27
11-5-80
QB
24 to NTIS

ornl

MASTER

ORNL/TM-7341
ENDF-303

OAK
RIDGE
NATIONAL
LABORATORY

UNION
CARBIDE

GLUCS: A Generalized Least-Squares Program for Updating Cross Section Evaluations with Correlated Data Sets

D. M. Hetrick
C. Y. Fu

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-7341
ENDF-303

Contract No. W-7405-eng-26

Engineering Physics Division

GLUCS: A GENERALIZED LEAST-SQUARES PROGRAM FOR UPDATING CROSS SECTION EVALUATIONS WITH CORRELATED DATA SETS

D. M. Hetrick
Computer Sciences Division

C. Y. Fu
Engineering Physics Division

Sponsor: R. W. Peelle

Research sponsored jointly by the Electric Power Research Institute under Interagency Agreement RP-1399-1 and RTS 78-143 and the Division of Basic Energy Sciences, U. S. Department of Energy, under contract W-7405-eng-26 with the Union Carbide Corporation

Date Published: October 1980

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

TABLE OF CONTENTS

ACKNOWLEDGMENTS

Many thanks go to Dr. Francis G. Perey for helpful discussions throughout the development of this computer code. We are grateful to Betty Waddell for typing this report. This research was sponsored jointly by the Electric Power Research Institute and the U. S. Department of Energy.

ABSTRACT

The PDP-10 FØRTRAN IV computer programs INPUT.F4, GLUCS.F4, and ØUTPUT.F4, which employ Bayes' theorem (or generalized least-squares) for simultaneous evaluation of reaction cross sections, are described. Evaluations of cross sections and covariances are used as input for incorporating correlated data sets, particularly ratios. These data are read from Evaluated Nuclear Data File (ENDF/B-V) formatted files. Measured data sets, including ratios and absolute and relative cross section data, are read and combined with the input evaluations via the least-squares technique. The resulting output evaluations have not only updated cross sections and covariances, but also cross-reaction covariances. These output data are written into ENDF/B-V format.

I. INTRODUCTION

Simultaneous evaluation of cross sections is required where ratio data exist. Since ratio measurements do not require absolute neutron flux determination, which is usually the largest source of uncertainties, correct use of ratio data could establish the relative magnitudes of various cross sections quite well. High-precision absolute data for one reaction would improve the absolute magnitudes of all other cross sections connected by high-precision ratios. In this report, we document a package of computer programs that employ a generalized least-squares technique (Bayes' theorem) for simultaneous evaluation of reaction cross sections. The idea is to combine the new set of measurements with the evaluation, which is assumed to represent older data. Evaluated cross sections and covariances from ENDF/B-V formatted files are used as input. Measured data sets, which are correlated internally due to common components in the measurements, are combined with the evaluations by the least-squares technique. The output evaluations, which are written into ENDF/B-V formatted files, have adjusted cross sections, covariances, and cross-reaction covariances. The code is quite general in that it is capable of handling large numbers of correlated evaluations, a variety of data types that may be correlated, and the most commonly used ENDF/B-V covariance formats and interpolation laws.

Bayes' theorem is described and an example is given in Section II. A summary of the programs with subroutine descriptions follows in Section III. The input data for the codes are thoroughly described in Section IV. In Section V is an explanation of how to execute the programs through use of a sample problem. Finally, a short summary is given in Section VI.

II. BAYES' THEOREM

We have used a property of probability theory referred to as Bayes' theorem to solve the problem: given evaluated data sets (cross sections and covariances) in ENDF/B-V format, incorporate results from other experimental measurements to update the evaluated data sets. The formulas below can be derived from the application of Bayes' theorem of conditional probabilities with normal distribution functions.[1] They can also be obtained as the least-squares estimate when there are no correlations between different sets of data[2], and in this derivation the assumption of normal density functions is not made.

Let the vector T stand for an estimate of the cross sections with a covariance matrix M specifying the uncertainties in this estimate (T and M define a normal probability density function, the "prior"). The vector T and the (nondiagonal) matrix M are retrieved from Files 3 and 33,[3] respectively, of the ENDF/B-V formatted files. Let the vector R stand for a new measurement of cross sections or ratios of cross sections (most likely at different energies) with uncertainties and correlations given by the covariance matrix V (R and V define a normal likelihood function). Then the vector T' representing the updated estimate of the cross sections with covariance matrix M' (T' and M' define the normal "posterior" probability density function) is

$$T' = T + A(N+V)^{-1} (R-R_T) \qquad (1)$$

and

$$M' = M - A(N+V)^{-1} A^t \quad . \qquad (2)$$

Here R_T represents the expected values of R based upon the prior distribution; the interpolation laws of ENDF/B-V are used to calculate R_T. The quantities A and N are given by $A=MG^t$ and $N=GA$, where G is the "sensitivity matrix" of the measurement. The elements g_{ij} of G are the partial derivatives of the i-th element of R_T with respect to the j-th element of T. Thus, the matrix G has the same number of columns as T and the same number of rows as R.

To illustrate, we have $g_{ij} = \frac{\partial (R_T)_i}{\partial T_j}$ where $(R_T)_i = \frac{t(E_{Ri})}{t'(E_{Ri})}$. Here, T_j represents the j-th element of the vector T of cross sections (at energies E_{Tj}) from one or more evaluated data sets. (A set here signifies data that have the same material (or MAT[3]) number). $(R_T)_i$ represents the i-th element of the vector R_T of expected values of R (at measured energies E_{Ri}) for one or more measured sets of data. The expression $t(E_{Ri})$ is the cross section value at energy E_{Ri} that is interpolated from the evaluated data set that corresponds to the material of the measured set. For example, if the evaluated data file specifies that linear-linear interpolation be used, then

$$t(E_{Ri}) = T_k + \frac{T_{k+1} - T_k}{E_{T(k+1)} - E_{Tk}} (E_{Ri} - E_{Tk}) \qquad (3)$$

where $E_{Tk} \leq E_{Ri} < E_{T(k+1)}$. If $(R_T)_i$ is an absolute cross section, then $t'(E_{Ri})$ is 1; however, if $(R_T)_i$ is a ratio, then the computer program that is documented in this paper can handle any of the following three cases:

a) $t'(E_{Ri})$ is similar to $t(E_{Ri})$ except it represents a different reaction.

b) a measured set contains ratios of cross sections (e.g., the $t(E_{Ri})$) of a reaction to the cross section at a fixed energy value for the same reaction. That is, $t'(E_{Ri}) = t(E_{Rf})$ where $t(E_{Rf})$ is the interpolated cross section (e.g., using equation (3)) at the fixed energy value E_{Rf}. (An example is given in Ref. 4.)

c) This case is similar to case (b), only $t'(E_{Ri}) = t'(E_{Rf})$.

To make the above clear, consider the following example. We reproduce a few energy points from the $^{56}Fe(n,p)$ and $^{65}Cu(n,2n)$ ENDF/B-V evaluation files below.

	E_{Tj} (MeV)	$T_j(E_{Tj})$ (b)	STD(%)	C Matrix		
$^{56}Fe(n,p)$						
	14.2	0.1088	2.0	100		
	15.0	0.0994	2.0	81	100	
	16.0	0.0818	2.0	73	81	100
$^{65}Cu(n,2n)$	14.5	0.9080	5.3	100		
	15.5	0.9910	5.2	90	100	
	16.5	1.0470	5.2	86	90	100

The C matrix represents the correlation matrix multiplied by 100 and STD represents the standard deviations. The two cross section sets were evaluated independently, and therefore their uncertainties are uncorrelated. Together they form the vector T and the covariance matrix M of Eqs. (1) and (2). That is,

$$T = \begin{pmatrix} 0.1088 \\ 0.0994 \\ 0.0818 \\ 0.9080 \\ 0.9910 \\ 1.0470 \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} 100 & & & & & \\ 81 & 100 & & & & \\ 73 & 81 & 100 & & & \\ 0 & 0 & 0 & 100 & & \\ 0 & 0 & 0 & 90 & 100 & \\ 0 & 0 & 0 & 86 & 90 & 100 \end{pmatrix}$$

where C is calculated from the symmetric matrix M that is computed from the ENDF/B-V files.

We provide selected energy points from two new measured sets below.[4] They represent cross sections of ^{56}F(n,p) and ratios of ^{65}Cu(n,2n) to ^{56}Fe(n,p).

	E_{Ri} (MeV)	R (b)	STD(%)	C Matrix	
^{56}Fe(n,p)	14.67	0.1081	1.06	100	
	15.3	0.1000	1.84	42	100
		(ratio)			
^{65}Cu(n,2n)/^{56}Fe(n,p)	14.67	8.8300	1.08	100	
	15.52	10.5200	1.52	23	100

The cross-correlation between ^{56}Fe(n,p) and ^{65}Cu(n,2n)/^{56}Fe(n,p) is:

$E_{Ri} \backslash E_{Rj}$	14.67	15.3
14.67	-19	2
15.52	2	-30

These two sets of data represent the data R and V in Eqs. (1) and (2). That is,

$$R = \begin{pmatrix} 0.1081 \\ 0.1000 \\ 8.8300 \\ 10.5200 \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} 100 & & & \\ 42 & 100 & & \\ -19 & 2 & 100 & \\ 2 & -30 & 23 & 100 \end{pmatrix}$$

and V is calculated from C, R, and the standard deviations STD.

