-02
5	2.28175	2.28424	9.160E-02
6	7013.72	7016.18	2.133E+02
7	888.780	888.787	1.606E-01
8	2.23491	2.22653	3.548E-01
9	987.784	984.573	1.478E+02
10	-1.52428	-1.52723	7.724E-01
11	1552.96	1555.50	6.798E+02

CORRELATION MATRIX

(1)	1.00											
(2)	0.46	1.00										
(3)	0.41	0.61	1.00									
(4)	0.19	0.22	0.16	1.00								
(5)	-0.50	-0.46	-0.46	0.18	1.00							
(6)	-0.42	-0.40	-0.36	0.15	0.66	1.00						
(7)	-0.25	-0.22	-0.23	0.45	0.65	0.55	1.00					
(8)	0.20	0.21	0.25	-0.44	-0.55	-0.48	-0.63	1.00				
(9)	0.24	0.25	0.30	-0.43	-0.62	-0.49	-0.63	0.80	1.00			
(10)	-0.05	-0.01	-0.05	0.07	0.05	-0.00	0.01	-0.26	-0.32	1.00		
(11)	0.05	0.01	0.05	-0.07	-0.05	0.00	-0.01	0.25	0.32	-1.00	1.00	
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(

X AXIS	Y EXP	Y ERROR	Y CALC
8.7073E+02	2.0748E+02	1.4404E+01	2.2569E+02
8.7166E+02	2.2835E+02	1.5111E+01	2.2426E+02
8.7260E+02	2.3453E+02	1.5314E+01	2.2283E+02
8.7354E+02	2.1067E+02	1.4514E+01	2.2141E+02
8.7447E+02	2.0227E+02	1.4222E+01	2.1998E+02
8.7541E+02	2.2817E+02	1.5105E+01	2.1855E+02
8.7634E+02	2.0103E+02	1.4178E+01	2.1712E+02
8.7728E+02	2.1020E+02	1.4498E+01	2.1413E+02
8.7822E+02	2.7731E+02	1.6653E+01	2.1946E+02
8.7915E+02	3.1261E+02	1.7681E+01	2.6974E+02
8.8009E+02	4.8673E+02	2.2062E+01	5.0561E+02
8.8103E+02	9.0281E+02	3.0047E+01	9.3224E+02
8.8196E+02	1.1174E+03	3.3428E+01	1.0590E+03
8.8290E+02	7.4945E+02	2.7376E+01	7.7762E+02
8.8383E+02	1.0221E+03	3.1970E+01	1.0278E+03
8.8477E+02	2.2952E+03	4.7908E+01	2.2649E+03
8.8571E+02	2.7120E+03	5.2077E+01	2.7358E+03
8.8664E+02	1.5532E+03	3.9411E+01	1.5573E+03
8.8758E+02	6.8256E+02	2.6126E+01	6.6628E+02
8.8852E+02	5.6967E+02	2.3868E+01	5.8480E+02
8.8945E+02	5.0331E+02	2.2435E+01	4.9568E+02
8.9039E+02	3.0527E+02	1.7472E+01	2.9527E+02
8.9132E+02	1.7627E+02	1.3277E+01	2.0762E+02
8.9226E+02	2.0131E+02	1.4188E+01	1.9352E+02
8.9320E+02	1.7952E+02	1.3398E+01	1.9139E+02
8.9413E+02	2.0877E+02	1.4449E+01	1.8995E+02

ACKNOWLEDGMENTS

The authors wish to thank J. Craven for his help with this project and in particular, for writing the subroutine ODFIO used for LSFODF input-output with ODF files. Helpful discussions about the P-search with G. F. Auchampaugh are gratefully acknowledged.