Therefore, we have the following (from Eq. (3) since the ENDF/B-V files for ^{56}Fe and ^{65}Cu specified linear-linear interpolation):

$$(R_T)_1 = t\,(E_{R1}) = T_1 + \frac{T_2 - T_1}{E_{T2} - E_{T1}}\,(E_{R1} - E_{T1}) = 0.10328$$

$$(R_T)_2 = t(E_{R2}) = T_2 + \frac{T_3 - T_2}{E_{T3} - E_{T2}} (E_{R2} - E_{T1}) = 0.09412$$

$$(R_T)_3 = \frac{t\,(E_{R3})}{t'\,(E_{R3})} = \frac{T_4 + \dfrac{T_5 - T_4}{E_{T5} - E_{T4}} (E_{R3} - E_{T4})}{T_1 + \dfrac{T_2 - T_1}{E_{T2} - E_{T1}} (E_{R3} - E_{T1})} = 8.9285$$

$$(R_T)_4 = \frac{t\,(E_{R4})}{t'\,(E_{R4})} = \frac{T_5 + \dfrac{T_6 - T_5}{E_{T6} - E_{T5}} (E_{R4} - E_{T5})}{T_2 + \dfrac{T_3 - T_2}{E_{T3} - E_{T2}} (E_{R4} - E_{T2})} = 10.993$$

We can now easily calculate g_{ij}:

$$g_{11} = \frac{\partial (R_T)_1}{\partial T_1} = 1.0 - \frac{E_{R1} - E_{T1}}{E_{T2} - E_{T1}} = 0.4125$$

$$g_{12} = \frac{\partial (R_T)_1}{\partial T_2} = \frac{E_{R1} - E_{T1}}{E_{T2} - E_{T1}} = 0.5875$$

$$g_{22} = \frac{\partial (R_T)_2}{\partial T_2} = 1.0 - \frac{E_{R2} - E_{T2}}{E_{T3} - E_{T2}} = 0.7$$

$$g_{23} = \frac{\partial (R_T)_2}{\partial T_3} = \frac{E_{R2} - E_{T2}}{E_{T3} - E_{T2}} = 0.3$$

$$g_{31} = \frac{\partial(R_T)_3}{\partial T_1} = \frac{-t\ (E_{R3})}{(t'\ (E_{R3}))^2} \left(1.0 - \frac{E_{R3} - E_{T1}}{E_{T2} - E_{T1}}\right) = -35.66$$

$$g_{32} = \frac{\partial(R_T)_3}{\partial T_2} = \frac{-t\ (E_{R3})}{(t'\ (E_{R3}))^2} \left(\frac{E_{R3} - E_{T1}}{E_{T2} - E_{T1}}\right) = -50.79$$

$$g_{34} = \frac{\partial(R_T)_3}{\partial T_4} = \frac{1}{t'\ (E_{R3})} \left(1.0 - \frac{E_{R3} - E_{T4}}{E_{T5} - E_{T4}}\right) = 8.0364$$

$$g_{35} = \frac{\partial(R_T)_3}{\partial T_5} = \frac{1}{t'\ (E_{R3})} \left(\frac{E_{R3} - E_{T4}}{E_{T5} - E_{T4}}\right) = 1.646$$

$$g_{42} = \frac{\partial(R_T)_4}{\partial T_2} = \frac{-t\ (E_{R4})}{(t'\ (E_{R4}))^2} \left(1.0 - \frac{E_{R4} - E_{T2}}{E_{T3} - E_{T2}}\right) = -58.47$$

$$g_{43} = \frac{\partial(R_T)_4}{\partial T_3} = \frac{-t(E_{R4})}{(t'\ (E_{R4}))^2} \left(\frac{E_{R4} - E_{T2}}{E_{T3} - E_{T2}}\right) = -63.34$$

$$g_{45} = \frac{\partial(R_T)_4}{\partial T_5} = \frac{1}{t'\ (E_{R4})} \left(1.0 - \frac{E_{R4} - E_{T5}}{E_{T6} - E_{T5}}\right) = 10.859$$

$$g_{46} = \frac{\partial(R_T)_4}{\partial T_6} = \frac{1}{t'\ (E_{R4})} \left(\frac{E_{R4} - E_{T5}}{E_{T6} - E_{T5}}\right) = .2216$$

and $g_{13} = g_{14} = g_{15} = g_{16} = g_{21} = g_{24} = g_{25} = g_{26} = g_{33} = g_{36} = g_{41} = g_{44} = 0.0$.

Thus, the matrix G is

$$G = \begin{pmatrix} .4125 & .5875 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.7 & 0.3 & 0.0 & 0.0 & 0.0 \\ -35.66 & -50.79 & 0.0 & 8.0364 & 1.646 & 0.0 \\ 0.0 & -58.47 & -63.34 & 0.0 & 10.859 & .2216 \end{pmatrix}$$

and we have the data necessary to calculate T' and M' in Eqs. (1) and (2).

Our computer program uses Eqs. (1) and (2) to solve the problem stated above. Evaluated cross sections and covariances from ENDF/B-V formatted files are used as the "prior" for incorporating new data sets, particularly ratios. Measured data sets, including ratios, and absolute and relative data, are combined with the input evaluations.

The computer code is quite general; however, the user should make note of the following restrictions:

a) the new data cannot be correlated to the old evaluation, i.e., R and T are assumed to be uncorrelated,

b) there are specific classes of ratio data allowed (see above),

c) no significant energy uncertainty is allowed or V needs to be altered,

d) there are no energy resolution problems,

e) specific interpolation laws are used (see Section III),

f) only "NI-type"[5] sub-subsections in file 33 of the ENDF/B-V data can be used to compute the covariance matrix; "NC-type"[5] sub-subsections are not allowed implying that NC in file 33 must be equal to 0.

III. THE MODEL PROGRAMS

The model developed to employ Bayes' theorem for simultaneous evaluation of cross sections for various reactions consists of three different interactive programs called INPUT.F4, GLUCS.F4, and ØUTPUT.F4. (The .F4 extension is used on the local PDP-10 computer and will be used throughout this report.) INPUT.F4 is a utility program that sets up input data for both the GLUCS.F4 and ØUTPUT.F4 programs. These data are evaluated energies, cross sections, and covariances read from ENDF/B-V formatted files. GLUCS.F4 reads the files of evaluated data that were output from INPUT.F4 and reads experimental energies, cross sections (or ratios of cross sections), and covariances (data typed in a file by the user). Subroutines that set up and solve Bayes' equations to yield adjusted cross sections and covariance matrices were borrowed and modified from the general-purpose Bayes' program BAYES[6]. Adjusted values are output in files that are read by the utility program ØUTPUT.F4, which outputs them back into ENDF/B-V format.

Discussions of each of the subprograms for all three programs are given below. A thorough discussion of the input to and output from these programs follows in Sections IV and V, respectively. A listing of the programs can be found in Appendix A. While reading this chapter it may be helpful to look at Appendix B, which contains the results of a sample case that was run on the ORELA PDP-10 computer.

Subroutine Descriptions (INPUT.F4)

MAIN

The MAIN program reads evaluated data from ENDF/B-V formatted files one set at a time, and writes the needed information into files that will later be read by GLUCS.F4 and ØUTPUT.F4. (Again, we define a set here as data that have the same material (or MAT[3]) number.) Run interactively from a terminal, the code prompts the user for the computer name of each of the files containing the evaluated energy, cross section, and covariance information (see Appendix B). The energies and cross sections are written into a file with logical unit number IWE (=3; i.e., FØR03.DAT) that will be read by GLUCS.F4. For the purpose of saving storage in the program, the covariance matrix M is calculated from the ENDF/B formatted data[5] for each set and written out in a temporary file (unit number IWT=12; i.e., FØR12.DAT). That is, the matrix M is calculated in blocks; the covariance of the first evaluated set being labeled block (1,1), the second set being block (2,2), etc. The cross-reaction covariances of matrix M [blocks (2,1), (3,1), etc.] are either set to 0.0 or calculated from the ENDF/B-V formatted data using the LB flag with a value of 3 or 6 (see discussion under subroutine WRILB6 below).[5] Some blocks may be set to 0.0 while others are not. Again, the program prompts the user for this information, asking first if any cross-reaction data will be read. If so, the code asks for the file name containing the cross-reaction data (for each off-diagonal block, one at a time) and then prompts the user to type MAT1 (material number) and MT1 (reaction type identifier)[5] (see Appendix B). Columns of blocks in matrix M are stored in array VAREVD at one time. For example, for

column 1, the program reads the temporary file IWT to retrieve the elements in block (1,1) and then does the calculations for blocks (2,1), (3,1), ..., (NEVSET,1) where NEVSET is the number of evaluated sets. The transpose of this column of blocks is written into a permanent file with unit number IWV (=4; i.e., FØR04.DAT) that will be read by the GLUCS.F4 program. Taking advantage of the symmetry of the matrix, blocks (2,1) (3,1), ..., (NEVSET,1) are stored in a temporary file with logical unit number IWB (=13; i.e., FØR13.DAT). Then, for column 2 of the blocked matrix M, the program retrieves the transpose of block (2,1) from file IWB [for block (1,2)], retrieves block (2,2) from file IWT, and does the calculations for blocks (3,2), ..., (NEVSET,2), storing them in file IWB. Again, the transpose of this column is written into file IWV, to be read later by GLUCS.F4. The program runs until the total matrix is stored in file IWV. We note here that the user may delete files IWT and IWB (FØR12.DAT and FØR13.DAT) after execution of INPUT.F4.

As mentioned above, the reason for handling the matrix M in this way is to save storage. For example, if the largest evaluated set had 75 pairs of points (energies, cross sections), and the total dimension of matrix M was 500 x 500, then the above procedure for calculating M requires storage for 75 x 500 or 37500 elements. If the program were written to store the lower triangular part of the matrix M (since it is symmetric), it would require storage for (500 x 501)/2 or 125250 elements, a factor of over 3 more.

Data are written by MAIN into two other files not mentioned yet. These files have unit numbers IWN (=2; i.e., FØR02.DAT) and IWM (=10;

i.e., FØR10.DAT). Written into file IWN are numbers that are important for dimensioning purposes in both GLUCS.F4 and ØUTPUT.F4. (See the discussion under the GLUCS.F4 and ØUTPUT.F4 subroutines in this section and Section IV for more information regarding these numbers.) The ENDF/B-V material numbers are written into file IWM. These numbers are read by ØUTPUT.F4 only so that the updated data can be written back into ENDF/B-V format. Explanation of these numbers is given in Chapter IV.

If the dimensions of the program need to be increased for a particular problem, the code will type an error message and stop. If this occurs, the user must not only increase the appropriate dimensions in the code, but increase the numbers in the particular IF statement within the code that caused the error statement to be printed.

During execution, MAIN calls two subroutines (PRNT and PRNTRT) and has the option to call a third subroutine (THNLIN) if the user wishes. These subroutines are discussed below.

Subroutine PRNT

This subroutine prints the energies, cross sections, standard deviations, and the correlation matrix of each evaluated set into a file with logical unit number IWP (=11, i.e., FØR11.DAT). These data are printed in a form that can easily be read by the user. The standard deviations are calculated from the diagonal elements of the covariance matrix M and the cross sections. The correlation matrix elements are calculated from the covariance matrix elements, the cross sections, and the standard deviations.

If the user thins an evaluated data set (see discussion under THNLIN below), then PRNT determines which rows (or columns) of the correlation matrix have the same correlation. This information is stored in the array IJTHN. For example, rows (columns) 5-7 in the correlation matrix of the first evaluated set given in Appendix D are fully correlated (or equal).

Subroutine PRNTRT

Subroutine PRNTRT prints the correlation matrix of the rectangular off-diagonal blocks of matrix M into the same file as subroutine PRNT above. This subroutine is called by the MAIN program only if any of the elements of the off-diagonal block being computed are non-zero.

Subroutine THNLIN

THNLIN is an optional subroutine that will thin the energies and cross sections of an evaluated set using linear interpolations with a tolerance of ϵ ($0.0 < \epsilon < 1.0$). The MAIN program prompts the user to type a 1 if the evaluated data are to be thinned (see Appendix B). Then THNLIN asks for the desired ϵ (called EPS in the code). As stated in the discussion under subroutine PRNT above, the array IJTHN contains information about which sections of the covariance matrix M have full correlation. The thinning process (i.e., linear interpolation) is used on the energies and cross sections corresponding to each fully correlated section separately. In other words, the first row (column) of each fully correlated section of the covariance matrix will never be thinned out. This means the energies and cross sections corresponding to that row (column) will be kept, whereas the energies and cross

sections associated with the other rows (columns) of the fully correlated section may be thinned out. For example, the energies and cross sections sections corresponding to the rows (columns) 1,2,5,8,10,12,14,17,19,21, 23,25,27,29,31,33,35,37,39,41,43, and 45 of the first correlation matrix in Appendix D will not be thinned out whereas the others may be, depending on the choice of ϵ . After the energies and cross sections are thinned, THNLIN removes the rows and columns of the appropriate block of the covariance matrix M. For the user's convenience, PRNT is called to print the thinned data.

Subroutine Descriptions (GLUCS.F4)

MAIN

The only function of the MAIN program is to call subroutines PAREXP, PAREVL, VAREVL, and PARNEW. These subroutines are discussed below. No other calculations are done by MAIN. However, two important numbers appear in this routine: NSIZER and NSIZEI. The user can estimate these numbers by doing the following simple calculations:

```
NSIZER = NEXPTS * (5 + NEXPTS + NEVSET + 2 * NEVPTS)
       + NLGEVS * (2 + NEXSET + NEVPTS)
       + 2 * NEXSET + NEVPTS
and NSIZEI = 1/2 * NEXPTS * (3 + NEXPTS)
       + 3 * NEXSET + NEVSET
```

where NEXPTS represents the total number of experimental points in the problem, NEVSET represents the number of evaluated sets, NEVPTS represents the total number of evaluated pairs of points (energies,

cross sections), NLGEVS represents the number of pairs of points in the largest evaluated set, and NEXSET represents the number of experimental sets. NSIZER and NSIZEI signify the total storage (dimensions) needed in the program for the real and integer arrays, respectively. They appear in the A and IA arrays in the CØMMØN statements PARRL and PARIN of MAIN. If the program stops during execution (see discussion under FUNCTIØNS IDIMR and IDIMI below) and prints an error message telling the user to increase the dimensions for either the real variables or integer variables, then the user must increase either NSIZER (for real) or NSIZEI (for integer). Note that the dimension of array A and the number NSIZER must be the same number, and likewise for array IA and NSIZEI.

FUNCTIØN IDIMR

The purpose of this subprogram is to keep track of the dimensions of the real arrays in the program. If this number is ever larger than NSIZER (see above), the following error message will be printed - AVAILABLE SIZE (REAL) = 'NSIZER' BUT YØU NEED ....., and the program will stop. Thus, all the user needs to do is increase NSIZER and the dimension of array A, both found in the MAIN program, recompile, and execute again.

FUNCTIØN IDIMI

The purpose of this subprogram is to keep track of the dimensions of the integer arrays in the program. If this number is ever larger than NSIZEI (see above) the following error message will be printed -

AVAILABLE SIZE (INTEGER) = 'NSIZEI' BUT YØU NEED ....., and the program will stop. Thus, all the user needs to do is increase NSIZEI and the dimension of array IA, both found in the MAIN program, recompile, and execute again.

Subroutine PAREXP

This subroutine determines the dimensions of arrays needed for the experimental parameters that are used in subroutine PAREX (see below) and allocates this storage in arrays A and IA. For example, the total number of experimental points (NEXPTS) is known. PAREXP sets aside a dimension of NEXPTS in array A for the experimental energies and also a dimension of NEXPTS in array A for the measured cross sections (or ratio of cross sections). Therefore if there are six experimental points, then A(1) through A(6) would be assigned as storage for the energies and A(7) through A(12) would be assigned as storage for the cross sections. PAREXP calls subroutine PAREX with an argument list of (A(IEEX), A(IXSEX), ...) where IEX = 1 and IXSEX = 7. When control passes to PAREX, the argument list is (EEX, XSEX, ...), where EEX is the experimental energy and XSEX is the experimental cross section (or ratio of cross sections).

PAREXP reads one card from a file with logical unit number IRX (=1; i.e., FØR01.DAT) that is provided by the user. This card will contain NEXPTS and NEXSET (number of experimental sets). See Section IV for further discussion of this input card.

Subroutine PAREX

Subroutine PAREX reads the experimental data (unit number IRX). These data include the experimental energies, cross sections (or ratio of cross sections), and standard deviations for each set, and information needed to calculate the covariance. In addition, PAREX reads flags that enable the program to recognize ratios (i.e., the ratio of cross sections of one reaction over another). Relative cross sections are treated as special ratios. The input data for PAREX is thoroughly explained in Section IV, and Appendix C has 2 sample input data sets (discussed in Section VI).

For the user's convenience, subroutine PAREX prints out the experimental data in readable form into a file with logical unit number IWP (= 14, i.e., FØR14.DAT). This helps the user determine whether the data were input correctly. There are checks within the routine that will stop execution if an error is detected in the data, and print an error message.

Subroutine PAREVL

PAREVL determines the dimensions of arrays needed for the evaluated parameters, the partial derivatives (sensitivity matrix G), and the theoretical cross sections or ratio of cross sections (R_T, see Section II) that are employed in subroutine PAREV (see below). The storage needed for these arrays is allocated in arrays A and IA as explained under subroutine PAREX above. PAREVL reads the file with unit

number IRN (=2; i.e., FØR02.DAT) that was output from INPUT.F4 to retrieve the numbers needed to determine the dimensions. Section IV has a discussion of this file. Execution is then passed to subprogram PAREV.

Subroutine PAREV

This routine retrieves the evaluated data from file IRE (=3, i.e., FØR03.DAT) that was output from INPUT.F4. These data are printed in readable form into file IWP (=14; i.e., FØR14.DAT). PAREV calculates the partial derivatives (the G matrix - array G in program) and the theoretical cross sections or ratios of cross sections (R_T or array TH in program). Subroutine GTHY (see below) is called to aid in these computations. All ratios are considered in the routine. At the end of PAREV, the theoretical cross sections (or ratios) along with the experimental values and the non-zero values of the sensitivity matrix G are printed in readable form into file IWP (i.e., FØR14.DAT).

Subroutine GTHY

This routine determines in which evaluated energy interval each experimental energy lies. Then it calls subroutine GTINT (discussed below) to compute information needed for the partial derivatives and theoretical cross sections (or ratios) by interpolation.

Subroutine GTINT

GTINT calculates information needed for the partial derivatives and theoretical cross sections (or ratios) by interpolation. The interpolation schemes are those read from the ENDF/B-V formatted files. That is, INPUT.F4 retrieves the interpolation range values and scheme identification numbers from each evaluated set and outputs these data into the file that is read by PAREV. GTINT allows four different interpolation schemes:[3] a) cross section constant in energy, b) cross section linear in energy, c) ln cross section linear in energy, and d) ln cross section linear in ln energy. If the interpolation identification number does not indicate one of these schemes, the program prints an error message and stops.

Subroutine VAREVL

Subroutine VAREVL determines the dimensions of arrays needed for the covariance matrix M of the evaluated data and for calculating MG^t. Again, the storage needed for these arrays is allocated in arrays A and IA. VAREVL then calls VAREV.

Subroutine VAREV

VAREV reads the covariance matrix M (called VAREVD) of the evaluated data from file IRV (=4, i.e., FØRO4.DAT) that was output from INPUT.F4. The subprogram reads rows of blocks of the matrix M at one time (see discussion under the MAIN program of INPUT.F4 above). That is, to save storage, the program reads one row of blocks at a time. For example, say there are three evaluated sets with 6 points (or pairs of points) in the first set, 5 points in the second set, and 7 points in

the third set. The first row of blocks ((1,1), (1,2), (1,3)) would be a 6 x 18 matrix, the second row a 5 X 18 matrix, and the third row a 7 x 18 matrix. Storage is allocated only for the largest row of blocks (i.e., 7 x 18) by subroutine VAREVL. VAREV reads the first row of blocks and then calls subroutine MULMG (see below) to multiply this section of the matrix M times the transpose of the sensitivity matrix G. These calculations are stored in array EMG and execution passes back to VAREV to read the second row of blocks of M. The same storage is used for this section as was the first section. MULMG is called again to do the multiplication, storing the results in the appropriate place in array EMG. These computations continue until the entire matrix M is read and the EMG matrix is calculated.

Subroutine MULMG

The sole purpose of MULMG is to take the product of the covariance matrix M (array VAREVD) and the transpose of the sensitivity matrix G (array G). The results are stored in the array EMG.

Subroutine PARNEW

PARNEW determines the dimensions of arrays needed for computing the new (updated) parameters. Storage needed for these arrays is assigned in arrays A and IA. Then, subroutine PARNE is called.

Subroutine PARNE

This routine makes some final calculations before calling subroutine XSVNEW to compute the adjusted cross sections and covariance matrix. First, it calls subroutine MULGMG to multiply the sensitivity matrix G by the EMG matrix (obtaining $N=GMG^{t}$). These values are stored

in the array EN. Then, the covariance of the experimental data is added to EN (giving N+V; see Section II). Next, the theoretical cross section data are subtracted from the experimental cross section data (obtaining $R-R_T$), with the results being stored in the TH array. Finally, PARNE calls subroutine XSVNEW.

Subroutine XSVNEW

Subroutine XSVNEW computes the new (updated) cross sections T' and covariance matrix M'. First, the routine calculates $(N+V)^{-1}$. Since (N+V) (EN array) is a real symmetric matrix, XSVNEW calls subroutine SSPCO from the LINPACK library[7] (see discussion below) to factorize EN. Then subroutine SSPSL[7] is called to compute $(N+V)^{-1}$ $(R-R_T)$. These results are multiplied by the EMG array to obtain $A(N+V)^{-1}$ $(R-R_T)$ as given in Section II. The evaluated cross section data are read from file IRE (i.e., FØR03.DAT) again and added to $A(N+V)^{-1}$ $(R-R_T)$ to obtain the adjusted cross sections. The evaluated energies and new cross sections are written into file IWE (=15; i.e., FØR15.DAT) and will be read later by ØUTPUT.F4. Subroutine SSPSL is called once more to compute $(N+V)^{-1}$ A^t, which is multiplied by the EMG array to obtain A $(N+V)^{-1}$ A^t. These results are subtracted from the covariance matrix M (read again from file IRV; i.e., FØR04.DAT) to obtain the adjusted covariance matrix M'. The new covariance is written into file IWV (=16; i.e., FØR16.DAT) which will be read later by ØUTPUT.F4.

It is noted here that subroutine SSPCØ estimates the condition number (called RCØND) of the matrix (N+V). If RCØND is so small that the logical expression ((1.0 + RCØND) .EQ. 1.0) is true, then (N+V) is singular to working precision. If this is the case, XSVNEW types a message and execution stops.