REFERENCES

1. G. F. Auchampaugh, "Mult: A FORTRAN Code for Least-Square Fitting of Neutron Cross-Section Data Using the Reich-Moore Multilevel Formalism," LA-5473-MS (March 1974).
2. G. C. Craven, "OPRODF, A Decsystem-10 Data Manipulation Program for ORELA Data Formated Files," ORNL/CDC/TM-45 (May 1978).
3. The subroutine SLV used to solve equation (6d) was written by G. de Saussure following the Cholesky method described in D. K. Faddeev, and V. N. Faddeeva (1963), "Computational Methods of Linear Algebra" (translated by Robert C. Williams from a Russian book of 1960). San Francisco: W. H. Freeman and Co., 620 pp. 114.
4. D. K. Olsen, G. de Saussure, R. B. Perez, F. C. Difilippo, R. W. Ingle, and H. Weaver, "150-m Measurement of 0.880- to 100.0-keV Neutron Transmissions Through Four Samples of ^{238}U," ORNL/TM-5915 (October 1977).
5. R. H. Moore and R. K. Zeigler, "The Solution of the General Least Squares Problem with Special Reference to High-Speed Computers," LA-2367 (March 1960).
6. D. K. Olsen, G. de Saussure, R. B. Perez, E. G. Silver, F. C. Difilippo, R. W. Ingle, and H. Weaver, *Nucl. Sci. Eng.* 62, 479 (1977).
7. G. de Saussure, D. K. Olsen, and R. B. Perez, "SIOB: A FORTRAN Code for Least-Squares Shape Fitting Several Neutron Transmission Measurements Using the Breit-Wigner Multilevel Formula," ORNL/TM-6286 (May 1978).
8. D. K. Olsen, G. L. Morgan and J. W. McConnel, to be published.
9. Clenshaw, Miller, and Woodger, *Algorithms for Special Functions, in Handbook Series, Special Functions, Numerische Mathematik* 4, (1963) p. 403-419.