Subroutines SSPCØ, SSPSL, and their related subprograms SAXPY, SSPFA, SSCAL, SDØT, SASUM, SSWAP, and ISAMAX will not be discussed here since they have been fully documented.[7] However, they do appear in the listing of the programs in Appendix A.

Subroutine Descriptions (ØUTPUT.F4)

MAIN

The only function of the MAIN program is to call the subroutines PAREXS, PARDIA, and PARREC. (Subroutine PARREC is called only if the number of evaluated data sets is at least 2). However, as in the MAIN program of GLUCS.F4, the two numbers NSIZER and NSIZEI and the arrays A and IA appear. These numbers and arrays have the same purpose here as they do in GLUCS.F4. To estimate NSIZER and NSIZEI, the user should do the following calculations:

```
NSIZER = 2 * NEVPTS + 3 * NEVSET + 7
       + NLGEVS * (3 + NEVPTS + 3 * NEVSET + 2 * NLGEVS)
```

and

```
NSIZEI = NEVSET * (3 * NLGEVS + 1) + NLGEVS * NLGEVS
```

where NEVSET, NLGEVS, and NEVPTS are as defined under the MAIN program of GLUCS.F4.

FUNCTIØN IDIMR and FUNCTIØN IDIMI

These subprograms have the same function in ØUTPUT.F4 as they do in GLUCS.F4. Refer to the discussion given under the subroutine descriptions of GLUCS.F4.

Subroutine PAREXS

This subroutine determines the dimensions of arrays needed for writing the material numbers, energies, and new adjusted cross sections into ENDF/B-V format (MF=3).[3] These dimensions are assigned in arrays A and IA. PAREXS reads the file with unit number IRN (=2; i.e., FØR02.DAT) that was output from INPUT.F4 to retrieve the numbers needed to determine the dimensions. Execution is then passed to subroutine RDMEXS.

Subroutine RDMEXS

RDMEXS reads data from files that were output from both the INPUT.F4 program (unit number IWM=10) and GLUCS.F4 program (unit number IRE=15). File IRM includes the material numbers and file IRE contains the interpolation numbers and the evaluated energies and corresponding adjusted cross sections for each evaluated set. Subroutine WRIEXS is called to write these data into ENDF/B-V format.

Subroutine WRIEXS

This subprogram writes the evaluated energies and adjusted cross sections into ENDF/B-V format. These data are written into files with different logical unit numbers for each evaluated set. The first evaluated set (with adjusted cross sections) is written into a file with

unit number IWR=18 (i.e., FØR18.DAT) while the last set is written into a file with unit number (17 + NEVSET). NEVSET is the number of evaluated sets.

Subroutine PARDIA

Subroutine PARDIA sets up the dimensions in arrays A and IA needed for writing the diagonal blocks of the new adjusted covariance matrix M' into ENDF/B-V format. The definition of blocks here is the same as that discussed under the MAIN program of INPUT.F4 in this chapter. Execution then passes to PARDI.

Subroutine PARDI

Initially, PARDI reads the elements in the first row of blocks (i.e., (1,1), (1,2), ..., (1,NEVSET) blocks) of the adjusted covariance matrix M' from file IRV (=16; i.e., FØR16.DAT) that was output from GLUCS.F4. The matrix elements are first stored in array VARNEW and then the lower triangular elements of the diagonal block (1,1) (corresponds to the first evaluated set) are stored in array VARBLK. Execution passes to subroutine CØRBLK and THNDIA (discussed below). On return from these subroutines, PARDI reads the second row of blocks (stored in VARNEW) and the lower triangular elements of diagonal block (2,2) are stored in VARBLK (assuming more than one set is involved in the problem). Subroutines CØRBLK and THNDIA are called again. This sequence continues until the entire matrix M' in file IRV is read.

Subroutine CØRBLK

This subprogram determines which rows (or columns) of the correlation matrix of the current data set are fully correlated (or equal). The elements of the correlation matrix are calculated from the elements of the covariance matrix M' (the diagonal blocks) and the standard deviations. The information regarding the fully correlated sections of the matrix are stored in array IJTHN. See the discussion under subroutine PRNT of INPUT.F4 in this chapter. The algorithm used in CØRBLK is the same as used in PRNT.

Subroutine THNDIA

THNDIA removes all the stored energies, cross sections, and standard deviations except the first one in each completely correlated (equal) section of the correlation matrix. These thinned data are stored in arrays EEVTHN (energies), XSTHN (cross sections), and STDTHN (standard deviations) for each set. Also, the covariance matrix (a diagonal block of M'-stored in VARBLK) is thinned. That is, the thinned correlation matrix shown on page 161 of Appendix D has no rows (or columns) that have full correlation (with the exception of the last two rows (columns) - the last row of the matrix is always kept). THNDIA then calls subroutines PNTDIA and WRILB5.

Subroutine PNTDIA

For the user's convenience, PNTDIA prints the thinned energies, cross sections, and standard deviations in readable form into file IWP (=17, i.e., FØR17.DAT) for each data set. It computes the correlation matrix from the thinned covariance matrix M' (for each diagonal block) and prints these results also.

Subroutine WRILB5

This subroutine takes the thinned energies, and elements of the covariance matrix M' (i.e., the diagonal blocks) for each data set and writes these data into ENDF/B-V format (MF=33) using the LB=5 flag.[5] The information for each set is written into the appropriate file (unit numbers 18, 19, ..., (17 + NEVSET)). For example, in subroutine WRIEXS (see discussion above) the energies and aujusted cross sections were written in ENDF/B-V format (MF=3) into file FØR18.DAT for the first data set. The thinned diagonal block (1,1) of the matrix M' contains the covariances of the cross section data appearing in FØR18.DAT. Therefore, this covariance information is written in the correct form (LB=5 and MF=33) into FØR18.DAT.

Subroutine PARREC

Subroutine PARREC sets up the dimensions in arrays A and IA needed for writing the off-diagonal rectangular blocks of the new adjusted covariance matrix M' (i.e., cross-reaction covariances) into ENDF/B-V format. Execution then passes to subprogram THNREC.

Subroutine THNREC

As subroutine PARDI, THNREC initially reads the elements in the first row of blocks of the matrix M' from file IRV, storing them in VAR-NEW. Then, THNREC thins the rectangular block (1,2) (cross-reaction covariances between set 1 and set 2) by using the previously stored information about the thinned energies (EEVTHN array). These data are stored in VARBLK. Subroutine PNTREC and WRILB6 are called to write these data into files (see discussion below). On return to THNREC, VARBLK is set to the thinned elements of block (1,3) (cross-reaction covariances between set 1 and set 3) assuming there are more than two data sets. (If not, execution is finished and the program stops). Again, subroutines PNTREC and WRILB6 are called. This sequence continues until the upper triangular blocks of M' have been read and thinned.

Subroutine PNTREC

PNTREC prints the final thinned correlation matrix (computed from M') of each off-diagonal rectangular block of matrix M'. These data are printed in readable form into file IWP (=17; e.g., FØR17.DAT). It is noted here that, whereas subroutine THNREC stores each of the upper triangular blocks of M' in VARBLK, PNTREC prints the transpose of the corresponding correlation matrices. Thus, the lower triangular blocks are printed, being consistent with the output of the program INPUT.F4.

Subroutine WRILB6

WRILB6 takes the thinned energies and cross sections, and cross-reaction covariances and writes these data into ENDF/B-V format (MF=33) using the LB=6 flag. The LB=6 flag has not been documented. However,

in order to allow the cross-reaction covariance matrix components to be entered efficiently into the ENDF/B-V files, the LB=6 flag was developed. This flag is similar to the LB=5 flag.[5] The LB=5 flag can be used on square matrices only, while the LB=6 flag here was developed especially for the rectangular off-diagonal cross-reaction covariance matrices. We note that the LB=3 flag could be used but it would be very inefficient.[5] However, the INPUT.F4 program described above can read the cross-reaction covariance components with either the LB=3 or LB=6 flag.

The following definition applies for LB=6 sub-subsections:

LB=6 Relative covariance matrix components

$$\mathrm{COV}(X_i, Y_j) = \sum_{k,l} P^{i;k}_{j;l} F_{xy;k,l} X_i Y_j \quad .$$

The dimensionless operator P and the cross sections X_i and Y_j are as defined in Ref. 5. A single list of energies E_{kl} is required to specify the energy intervals labeled by the indices k and l. That is, a single array contains the energies for both the columns (E_k) and rows (E_l) of the off-diagonal matrix with the energies corresponding to the columns given first. The numbers $F_{xy;k,l}$ represent fractional components correlated over the energy intervals E_k and E_l. The structure of an LB=6 sub-subsection is (see Ref. 3):

[MAT, 33, MT/b,b;b,LB=6;NC,NR/{E_{kl}} {$F_{k,l}$}] LIST.

NC is the number of energies corresponding to the columns of the off-diagonal matrix defining (NC-1) energy intervals.

NR is the number of energies corresponding to the rows of the off-diagonal matrix defining (NR-1) energy intervals.

The matrix elements $F_{k,l}$ are ordered by rows in the array $\{F_{k,l}\}$:

$$\{F_{k,l}\} = F_{1,1}, F_{1,2}, \cdots, F_{1,NC-1}; F_{2,1}, \cdots, F_{2,NC-1};$$
$$\cdots, F_{NR-1,1}, F_{NR-1,2}, \cdots, F_{NR-1,NC-1}.$$

There are (NC-1) (NR-1) numbers in the array $\{F_{k,l}\}$. Therefore, the total number of entries in the two arrays $\{E_{k,l}\}$ and $\{F_{k,l}\}$ is NC + NR + (NC-1) (NR-1).

As in subroutine WRILB5, the information from each off-diagonal block is written into the appropriate file (unit numbers 18, 19, ..., (17 + NEVSET)). For example, block (1,2) is written in the correct form (LB=6, MF=33) into FØR19.DAT and the transpose (i.e., block (2,1)) is written into file FØR18.DAT. In other words, information in the columns of blocks of matrix M' are written into the same file.

IV. INPUT TO THE MODEL PROGRAMS

This section gives a complete summary of the READ statements along with their FØRMAT statements and a brief description of the input parameters for each of the model programs. Prompting messages and user responses are described. It is noted here that the only data file that the user needs to type is the experimental data file (discussed under subroutines PAREXP and PAREX below); other data are read directly from ENDF/B-V formatted files or from files generated by another of the programs. This summary makes it possible for the user to generate the experimental data file, and familiarizes the reader with the data that are read from the ENDF/B-V files and with the parameter definitions.

We stress that the programs read files with different logical input device numbers. For example, the unit number used in the PAREX subroutine of GLUCS.F4 to read the experimental data is IRX=1 (i.e., file would be FØR01.DAT). If the user wishes to change this number, only one FØRTRAN statement within subroutine PAREX (i.e., IRX=1) would have to be altered. (Note: IRX would have to be changed in subroutine PAREXP also). The other logical device numbers used are given in the discussion below.