APPENDIX

```
      REAL *8 FN,FO
      DOUBLE PRECISION FILE
      COMMON/VAR/FN(210)
      COMMON/DAT/F(1)
      COMMON/DTA/CHIN,MV2,DOF,PO(18)
      COMMON/FCN/M,N,INTIAL,PN(18)
      COMMON/VARI/FO(210)
      DIMENSION IDS(4),TITLE(16)
      OPEN (DEVICE='TTY',UNIT=6)
      OPEN (UNIT=20,FILE='LSFODF.INP')
      READ (20,401) TITLE
      READ (20,402) M,IT2,N,IFDM,IP1,IP2,IP3,IPS,IP2,IES,NPSM,RATIO,
     1ETA,CMIN,CMAX
      IEC=0
      IF (N.EQ.0) READ (20,403) FILE,ISC,IEC,(IDS(I),I=1,3+IFDM)
      DO 10 NV=1,M
   10 READ (20,404) PN(NV)
      CLOSE (UNIT=20)
      IF (N.EQ.0) GOTO 30
      OPEN (UNIT=21,FILE='LSFODF.DAT')
      DO 20 ID=1,N*5,5
      IF (IFDM.LT.2) READ (21,404) F(ID),F(ID+1),F(ID+2)
   20 IF (IFDM.GT.1) READ (21,404) F(ID),F(ID+4),F(ID+1),F(ID+2)
      CLOSE (UNIT=21,FILE='LSFODF.DAT')
      GOTO 50
   30 N=IEC-ISC+1
      INS=3+IFDM
      CALL ODFIO(20,FILE,IFB,INS,INC,IT,J,I)
      DO 40 I=1,3
      J=I
   40 CALL INODF(20,IFB,INS,IDS(J),ISC,N,F(J),5)
      IF (IFDM.EQ.1) CALL INODF(20,IFB,INS,IDS(J),ISC,N,F(5),5)
C INITIALIZE CONSTANTS AND FIRST ITERATICN
   50 IF (IT2.EQ.0) IT2=10
      IF (NPSM.EQ.0) NPSM=5
      IF (RATIO.EQ.0) RATIO=0.999
      IF (ETA.EQ.0) ETA=0.75
      IF (CMIN.EQ.0) CMIN=.001
      IF (CMAX.EQ.0) CMAX=1.
      R=(CMAX/CMIN)**(1./FLOAT(NPSM-1))
      OPEN (UNIT=22,FILE='LSFODF.PRT')
      WRITE (22,401) TITLE
      WRITE (22,406) M,IT2,N,IFDM,IP1,IP2,IP3,IPS,IE2,IES,NPSM,
     1RATIO,ETA,CMIN,CMAX
      IF (IEC.NE.0) WRITE (22,407) FILE,ISC,IEC,(IDS(I),I=1,3+IFDM)
      WRITE (22,408) N,M
      DOF=1./FLOAT(N-M)
      INTIAL=0
      IT=0
      ITS=0
      IP3=IP3-1
      CALL PHICHI
      MV1=M+1
      MV2=M*(M+3)/2
      MV4=MV2+M
      IF (IP1*IP1.EQ.1) WRITE (22,409) IT
      IF (IP2*IP2.EQ.1) CALL ODFGEN(IT,N,IFDM)
      IF (IP1*IP1.EQ.1) GOTO 350
C
C START ITERATION; ZERO NEW MATRIX
   60 IT=IT+1
```

```
      IF (IF2.NE.1) TYPE 400,IT
      WRITE (22,409) IT
      CHIO=CHIN
      DO 70 NV=1,M
      PO(NV)=PN(NV)
   70 PN(NV)=0
      DO 80 NV=MV1,MV4
   80 PN(NV)=0
C COMPUTE VECTOR AND MATRIX; STORE NEW MATRIX
      CALL PHINEQ
      DO 90 NV=1,MV2
   90 PO(NV)=PN(NV)
      CALL SLV(M)
      CALL PHIPRM(ETA)
C
C ENTER P-SEARCH ON DIAGONAL ELEMENTS OF MATRIX
      CHI=CHIN
      IF (IPS) 160,100,110
  100 IF (CHIN.LE.RATIO*CHIO) GOTO 160
  110 WRITE (22,410)
      NPS=0
      NP=0
      P1=0
      WRITE (22,411) NPS,P1,CHIN
C SET P TO LARGEST DIAGONAL ELEMENT
      P=0
      L=0
      DO 120 NV=MV1,2,-1
      L=L+NV
  120 IF (P.LT.PO(L)) P=PO(L)
      P=P*CHIN
C ADD P TO MATRIX DIAGONAL
  130 L=MV1
      DO 140 NV=1,M
      DO 140 NVP=NV,M
      PN(L)=PO(L)
      IF (NVP.EQ.NV) PN(L)=PN(L)+P
  140 L=L+1
      NPS=NPS+1
C INVERT MATRIX AND COMPUTE NEW CHI SQUARED
      CALL SLV(M)
      CALL PHIPRM(ETA)
      WRITE (22,411) NPS,P,CHIN
      IF (P.EQ.P1) GOTO 160
      IF (IF2.EQ.1.AND.CHIN.LE.RATIO*CHIO) GOTO 160
      IF (CHI.LE.CHIN) GOTO 150
      CHI=CHIN
      P1=P
      NP=NPS
  150 P=P*R
      IF (NPS.LT.NPSM) GOTO 130
      P=P1