The user is referred to Appendix B for the teletype output for a sample problem run on the ORELA PDP-10 computer and to Appendix C for the sample input data for the same problem. Again, note that part of

the input data for one program may be the output from the previous program. FØR02.DAT shown in Appendix C as input for GLUCS.F4 is actually a file produced by INPUT.F4. (Notice that the sample problem given in the Appendices required two runs of each of the programs. See Section V for a more complete discussion.)

(Note: reasonable proficiency with the FØRTRAN IV computer language is needed by the user to fully understand the READ statements, FØRMAT statements, DØ loops, etc., that appear in this chapter.)

Input Data for INPUT.F4

Data Typed by the User During Execution

The MAIN program prompts the user to type the number of evaluated sets being considered in the problem (see Appendix B). This number must be ≤ 10 in this version because of dimensioning. Then, the code tells the user to type a 1 if any cross-reaction covariances will be read. The program asks for the names of the files containing the ENDF/B-V formatted data. For each data set, the program cues the user to type MT (reaction type identifier) and to type a 1 if the data are to be thinned. If the data in a particular file are to be thinned, subroutine THNLIN asks for EPS (the tolerance ϵ; $0 < \epsilon < 1$). Finally, MAIN prompts the user to type a 1 if the cross-reaction covariances for two data sets are to be read. If so, MAIN asks for the file name containing the cross-reaction covariances and for MAT1 and MT1.[5]

Appendix C contains the first evaluated ENDF/B-V formatted file that was used in the sample problem. The others are omitted to save space.

Data Read by the MAIN Program

The following READ statements read the data from the ENDF/B-V formatted files.[3] They appear here as they do in the listing in Appendix A. The user does not have to type these data. We include the following to familiarize the reader with the data that are read from the ENDF/B-V files and with the parameter definitions.

```
      :
      IRD= 22
      MFF=  3
   30 READ(IRD,35)MAT,MF,MT
   35 FORMAT(66X,I4,I2,I3)
      IF(MF.NE.MFF.ØR.MT.NE.MTT)GØ TØ 30
      :
```

IRD = logical input data file number used in the READ statements.

MFF, MF = ENDF/B-V file number.

MAT = ENDF/B-V material number.

MT = ENDF/B-V reaction type identifier.

MTT = MT that is typed by user during execution (refer to Appendix B).

```
      :
      BACKSPACE IRD
      READ(IRD,40)ZA,AWR
   40 FØRMAT(2E11.4)
      :
```

ZA = the (Z,A) designation for the material.

AWR = the ratio of the mass of an atom (or molecule) to that of the neutron.

```
            :
            READ(IRD,45)Q,NR,NPSET
       45   FØRMAT(11X,E11.4,22X,2I11)
            :
Q               = reaction Q-value (eV).
NR              = the number of different interpolation ranges
                  in the evaluated set (must be ≤3 because of
                  dimensioning).
NPSET           = number of pairs (energies, cross sections) in
                  the evaluated set (must be ≤75 because of
                  dimensioning).
            :
            READ(IRD,50)(NBT(I),INTP(I),I=1,NR)
       50   FØRMAT((6I11))
            :
NBT(I)            = number value separating the I-th and the
                    (I + 1)-th interpolation ranges.
INTP(I)           = the interpolation scheme identification number
                    used in the I-th range.
            :
            READ(IRD,55)(ESET(I),XSSET(I),I=1,NPSET)
       55   FØRMAT(6E11.4)
            :
ESET(I)           = energies for the evaluated set (eV)
XSSET(I)          = cross sections for the evaluated set (b).
            :
            MFF = 33
       60   READ(IRD,35)MAT,MF,MT
            IF(MF.NE.MFF.ØR.MT.NE.MTT)GØ TØ 60
            BACKSPACE IRD
```

```
            READ(IRD,65)NL,MF,MT
       65   FØRMAT(55X,I11,4X,I2,I3)
            :
MFF,MF,MT,MTT are as defined above.
NL              = number of subsections within the section.[5]
            :
       68   READ(IRD,70,END=72)AMAT1,MT1,ANC,ANI
       70   FØRMAT(22X,4F11.0)
            IF(AMAT1.EQ.0.0.AND.AMT1.EQ.FLØAT(MTT).AND.ANC.EQ.0)GØ TØ 74
            GØ TØ 68
       72   TYPE 73
       73   FØRMAT(1X,'INCØRRECT DATA (MF=33) - STØP')
            STØP
            :
AMAT1           = 0 here, signifying that MAT1 = MAT.[5]
AMT1            = MTT (or MT).
ANC             = number of "NC-type" sub-subsections[5]; will
                  be 0 here.  (See the list of restrictions
                  given in Section II.)
ANI             = number of "NI-type" sub-subsections.[5]
            :
       74   NI=IFIX(ANI)
            DØ 190 L=1,NI
            READ(IRD,75)LS,LB,NT,NP
       75   FØRMAT(22X, 4I11)
            :
LS              = a flag indicating whether the F matrix below
                  is symmetric or not.  LS is used only when
                  LB=5 and should equal to 1 since the F matrix
```

is always symmetric[5]. If LS≠1, the program prints an error message and stops (see below).

LB = a flag that determines the meaning of the numbers in arrays EKL and F below.[5] The program handles LB=0, 1, or 5 in this section.

NT = the total number of entries in the two arrays EKL and F; used when LB=5 only (see below). (NT must be ≤1332 because of dimensioning).

NP = the total number of pairs of numbers (EKL, F arrays) if LB=0 or 1.

= the number of entries in the array EKL if LB=5. NP must be ≤75 because of dimensioning.

```
      :
      IF(LB.EQ.5)GØ TØ 140
C LB=0 or LB=1 here.
      READ(IRD,55)(EKL(K),F(K),K=1,NP)
      :
```

EKL(K) = energies (eV).

F(K) = covariance component associated with the energy interval between the two entries EKL(K) and EKL(K+1).[5]

```
      :
C LB=5 here.
  140 IF(LS.EQ.1)GØ TØ 148
      TYPE 145,LS
  145 FØRMAT(1X,'LS =  ',I2,'MATRIX NØT SYMMETRIC - STØP')
      STØP
  148 NT = NT-NP
```

```
            READ(IRD,55)(EKL(K),K=1,NP),(F(K),K=1,NT)
      190   CØNTINUE
            :
            :
EKL(K)           = energies (eV).

F(K)             = fractional components correlated over the
                   energy intervals EKL(K) and EKL(K').[5]
```

In the above, data read included the energies and cross sections from the MF=3 file and the covariances of the cross sections from the MF=33 file. The following gives the READ statements in the code for reading the cross-reaction covariances from the MF=33 file.

```
            :
            :
      305   READ(IRD,310,END=72)AMAT1,AMT1
            IF(AMAT1.NE.AMATL1.ØR.AMT1.NE.AMTN1)GØ TØ 305
      310   FØRMAT(22X, 2F11.0)
            :
            :
AMATL1           = MAT1 that was typed by the user during
                   execution[5] (refer to Appendix B).
AMAT1            = MAT1.
AMTN1            = MT1 that was typed by the user during
                   execution[5] (refer to Appendix B).
AMT1             = MT1.
            :
            :
            BACKSPACE IRD
            READ(IRD,75)MAT1,MT1,NC,NI.
            :
            :
MAT1 and MT1 are as above.
NC               = number of "NC-type" sub-subsections.[5]
NI               = number of "NI-type" sub-subsections.[5]
```

```
      :
      DØ 350 L=1,NI
      READ(IRD,75)LT,LB,NC,NR
      :
```

In order to read the cross-reaction covariances, LB must be equal to 3 or 6 here. If LB=3, there are two sets of pairs of numbers. Each set is referred to as an E table - the E_k table and the E_l table.[5] LT is equal to the number of pairs in the E_l table, NR is equal to the total number of pairs, and NC is not used. If LB=6, then LT is not used, and NC corresponds to the number of energies in one set ((NC-1) is equal to the number of columns in the cross-reaction covariance matrix in the ENDF/B-V file) and NR corresponds to the number of energies in the other set ((NR-1) is equal to the number of rows in the cross-reaction covariance matrix in the ENDF/B-V file).

LT = number of pairs of numbers in the second table (E_l table) when LB=3.
= 0 when LB=6.

LB = 3 or 6, the numerical value that defines the representation of the cross-reaction covariance matrix components in the ENDF/B-V formatted file.

NC = a dummy value if LB=3.
= number of energies in the first set if LB=6.

NR = total number of pairs of numbers in the E_k and E_l tables if LB=3.
= number of energies in the second set if LB=6.

(If LB=3, NR must be ≤ 75 because of dimensioning. IF LB=6, then NC+NR must be ≤ 75 and (NC-1) * (NR-1) must be ≤ 1332 because of dimensioning.)

```
      :
      GØ TØ (311,311,313,311,311,316),LB
311   TYPE 312,LB
312   FØRMAT(IX,'STØP: LB = ',I2,' (SHØULD BE 3 ØR 6)')
      STØP
313   NP=NR
      IF(NP.LE.75)GØ TØ 315
      TYPE 314,NP
314   FØRMAT(IX,'STØP - INCREASE DIMENSIØNS: NP = ',I3)
      STØP
315   READ(IRD,55)(EKL(KL),F(KL),KL=1,NP)
      :
      GØ TØ 319
      :
```

EKL(KL) = energies (eV); energies in both the E_k and E_l tables are stored in this array (LB=3 here).

F(KL) = number associated with the energy interval between the two entries EKL(KL) and EKL(KL+1).[5] Numbers corresponding to both the E_k and E_l tables are stored in this array.

```
      :
316   NP=NC+NR
      NCM1=NC-1
      NPF=NCM1*(NR-1)
      IF(NP.LE.75.AND.NPF.LE.1332)GØ TØ 318
```

```
      TYPE 317,NP,NPF
  317 FØRMAT(IX,'STØP, INCREASE DIMENSIØNS: NP= ',I3,' NPF= ',I4)
      STØP
  318 READ(IRD,55)(EKL(KL),KL=1,NP),(F(KL),KL=1,NPF)
      .
      .
      .
  319 CØNTINUE
      .
      .
      .
  350 CØNTINUE
      .
      .
      .
```

EKL(KL) = energies (eV); energies for both sets are stored in this array (LB=6 here).

F(KL) = fractional components correlated over the two energy intervals. For example, if NC=4 and NR=2 implying that the F matrix is 1 by 3, then F(3) is the component associated with two energy intervals with entries EKL(3) and EKL(4) being the first interval and EKL(5) and EKL(6) being the second interval.

The other READ statements that appear in the MAIN program read the temporary files that were written by the program and will not be given here. Refer to the discussion under the MAIN program of INPUT.F4 in Section III.

Input Data for GLUCS.F4

Data Typed by the User During Execution

Subroutine PAREX will prompt the user to type in the fixed energy value (in MeV) if a relative measurement was made for the experimental data. That is, the experimental data for a particular reaction were obtained from a shape measurement, combined with a ratio measurement at a fixed energy. See Appendix B for the prompting message typed by GLUCS.F4 during execution.

The discussion given below for data read by subroutines PAREXP and PAREX tells the user how to type the experimental file for all cases. Refer to Appendix C where the experimental data for the two cases involved in the sample problem are given. Both are named FØRO1.DAT.

Data Read by Subroutine PAREXP

```
      :
      IRX = 1
      READ(IRX,10)NEXPTS,NEXSET
   10 FØRMAT(1X,2I3)
      :
```

IRX = logical input data file number. This is the unit number used for the experimental data file that is typed by the user. For the PDP-10 computer, the name of this file would be FØRO1.DAT.

NEXPTS = total number of experimental points in all experimental sets.

NEXSET = number of experimental sets.

Data Read by Subroutine PAREX

```
      :
      IRX=1
      NINEX=1
      DØ 60 NEX=1,NEXSET
      READ(IRX,10)NEXP(NEX),ISTEVN(NEX),ISTEVD(NEX)
   10 FØRMAT(1X,3I3)
      :
```

NEXP(NEX) = for example, if NEXSET=3, NEXPTS=16, and there are 4 points in the first experimental set and 6 points in the second experimental set (leaving 6 in the third set), then NEXP(1)=4, NEXP(2)=10, and NEXP(3)=16.

ISTEVN(NEX) = the number of the evaluated data set that corresponds to the numerator of the ratio (of cross sections) for experimental data set NEX. For example, the data for the first experimental set for the second run in the sample problem (see FØR01.DAT, page 143, Appendix C) are from the ratio $^{56}Fe(n,p)/^{32}S(n,p)$. The numerator $^{56}Fe(n,p)$ corresponds to the first evaluated data set in the execution of INPUT.F4 (see page 129, Appendix B). Therefore ISTEVN(1) = 1.

ISTEVD(NEX) = the number of the evaluated data set that corresponds to the denominator of the ratio (of cross sections) for experimental data set NEX. Set ISTEVD(NEX) = 0 if there is no ratio. In the example above, ^{32}S (n,p) was the third evaluated set in the problem (second run), signifying that ISTEVD(1) = 3. If the ratio involves a relative measurement, then ISTEVD(NEX) should be set to - ISTEVN(NEX). For example, in the third experimental set of the same data mentioned above, the data given are ratios of ^{56}Fe (n,p) cross sections to the same reaction cross section at a fixed energy value of 14.5 MeV. Therefore, ISTEVN(3) = 1 and ISTEVD(3) = -1. When the program determines that ISTEVD(3) is negative, it prompts the user to type in the fixed energy value discussed above.

```
      :
      DØ 20 J = NINEX,NEXP(NEX)
      READ(IRX,30)EEX(J),XSEX(J),STDEX(J)
   20 EEX(J) = EEX(J)*1.0E+6
   30 FØRMAT(1X,3(F10.4,1X))
      NINEX = NEXP(NEX)+1
   60 CØNTINUE
      :
```

EEX(J) = the experimental energy (in MeV).

```
XSEX(J)          = the experimental cross section (in barns) or
                   the ratio of cross sections (dimensionless).
STDEX(J)         = standard deviation (%).
                 .
                 .
                 .
                 READ(IRX,10)IFLAG
                 .
                 .
                 .
IFLAG                 = a control variable for whether the code reads
                        the experimental correlation matrix or relative
                        covariance matrix.  The elements of either of
                        these matrices are usually easier to type into
                        a file than the covariance matrix.  The code
                        then converts either ICØR or VAREXD below to
                        the experimental covariance matrix.
                      = 0 implies that the correlation matrix is read.
                        (See FØR01.DAT for first run of sample problem,
                        page 136 of Appendix C.)
                      = 1 implies that the relative covariance matrix
                        is read.  (See FØR01.DAT for second run of
                        sample problem, pages 143-149 of Appendix C.)
                 .
                 .
                 .
                 IF(IFLAG.EQ.1)GØ TØ 78
                 READ(IRX,70)(ICØR(I),I=1,NCØVEX)
           70    FØRMAT(20(I3,1X))
                 .
                 .
                 .
                 GØ TØ 90
                 .
                 .
                 .
```

ICØR(I) = lower triangular elements of the experimental correlation matrix (matrix is symmetric) by rows. The correlation coefficients should be multiplied by 100; thus, the diagonal elements are 100 (dimensionless).

NCØVEX = number of lower triangular elements in the matrix (= NEXPTS*(NEXPTS+1)/2).

```
      :
      :
   78 READ(IRX,80)(VAREXD(I),I = 1,NCØVEX)
   80 FØRMAT(8F10.4)
      :
      :
   90 CØNTINUE
      :
      :
```

VAREXD(I) = lower triangular elements of the experimental relative covariance matrix (matrix is symmetric) by rows. ($\%^2$)

Data Read by Subroutine PAREVL

```
      :
      :
      IRN  = 2
      READ(IRN,10)NEVPTS,NEVSET,NLGEVS
   10 FØRMAT(1X,6I3)
      :
      :
```

IRN = logical input data file number. This file was output from the INPUT.F4 program (into FØR02.DAT). Therefore, the user does not have to type these data.

NEVPTS = total number of evaluated points (all sets).

NEVSET = number of evaluated sets.

NLGEVS = number of points in the largest evaluated set.

Data Read by Subroutine PAREV

```
            :
            IRE  = 3
            DØ 110 NEV=1,NEVSET
            READ(IRE,20)NPSET,NEVP(NEV),NR
            :
```

IRE = logical input data file number. This file was output from the INPUT.F4 program (into FØR03.DAT). Therefore, the user does not have to type these data.

NPSET = number of points in set NEV.

NEVP(NEV) = for example, if NEVSET=3, NEVPTS=16, and there are 4 points in the first evaluated set, and 6 in the second evaluated set (leaving 6 in the third set), then NEVP(1)=4, NEVP(2)=10, and NEVP(3)=16.

NR = number of interpolation ranges for set NEV.

```
            :
            READ(IRE,20)(NBT(I),INTP(I),I=1,NR)
      20    FØRMAT(1X,6I3)
            :
```

NBT(I) = number value separating the I-th and (I+1)-th interpolation ranges.

INTP(I) = interpolation scheme identification number used in the I-th range.

```
            :
            DØ 30 I=1,NPSET
      30    READ(IRE,40)EEV(I),XSEV(I)
```

```
       40   FØRMAT(1X,1PE11.4,1X,1PE11.4)
       110  CØNTINUE
            :
            .
```

EEV(I) = evaluated energies in set NEV (eV).

XSEV(I) = evaluated cross sections in set NEV (b).

Data Read by Subroutine VAREV

```
            :
            .
            IRV=4
            NPSET=NEVP(1)
            DØ 40 NEV=1,NEVSET
            IF(NEV.NE.1)NPSET=NEVP(NEV)-NEVP(NEV-1)
            DØ 20 J=1,NPSET
       20   READ(IRV,30)(VAREVD(I,J),I=1,NEVPTS)
       30   FØRMAT(1X,10(1PE11.4))
       40   CØNTINUE
            :
            .
```

IRV = logical input data file number. This file was output from the INPUT.F4 program (into FØRO4.DAT). Therefore, the user does not have to type these data.

VAREVD(I,J) = covariance matrix elements for the evaluated data (b^2).

Data Read by Subroutine XSVNEW

Subroutine XSVNEW reads data from files IRE (FØRO3.DAT) and IRV (FØRO4.DAT) again. These data files were rewound after they were read the first time and the same data that were read from files IRE and IRV are read again. Thus, we will not discuss the READ statements here.

Input Data for ØUTPUT.F4

The user does not have to type in data during execution of ØUTPUT.F4, nor does the user type any files in preparation for execution. The program reads data files that were output from both the INPUT.F4 and GLUCS.F4 programs. These data are summarized in the following.

Data Read by Subroutine PAREXS

```
      .
      .
      IRN = 2
      READ(IRN,10)NEVPTS,NEVSET,NLGEVS
10    FØRMAT(1X,3I3)
      .
      .
```

IRN,NEVPTS,NEVSET, and NLGEVS have the same definition here as in the discussion under subroutine PAREVL of GLUCS.F4 above.

Data Read by Subroutine RDMEXS

```
      .
      .
      IR = 15
      II = 0
      DØ 50 NEV=1,NEVSET
      READ(IRE,10)NPSET,NEVP(NEV),NR
10    FØRMAT(1X,6I3)
      .
      .
```

```
IRE             = logical input data file number.  This file was
                  output from the GLUCS.F4 program (into FØR15.DAT).
```

NPSET,NEVP(NEV),and NR have the same definitions here as given under subroutine PAREV above.

```
                :
                READ(IRE,10)(NBT(I),INTP(I)=1,NR)
                :
```

NBT(I) and INTP(I) have the same definitions here as given under subroutine PAREV above.

```
                :
                DØ 20 I=1,NPSET
                II=II+1
        20      READ(IRE,30)EEV(II),XSNEW(II)
        30      FØRMAT(1X,1PE11.4,1X,1PE11.4)
                :
EEV(II)             = evaluated energies (eV).
XSNEW(II)           = new adjusted cross sections as calculated by
                      GLUCS.F4 (b).
                :
                IRM = 10
                READ(IRM,40)MAT(NEV),MT(NEV),Q(NEV),ZA(NEV),AWR(NEV)
        40      FØRMAT(1X,I4,I3,1X,3(1PE11.4))
        50      CØNTINUE
                :
IRM                 = logical input data file number.  This file was
                      output from INPUT.F4.
MAT(NEV)            = ENDF/B-V material number for set NEV.
MT(NEV)             = ENDF/B-V reaction type identifier for set NEV.
Q(NEV)              = reaction Q-value for set NEV (eV).
```

ZA(NEV) = the (Z,A) designation for the material for set NEV.

AWR(NEV) = the ratio of the mass of an atom to that of the neutron for set NEV.

Data Read by Subroutine PARDI

```
      :
      :
      IRV = 16
      DØ 60 NEV=1,NEVSET
      DØ 20 J=1,NPSET
   20 READ(IRV,30)(VARNEW(I,J),I=1,NEVPTS)
   30 FØRMAT(1X,10(1PE11.4))
   60 CØNTINUE
      :
      :
```

IRV = logical input data file number. This file was output from the GLUCS.F4 program.

NEVSET = number of evaluated sets.

NPSET = number of points in set NEV. This number is calculated from NEVP(NEV) within the code.

VARNEW(I,J) = elements of the new adjusted covariance matrix calculated by Bayes theorem (b^2).