      NPS=NP-1
      GOTO 130
C
C ENTER ETA SEARCH ON SOLUTION VECTOR LENGTH
  160 IF (IES) 220,170,180
  170 IF (CHIN.LE.RATIO*CHIO) GOTO 230
  180 ETA1=0
      ETA2=ETA
      ETAS=ETA
```

```
      ETA3=ETA+ETA
      CHI1=CHIO
      WRITE (22,412) ETA1,CHI1,ETA2,CHIN
  190 CHI2=CHIN
      CALL PHIPRM(ETA3)
      CHI3=CHIN
      ETAVN=ETA1+0.5*ETAS*(CHI3-4.*CHI2+3.*CHI1)/(CHI3-2.*CHI2+CHI1)
      CALL PHIPRM(ETAVN)
      WRITE (22,405) ETAVN,CHIN
      IF (ABS(ETA2-ETAVN).LT.0.1*ETAVN) GOTO 220
      IF (ABS(CHI2-CHIN).LT.(1.-RATIO)*CHIN) GOTO 220
      IF (ETAVN.LT.ETA3) GOTO 200
      ETA1=ETA3
      CHI1=CHI3
      GOTO 210
  200 IF (ETAVN.LT.ETA2) GOTO 210
      ETA1=ETA2
      CHI1=CHI2
  210 ETA2=ETAVN
      ETAS=ETA2-ETA1
      ETA3=ETA2+ETAS
      GOTO 190
  220 IF (CHIN.GT.RATIO*CHIO) ITS=1
C
C MULTIPLY MATRIX BY CHI SQUARED
  230 IF (NPS.EQ.0) GOTO 250
      DO 240 NV=NV1,NV2
  240 FN(NV)=FO(NV)
      CALL SLV(M)
  250 DO 260 NV=NV1,NV2
  260 FN(NV)=FN(NV)*CHIN
C FIND ERRORS,  STORE IN SOLUTION VECTOR
      L1=NV2
      L=0
      DO 270 NV=NV1,2,-1
      L=L+NV
      L1=L1+1
  270 FN(L1)=DSQRT(FN(L))
C COMPUTE CORRELATION MATRIX
      L=NV1
      DO 280 NV=1,M
      DO 280 NVP=NV,M
      FN(L)=FN(L)/(FN(NV2+NV)*FN(NV2+NVP))
  280 L=L+1
C
C PRINT PARAMETERS, ERRORS
      WRITE (22,413) CHIO,CHIN
      DO 290 NV=1,M
  290 WRITE (22,414) NV,PO(NV),PN(NV),FN(NV+NV2)
C PRINT CORRELATION MATRIX
      IF (IT.GE.IT2) ITS=1
      IF (IF3) 300,310,330
  300 IF (ITS.EQ.0) GOTO 330
  310 WRITE (22,415)
      DO 320 NV=1,M
  320 WRITE (22,416) NV,(FN(M+NV+(NVP-1)*(2*M-NVP)/2),NVP=1,NV)
      WRITE (22,417) (NV,NV=1,M)
C PRINT INPUT DATA AND FIT
  330 IF (IF1-1) 340,350,360
  340 IF (ITS.EQ.0) GOTO 360
  350 IF (IFDM.EQ.0) WRITE (22,418)
```

```
      IF (IFDM.EQ.1) WRITE (22,419)
      IF (IFDM.EQ.0) WRITE (22,420)  ((F(ID+I),I=0,3),ID=1,N*5,5)
      IF (IFDM.EQ.1) WRITE (22,421)  ((F(ID+I),I=0,4),ID=1,N*5,5)
  360 IF (IF2-1) 370,380,390
  370 IF (ITS.EQ.0) GOTO 390
C MAKE ODF FILE FOR PLOT
  380 CALL CDFGEN(IT,N,IFDM)
  390 IF (ITS.EQ.0) GOTO 60
      CLOSE (UNIT=22,FILE='LSFODF.PRT')
C
  401 FORMAT (16A5)
  402 FORMAT (2I3,I6,7I2,I3,4F10.6)
  403 FORMAT (A10,2I6,4I3)
  404 FORMAT (4E12.6)
  406 FORMAT (/'CONTROLS ='/2I3,I6,7I2,I3,4F10.6)
  407 FORMAT (/'ODF CONTROL = ',A10,2I6,4I3)
  408 FORMAT (/'NUMBER OF DATA POINTS =',I4/'NUMBER OF PARAMETERS ='I2)
  409 FORMAT (//'ITERATION NUMBER',I4/)
  400 FORMAT (' ITERATION ',I2)
  410 FORMAT (/12X,'P  SEARCH'/4X,'NPS',6X,'P',12X'CHI SQUARE')
  411 FORMAT (I6,1PE14.3,1PE16.4)
  412 FORMAT (/10X,'ETA SEARCH'/8X,'ETA',9X,'CHI SQUARE',
     12(/1PE14.3,1PE16.4))
  405 FORMAT (1PE14.3,1PE16.4)
  413 FORMAT (/'OLD CHI SQUARE = ',1PE9.3,4X,'NEW CHI SQUARE = ',
     11PE9.3/'PARAMETERS'/3X,'NUMBER',4X,'OLD VALUE',6X,'NEW VALUE',9X,
     2'ERROR')
  414 FORMAT (3X,I3,7X,G12.6,3X,G12.6,2X,1PE12.3)
  415 FORMAT (/'CORRELATION MATRIX')
  416 FORMAT (/' (',I2,')',2X,20F6.2)
  417 FORMAT (/5X,18(2X,'(',I2,')'))
  418 FORMAT (/4X,'X AXIS',8X,'Y EXP',9X,'Y ERROR',7X,'Y CALC')
  419 FORMAT (/4X,'X AXIS',8X,'Y EXP',9X,'Y ERROR',7X,'Y CALC',8X,
     1'Z AXIS')
  420 FORMAT (4(1PE14.4))
  421 FORMAT (5(1PE14.4))
      END
C
      SUBROUTINE PHINEQ
      REAL *8 FN
      COMMON/VAR/FN(210)
      COMMON/DAT/F(1)
      COMMON/DTA/CHIN,NV2,DOF,PO(18)
      COMMON/FCN/M,N,INTIAL,PN(18)
      COMMON/DER/DFDP(18)
C COMPUTE VECTOR AND MATRIX
      DO 10 ID=1,N*5,5
      X=F(ID)
      Z=F(ID+4)
      F(ID+3)=FCTN(X,Z)
      W=1./(F(ID+2)*F(ID+2))
      T1=F(ID+1)-F(ID+3)
      L=M+1
      DO 10 NV=1,M
      FN(NV)=FN(NV)+W*T1*DFDP(NV)
      DO 10 NVP=NV,M
      FN(L)=FN(L)+W*DFDP(NV)*DFDP(NVP)
   10 L=L+1
      RETURN
C
      ENTRY PHIPRM(ETA)
```

```
C COMPUTE NEW PARAMETERS
      DO 20 NV=1,N
   20 PN(NV)=PO(NV)+ETA*PN(NV+NV2)
C
      ENTRY PHICHI
C COMPUTE CHI SQUARED, TC-NEW
      PHI=0
      DO 30 ID=1,N*5,5
      X=F(ID)
      Z=F(ID+4)
      F(ID+3)=FCTN(X,Z)
      T1=(F(ID+1)-F(ID+3))/F(ID+2)
   30 PHI=PHI+T1*T1
      CHIN=PHI*DOF
      RETURN
      END
C
      SUBROUTINE SLV(N)
      IMPLICIT REAL * 8 (A-H,O-Y)
      COMMON/VAR/F(210)
      IF(N-1) 1,2,3
    1 WRITE (6,101)
  101 FORMAT (1H0,'NEGATIVE ARGUMENT IN SLV')
      RETURN
    2 F(2)=1./F(2)
      F(3)=F(2)*F(1)
      RETURN
    3 K=N+1
      DO 10 M=1,N
      M1=M-1
      DO 10 L=M,N
      A=0
      IF(M1.LE.0)   GO TO 6
      KL=N+L
      KM=N+M
      DO 5 LM=1,M1
      A=A+F(KL)*F(KM)
      J=N-LM
      KL=KL+J
    5 KM=KM+J
    6 T=F(K)-A
      IF(L.GT.M)   GO TO 9
      IF(T.GT.0)   GO TO 7
      WRITE (6,102) T
  102 FORMAT (1H0,'FAILURE IN SLV  T=',G20.5)
      T=-T
    7 D=DSQRT(T)
      F(K)=D
      GO TO 10
    9 F(K)=T/D
   10 K=K+1
      K=N+1
      F(K)=1./F(K)
      DO 12 L=2,N
      K=K+N-L+2
      T=1./F(K)
      F(K)=T
      L1=L-1
      KL=N+L
      KM=N
      DO 12 M=1,L1
```

```
      LK=KL
      A=0
      DO 11 LM=M,L1
      I1=KM+LM
      A=A-F(KL)*F(I1)
   11 KL=KL+N-LM
      F(LK)=A*T
      J=N-M
      KL=LK+J
   12 KM=KM+J
      K=N+1
      DO 14 M=1,N
      KL=K
      DO 14 L=M,N
      KM=K
      A=0
      I1=N-L+1
      DO 13 LM=1,I1
      A=A+F(KL)*F(KM)
      KL=KL+1
   13 KM=KM+1
      F(K)=A
   14 K=K+1
      DO 18 L=1,N
      A=0
      IJ=N+L
      JI=0
      J=N-1
      DO 17 M=1,N
      JI=JI+1
      A=A+F(IJ)*F(JI)
      IF(M.GE.L)    GO TO 16
      IJ=IJ+J
      J=J-1
      GO TO 17
   16 IJ=IJ+1
   17 CONTINUE
      F(K)=A
   18 K=K+1
      RETURN
      END
C
      SUBROUTINE ODFGEN(IT,N,ISW)
      DOUBLE PRECISION FILE
      COMMON/DAT/F(1)
C IT         ITERATION NUMBER
C N          NUMBER CHANNELS
C ISW        (NUMBER DATASETS)-4
      ENCODE(10,11,FILE) IT
      IF (IT.LT.100) ENCODE(10,12,FILE)IT
      IF (IT.LT.10) ENCODE(10,13,FILE) IT
      OPEN (UNIT=20,FILE=FILE,ACCESS='SEQOUT')
      CLOSE (UNIT=20)
      NDS=4+ISW
      TYPE 14,FILE,N,NDS
      CALL ODFIO(20,FILE,3,NDS,N,3,-1,3)
      DO 10 I=1,NDS
      J=I
   10 CALL OUTODF(20,3,NDS,J,1,N,F(J),5)
      RETURN
   11 FORMAT('FOR',I3,'.DAT')
```

```
   12 FORMAT('FOR',I2,'.DAT')
   13 FORMAT('FOR',I1,'.DAT')