Data Read by Subroutine THNREC

Subroutine THNREC reads the adjusted covariance matrix elements from file IRV (FØR16.DAT) again. That is, this file was rewound after it was read the first time and THNREC reads it once more.

V. SAMPLE PROBLEM

Appendix B contains the teletype output for a sample problem using the INPUT.F4, GLUCS.F4, and ØUTPUT.F4 programs. Notice that the problem requires two runs (or cycles) of the programs. (The two cycles will become clear if the reader refers to Ref. 4.) In the first run, there are two evaluated sets, the first one being ^{56}Fe(n,p) (in file FE56.DAT) and the second one ^{65}Cu(n,2n) (in file CU65.DAT). For the user's convenience, the evaluated data for ^{56}Fe(n,p) are given in Appendix C; the data for ^{65}Cu(n,2n) are not included in order to save space. Note that no cross-reaction covariances were read by INPUT.F4 for this first run, and that the data for each set were thinned with ϵ=0.9. Normally, ϵ is much smaller, but we thinned the data with a large ϵ to cut down on the amount of output to be shown in this report. File FØR11.DAT (output from INPUT.F4), which incudes information about the evaluated sets, is given in Appendix D. Files FØR02.DAT, FØR03.DAT, FØR04.DAT, and FØR10.DAT, output from the first run of INPUT.F4, are included in Appendix C. Files FØR02.DAT, FØR03.DAT, and FØR04.DAT are then used as input for GLUCS.F4. The file FØR01.DAT shown in Appendix C (sample input-first run) is the measured data typed by the user and read by GLUCS.F4. The first experimental set given in FØR01.DAT for the first run represents an absolute measurement for ^{56}Fe(n,p) while the second set represents the ratios of the cross sections ^{65}Cu(n,2n)/^{56}Fe(n,p). The data given at the end of FØR01.DAT are the lower triangular elements of the correlation matrix. These two sets are correlated due to common components in the measurements. Refer to Section IV where instructions are given to type this file.

Since there are no relative measurements used for the first run, GLUCS.F4 does not prompt the user for any information. File FØR14.DAT (output from GLUCS.F4), which includes the printout of the measured data, the evaluated energies and cross sections, the theoretical cross sections (or ratios), and the non-zero sensitivity matrix elements, is given in Appendix D. Files FØR15.DAT and FØR16.DAT (given in Appendix C) are output by GLUCS.F4 and they, along with FØR02.DAT and FØR10.DAT, are read by ØUTPUT.F4 during execution.

A portion of the output from ØUTPUT.F4 for the first run is given in Appendix D following FØR11.DAT and FØR14.DAT. FØR17.DAT includes the final thinned adjusted data for the two sets in readable form, while FØR18.DAT and FØR19.DAT contain the adjusted data for ^{56}Fe(n,p) and ^{65}Cu(n,2n), respectively, in ENDF/B-V format. We do not include FØR19.DAT in Appendix D in order to conserve space.

After execution of the programs for the first cycle of the problem, the temporary file FØR12.DAT (see Chapter III) was deleted. Also, files FØR01.DAT, FØR02.DAT, FØR15.DAT, FØR16.DAT, FØR17.DAT, FØR18.DAT, and FØR19.DAT are all renamed using extension .ØNE so that the second cycle of the problem can be run. (See teletype output in Appendix B). The new ENDF/B-V formatted files FØR18.ONE (^{56}Fe(n,p)) and FØR19.ONE (^{65}Cu(n,2n)) are then used in the second run for the problem. Also, the evaluated data for ^{32}S(n,p) (in file S32.DAT) is used for this second run, so that there are three evaluated sets. Note that FØR01.DAT for

the second run (Appendix C) includes different measured data. Initially, these data were typed by the user into file FØRO1.TWØ. After execution of the first run, FØRO1.TWØ is renamed to FØRO1.DAT. This procedure is included in the teletype output of Appendix B.

When executing INPUT.F4 the second time, the cross-reaction covariance data for $^{56}Fe(n,p)$ and $^{65}Cu(n,2n)$ is available from file FØR18.ONE. Therefore, the user tells the program to read these data; see the teletype output of the second run in Appendix B. Again, the data from the third set were thinned with $\epsilon = 0.9$ in order to cut down on the amount of output. (The data in the first two sets were already thinned in the first run.) As for the first run, files FØRO2.DAT, FØRO3.DAT, FØRO4.DAT, and FØR10.DAT are output from INPUT.F4 during the second run. These files are not included in this report. However, we do include file FØR11.DAT (output from INPUT.F4) in Appendix D. Also, the FØRO1.DAT measured data file for the second run is given in Appendix C. Four measured sets are used this time and are correlated due to common components in the measurements. The first experimental set given is the ratios of the cross sections of $^{56}Fe(n,p)/^{32}S(n,p)$. The second set is the ratios of the cross sections $^{65}Cu(n,2n)/^{32}S(n,p)$. The third measured set contains ratios of the $^{56}Fe(n,p)$ cross sections to the cross section of the same reaction at the fixed energy value of 14.5 MeV, while the fourth set includes similar ratios for $^{65}Cu(n,2n)$. The data given at the end of this second FØRO1.DAT file are the lower triangular elements of the relative covariance matrix (not correlation matrix). Again, see Section IV for specific instructions on how to type this file.

Since there are relative measurements for the second case, GLUCS.F4 prompts the user to type the fixed energy values. Refer to the teletype output in Appendix B for the prompting message and the user response. Again, FØR14.DAT from the second run of GLUCS.F4 is included in Appendix D. Files FØR15.DAT and FØR16.DAT that were output from GLUCS.F4 during the second run are not included in this report.

The problem is completed by executing ØUTPUT.F4, which reads the files FØR02.DAT, FØR10.DAT, FØR15.DAT, and FØR16.DAT that were output from the second runs of INPUT.F4 and GLUCS.F4. This time, three adjusted data sets (FØR18.DAT, FØR19.DAT, and FØR20.DAT) are output from ØUTPUT.F4. Only FØR18.DAT (^{56}Fe(n,p)) is included in Appendix D. Also, FØR17.DAT, which contains the final thinned adjusted data for ^{56}Fe(n,p), ^{65}Cu(n,2n), and ^{32}S(n,p) in readable form, is given in Appendix D.

VI. SUMMARY

The programs INPUT.F4, GLUCS.F4, and ØUTPUT.F4 have been used to solve numerous smaller sample problems that could be done with a calculator. All the different data types that the programs could handle were tried, and correct results obtained. The sample problem given in this report represents a real problem with the exception that the ϵ used (see Section V) would be too large. The results of this problem (with ϵ = 0.0) have been documented.[4]

REFERENCES

1. J. B. Dragt, J. W. M. Dekker, H. Gruppelaar, and A. J. Janssen, "Methods of Adjustment and Error Evaluation of Neutron Capture Cross Sections; Application to Fission Product Nuclides," Nuclear Sci. Eng. 62, 117 (1977).

2. F. G. Perey, "Spectrum Unfolding by the Least-Squares Method," Proceedings IAEA Technical Committee Meeting on Current Status of Neutron Spectrum Unfolding, Oak Ridge, Tennessee, October 10-12, 1977.

3. D. Garber, C. Dunford, and S. Pearlstein, "Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF," ENDF-102, BNL-NCS-50496 (1975).

4. C. Y. Fu, D. M. Hetrick, and F. G. Perey, "Simultaneous Evaluation of $^{32}S(n,p)$, $^{56}Fe(n,p)$, and $^{65}Cu(n,2n)$ Cross Sections," Proceedings Int. Conf. on Nuclear Cross Sections for Technology, Knoxville, Tennessee, October 22-26, 1979.