   14 FORMAT(' CREATING ',A10,I5,' CHANNELS',I4,' DATASETS')
      END
C
      SUBROUTINE ODFIO(IUNIT,FILE,IFB,INS,INC,IMODE,IENER,IRUN)
      DIMENSION ARRAY(1)
      DOUBLE PRECISION FILE
      DIMENSION BUF(125),IBUF(125)
      EQUIVALENCE (BUF(1),IBUF(1))
      DATA ZERO/0/
      OPEN(UNIT=IUNIT,FILE=FILE,ACCESS='RANDOM',RECORD=126)
      READ(IUNIT#1,END=12)IBUF,JUNK
      INS=(IBUF(8).AND."374000000000)/"4000000000
      IF(IBUF(8).LT.0)INS=INS+1
      IFB=(IBUF(8).AND."003770000000)/"10000000
      INC=IBUF(8).AND."7777777
      IMODE=(IBUF(2).AND."30000000)/"10000000
      IENER=IBUF(8)
      IRUN=IBUF(2).AND."7777777
      I=(INC/125)+1
      IF(I*125.NE.INC) I=I+1
      IBLK=IFB+(I*INS)
      READ(IUNIT#IBLK,END=1)JUNK,IBUF
      RETURN
    1 RETURN
C
      ENTRY OUTODF(IUNIT,IFB,INS,ISN,ISC,INC,ARRAY,INDEX)
      IF(INS.LE.0) RETURN
      IF(ISN.LE.0) RETURN
      IF(ISN.GT.INS) RETURN
      IF(ISC.LE.0) RETURN
      IF(INC.LE.0) RETURN
      IBC=1
      ILC=INC
      IBLK=(ISC-1)/125
      I=ISC-(IBLK*125)
      IBLK=(IBLK*INS)+IFB+ISN-1
      IF(I.EQ.1)GO TO 3
      READ(IUNIT#IBLK)JUNK,IBUF
      L=I+ILC-1
      IF(L.GT.125)L=125
      DO 2 J=I,L
      BUF(J)=ARRAY(IBC)
    2 IBC=IBC+INDEX
      WRITE(IUNIT#IBLK)ZERO,IBUF
      ILC=ILC-(L-I+1)
      IF(ILC.EQ.0) RETURN
      IBLK=IBLK+INS
    3 I=ILC/125
      IF(I.EQ.0)GO TO 5
      DO 4 J=IBC,IBC-1+(I*125*INDEX),125*INDEX
      WRITE(IUNIT#IBLK)ZERO,(ARRAY(K),K=J,J+(125*INDEX),INDEX)
    4 IBLK=IBLK+INS
      IBC=IBC+(I*125*INDEX)
      ILC=ILC-(I*125)
      IF(ILC.EQ.0) RETURN
    5 READ(IUNIT#IBLK)JUNK,IBUF
      DO 6 J=1,ILC
      BUF(J)=ARRAY(IBC)
    6 IBC=IBC+INDEX
```

```
      WRITE(IUNIT'IBLK)ZERO,IBUF
      RETURN
C
      ENTRY INODF(IUNIT,IFB,INS,ISN,ISC,INC,ARRAY,INDEX)
      IF(INS.LE.0) RETURN
      IF (ISN.LE.0) RETURN
      IF(ISN.GT.INS) RETURN
      IF (ISC.LE.0) RETURN
      IF(INC.LE.0) RETURN
      IBC=1
      ILC=INC
      IBLK=(ISC-1)/125
      I=ISC-(IBLK*125)
      IBLK=(IBLK*INS)+IFB+ISN-1
      IF(I.EQ.1)GO TO 8
      READ(IUNIT'IBLK)JUNK,IBUF
      IBLK=IBLK+INS
      L=I+ILC-1
      IF(L.GT.125) L=125
      DO 7 J=I,L
      ARRAY(IBC)=BUF(J)
    7 IBC=IBC+INDEX
      ILC=ILC-(L-I+1)
      IF(ILC.EQ.0) RETURN
    8 I=ILC/125
      IF(I.EQ.0)GO TO 10
      DO 9 J=IBC,IBC-1+(I*125*INDEX),125*INDEX
      READ(IUNIT'IBLK)JUNK,(ARRAY(K),K=J,J-1+(125*INDEX),INDEX)
    9 IBLK=IBLK+INS
      IBC=IBC+(I*125*INDEX)
      ILC=ILC-(I*125)
      IF(ILC.EQ.0) RETURN
   10 READ(IUNIT'IBLK)JUNK,IBUF
      DO 11 J=1,ILC
      ARRAY(IBC)=BUF(J)
   11 IBC=IBC+INDEX
      RETURN
   12 DO 13 I=1,125
   13 IBUF(I)=0
      IF(IFB.LE.2) IFB=3
      IBUF(1)="176000001
      IBUF(2)=(IMODE*"10000000).OR.IRUN
      IBUF(7)="2000000
      IBUF(8)=INC.OR.(IFB*"10000000).OR.(INS*"4000000000)
      IF(IENER.LT.0) IBUF(8)=(IBUF(8).OR."400000000000)-"4000000000
      WRITE(IUNIT'1) IBUF,ZERO
      I=(INC/125)+1
      IF(I*125.NE.INC) I=I+1
      IBLK=IFB+(I*INS)
      WRITE(IUNIT'IBLK)ZERO,IBUF
      RETURN
      END
```

ORNL/TM-6545
Dist. Category UC-79d

INTERNAL DISTRIBUTION

1. L. S. Abbott
2. G. T. Chapman
3. J. G. Craven
4. J. W. T. Dabbs
5. J. K. Dickens
6. J. L. Fowler
7. C. Y. Fu
8. H. Goldstein (consultant)
9. R. Gwin
10. J. Halperin
11. J. Harvey
12. D. M. Hetrick
13. N. W. Hill
14. D. J. Horen
15. R. W. Ingle
16. C. H. Johnson
17. D. C. Larson
18. R. L. Macklin
19. F. C. Maienschein
20. J. W. McConnell
21. G. L. Morgan
22-31. ORELA Library
32-51. D. K. Olsen
52. R. W. Peelle
53. C. M. Perey
54. F. G. Perey
55. R. B. Perez
56. S. Raman
57. RSIC
58. G. de Saussure
59. G. G. Slaughter
60. R. R. Spencer
61. J. H. Todd
62. L. W. Weston
63. A. Zucker
64. P. Greebler (consultant)
65. W. B. Lowenstein (consultant)
66. R. E. Uhrig (consultant)
67. R. Wilson (consultant)
68-69. Central Research Library
70. ORNL Y-12 Technical Library Document Reference Section
71. Laboratory Records Dept.
72. ORNL Patent Office
73. Laboratory Records (RC)

EXTERNAL DISTRIBUTION

74. Assistant Manager for Energy Research and Development, DOE-ORO
75. DOE Oak Ridge Operations, Reactor Division, P. O. Box E, Oak Ridge, Tennessee 37830: Director
76-77. DOE Division of Reactor Research and Development, Washington, DC 20545: Director
78-304. For distribution as shown in TID-4500 Distribution Category UC-79d, Liquid Metal Fast Breeder Reactor Physics - Base