5. F. G. Perey, "The Data Covariance Files for ENDF/B-V," ORNL/TM-5938, ENDF-249 (1977).

6. N. M. Larson, Private Communication, to be published as an ORNL Report.

7. J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, "Linpack Users Guide," SIAM, 1979.

Appendix A

Program Listings

```
C THIS PROGRAM (INPUT.F4) IS A UTILITY PROGRAM THAT SETS UP THE INPUT DATA FOR
C GLUCS.F4 (SOLVES BAYES THEOREM) AND OUTPUT.F4 (WRITES OUT NEW DATA IN
C ENDF/B FORMAT). IT IS AN INTERACTIVE PROGRAM. DATA ARE READ FROM ENDF/B
C (UNIT 22 USED; MF=3 AND MF=33 ARE FILE #'S) AND INFORMATION IS OUTPUT
C IN FILES WITH UNIT #'S AS FOLLOWS:
C
C IWN (=2; IE FOR02.DAT): INFORMATION WRITTEN IN THIS FILE INCLUDES-
C    NSETF= TOTAL (FINAL) # OF PAIRS (ENERGIES, X-SECTIONS) FROM ALL
C           EVALUATED SETS,
C    NEVSET= # OF EVALUATED SETS,
C    NLGSET= # OF PAIRS IN LARGEST EVALUATED SET.
C
C IWE (=3): INFORMATION WRITTEN IN THIS FILE INCLUDES (FOR EACH EVALUATED
C           SET NEV) -
C    NPSET= # OF PAIRS (ENERGIES, X-SECTIONS) IN EACH SET,
C    NEVP(NEV)= EG, IF NSETF=16, NEVSET=3, AND THERE ARE 4 PAIRS IN EVALUATED
C               SET 1 AND 6 PAIRS IN SET 2, THEN NEVP(1)=4, NEVP(2)=10,
C               NEVP(3)=16,
C    NR= # OF INTERPOLATION RANGES IN EACH EVALUATED SET,
C    NBT(I)= # VALUE SEPARATING THE I-TH AND (I+1)-TH INTERPOLATION
C            RANGES,
C    INTP(I)= THE INTERPOLATION SCHEME IDENTIFICATION # USED IN THE
C             I-TH RANGE,
C    ESET(I),I=1,NPSET= ENERGIES FOR SET NEV
C    XSSET(I),I=1,NPSET= X-SECTIONS FOR SET NEV.
C
C IWV (=4): INFORMATION WRITTEN IN THIS FILE INCLUDES -
C    VAREVD(J)= COVARIANCE (MATRIX M) OF THE EVALUATED DATA.
C
C IT IS NOTED HERE THAT (TO SAVE STORAGE) THE COVARIANCE MATRIX M IS CALCULATED
C IN BLOCKS; EG EVALUATED SET 1 IS BLOCK (1,1), SET 2 IS BLOCK (2,2), ETC.
C OFF-DIAGONAL (RECTANGULAR) BLOCKS ARE CALCULATED USING THE LB = 3 TYPE
C STRUCTURE OF COVARIANCE FILES IN ENDF/B (FILE 33). COLUMNS OF BLOCKS
C AT ONE TIME (1ST COLUMN IS (1,1), (2,1), (3,1), ETC) ARE STORED IN ARRAY
C VAREVD AND THE TRANSPOSE IS WRITTEN IN FILE IWV. THE SECOND COLUMN OF
C BLOCKS WOULD BE (1,2), (2,2), (3,2), ETC. THE PROGRAM RUNS UNTIL THE TOTAL
C MATRIX M IS CALCULATED & WRITTEN.
C
C IWM (=10): INFORMATION WRITTEN IN THIS FILE INCLUDES (FOR EACH SET -
C            THESE DATA ARE FROM THE ENDF/B FILES)-
C    MAT= MATERIAL #,
C    MT= REACTION TYPE IDENTIFIER,
C    Q= REACTION Q-VALUE,
C    ZA= THE (Z,A) DESIGNATION FOR A MATERIAL,
C    AWR= THE RATIO OF THE MASS OF AN ATOM (OR MOLECULE) TO THAT OF
C         THE NEUTRON.
C
C IWP (=11): INFORMATION IN THIS FILE INCLUDES THE EVALUATED DATA OF EACH SET,
C            PRINTED IN READABLE FORM FOR THE USERS CONVENIENCE. SEE
C            SUBROUTINES PRNT & PRNTRT BELOW FOR MORE INFORMATION.
C
C IWT (=12): THIS FILE IS A TEMPORARY FILE THAT THE PROGRAM USES TO WRITE
C            THE COVARIANCE OF THE DIAGONAL BLOCKS OF MATRIX M. LATER, THE
C            PROGRAM READS THIS FILE IN ORDER TO WRITE THE FINAL COVARIANCE
```

```
C            INTO FILE IWV. AFTER EXECUTION OF INPUT.F4, THIS FILE MAY BE DELETED.
C
C IWB (=13): THIS FILE IS A TEMPORARY FILE THAT THE PROGRAM USES TO WRITE
C            THE COVARIANCE OF THE OFF-DIAGONAL BLOCKS OF MATRIX M. LATER,
C            THE PROGRAM READS THIS FILE IN ORDER TO WRITE THE FINAL COVARIANCE
C            INTO FILE IWV. AFTER EXECUTION OF INPUT.F4, THIS FILE MAY BE DELETED.
C
C IF THE DIMENSIONS OF THIS PROGRAM NEED TO BE INCREASED THE PROGRAM
C WILL TELL YOU SO BY STOPPING AND WRITING AN ERROR MESSAGE. DEFINITIONS
C OF OTHER IMPORTANT PARAMETERS APPEAR IN COMMENT STATEMENTS THROUGHOUT
C THE PROGRAM.
      DIMENSION INTP(3),ESET(75),XSSET(75)
      DIMENSION E(500),XSTN(500),VAREVD(37500)
      DIMENSION EKL(75),F(1332),S(75,75),NEVP(10),IXREAC(45)
C 1332 IN F ARRAY IS POSSIBLE DIMENSION NEEDED FOR X-REACTION COVARIANCE
C IE 36*37.
      COMMON/NUMBER/NPSET
      COMMON/INTERP/NR,NBT(3)
      COMMON/STAND/IXRN,STDEV(500)
      COMMON/RECPAR/NER,NEVC,NCIN,NCF,NRIN,NRF
      DOUBLE PRECISION IFILE
      IRD=22
      IWN=2
      IWE=3
      IWV=4
      IWM=10
      IWP=11
      IWT=12
      IWB=13
 3    TYPE 5
 5    FORMAT(1X,'  EVAL SETS = (0.LT. .LE.10) ',$)
      ACCEPT 10,NEVSET
 10   FORMAT(I)
      IF(NEVSET.GT.0.AND.NEVSET.LE.10)GO TO 12
      TYPE 11,NEVSET
 11   FORMAT(1X,'ERROR, NEVSET = ',I,' TRY AGAIN')
      GO TO 3
 12   NSETF=0
      TYPE 13
 13   FORMAT(1X,'IF WILL BE READING CROSS-REACTION COVARIANCE FILE(S)
     $, TYPE 1 ',$)
      ACCEPT 10,IXRN
      DO 270 NEV=1,NEVSET
      TYPE 15,NEV
 15   FORMAT(1X,'INPUT FILE ',I2,' = ',$)
      ACCEPT 20,IFILE
 20   FORMAT(A10)
      OPEN(UNIT=IRD,FILE=IFILE)
      TYPE 25
 25   FORMAT(1X,'MT = ',$)
      ACCEPT 10,MTT
      MFF=3
C READ FILE 3 INFORMATION (ENERGIES, X-SECTIONS)
 30   READ(IRD,35)MAT,MF,MT
```

```
 35     FORMAT(66X,I4,I2,I3)
        IF(MF.NE.MFF.OR.MT.NE.MTT)GO TO 30
        BACKSPACE IRD
        READ(IRD,40)ZA,AWR
 40     FORMAT(2E11.4)
        READ(IRD,45)Q,NR,NPSET
 45     FORMAT(11X,E11.4,22X,2I11)
        IF(NR.LE.3.AND.NPSET.LE.75)GO TO 48
        TYPE 46,NR,NPSET
 46     FORMAT(1X,'STOP - INCREASE DIMENSIONS: NR = ',I1,' NPSET = ',I3)
        STOP
 48     READ(IRD,50)(NBT(I),INTP(I),I=1,NR)
 50     FORMAT((6I11))
        READ(IRD,55)(ESET(I),XSSET(I),I=1,NPSET)
 55     FORMAT(6E11.4)
        MFF=33
C READ FILE 33 INFORMATION (COVARIANCE)
 60     READ(IRD,35)MAT,MF,MT
        IF(MF.NE.MFF.OR.MT.NE.MTT)GO TO 60
        BACKSPACE IRD
        READ(IRD,65)NL,MF,MT
C NL DENOTES # OF SUBSECTIONS WITHIN A SECTION
 65     FORMAT(55X,I11,4X,I2,I3)
 68     READ(IRD,70,END=72)AMAT1,AMT1,ANC,ANI
 70     FORMAT(22X,4F11.0)
        IF(AMAT1.EQ.0.0.AND.AMT1.EQ.FLOAT(MTT).AND.ANC.EQ.0.0)GO TO 74
        GO TO 68
 72     TYPE 73
 73     FORMAT(1X,'INCORRECT DATA (MF=33) - STOP')
        STOP
 74     NI=IFIX(ANI)
C NI IS THE # OF NI TYPE SUB-SUBSECTIONS (SEE ORNL/TM-5938)
        NVSET=NPSET*(NPSET+1)/2
        DO 80 I=1,NVSET
 80     VAREVD(I)=0.0
        DO 190 L=1,NI
        READ(IRD,75)LS,LB,NT,NP
 75     FORMAT(22X,4I11)
C LB IS A FLAG THAT DETERMINES MEANING OF #'S IN ARRAYS EKL & F BELOW.
C NP IS # OF PAIRS OF #'S IN THESE ARRAYS, AND NT & LS ARE DEFINED
C UNDER LB=5 SECTION BELOW.
        IF(NT.LE.1332.AND.NP.LE.75)GO TO 84
        TYPE 82,NT,NP
 82     FORMAT(1X,'STOP - INCREASE DIMENSIONS: NT = ',I3,' NP = ',I3)
        STOP
 84     IF(LB.EQ.5)GO TO 140
C LB MUST BE EITHER 0 OR 1 HERE.
        IF(LB.EQ.0.OR.LB.EQ.1)GO TO 88
        TYPE 86,LB
 86     FORMAT(1X,'LB = ',I3,' PROGRAM CANNOT HANDLE - STOP')
        STOP
 88     READ(IRD,55)(EKL(K),F(K),K=1,NP)
C EKL(K) REFERS TO THE ENERGY AND F(K) IS A NUMBER ASSOCIATED WITH THE
C ENERGY INTERVAL BETWEEN EKL(K) AND EKL(K+1).
```

```
        DO 85 I=1,NPSET
        DO 85 K=1,NP
 85     S(K,I)=0.0
C S IS AN OPERATOR SET TO 0 OR 1 DEPENDING ON WHETHER ESET(I) IS IN
C THE INTERVAL EKL(K) TO EKL(K+1)
        K1=1
        NPM1=NP-1
        DO 100 I=1,NPSET
        DO 90  K=K1,NPM1
        KC=K
 90     IF(ESET(I).GE.EKL(K).AND.ESET(I).LT.EKL(K+1))GO TO 95
        K=NP
 95     S(K,I)=1.0
        K1=KC
 100    CONTINUE
        LB=LB+1
        GO TO (105,120),LB
C HERE, LB IS 0
 105    M=0
        DO 115 I=1,NPSET
        DO 115 J=1,I
        M=M+1
        DO 110 K=1,NP
 110    VAREVD(M)=VAREVD(M)+S(K,I)*S(K,J)*F(K)
 115    CONTINUE
        GO TO 190
C HERE, LB IS 1
 120    M=0
        DO 130 I=1,NPSET
        DO 130 J=1,I
        M=M+1
        DO 125 K=1,NP
 125    VAREVD(M)=VAREVD(M)+S(K,I)*S(K,J)*F(K)*XSSET(I)*XSSET(J)
 130    CONTINUE
        GO TO 190
C HERE, LB IS 5
C LS = 1 SIGNIFIES A SYMMETRIC MATRIX. NT = TOTAL NUMBER OF ENTRIES
C IN THE TWO ARRAYS EKL AND F.
 140    IF(LS.EQ.1)GO TO 148
        TYPE 145,LS
 145    FORMAT(1X,'LS = ',I2,' MATRIX NOT SYMMETRIC - STOP')
        STOP
 148    NT=NT-NP
        READ(IRD,55)(EKL(K),K=1,NP),(F(K),K=1,NT)
        M=0
        NPM1=NP-1
        DO 185 I=1,NPSET
        DO 150 K=1,NPM1
 150    IF(ESET(I).GE.EKL(K).AND.ESET(I).LT.EKL(K+1))GO TO 155
        K=NPM1
 155    DO 185 J=1,I
        M=M+1
        DO 160 KP=1,NPM1
 160    IF(ESET(J).GE.EKL(KP).AND.ESET(J).LT.EKL(KP+1))GO TO 165
```

```
      KP=NPM1
165   ICOUNT=0
      DO 170 II=1,NPM1
      DO 170 JJ=II,NPM1
      ICOUNT=ICOUNT+1
170   IF(JJ.EQ.K.AND.II.EQ.KP)GO TO 180
      TYPE 175
175   FORMAT(1X,'ERROR IN LOOP 170 - STOP')
      STOP
180   VAREVD(M)=VAREVD(M)+XSSET(I)*XSSET(J)*F(ICOUNT)
185   CONTINUE
190   CONTINUE
      TYPE 195,NEV
195   FORMAT(1X,'TO THIN # ',I2,' SET, TYPE 1 ',$)
      ACCEPT 10,ITHIN
      WRITE(IWP,200)NEV
200   FORMAT(//,1X,'EVAL. DATA OF # ',I2,' SET'/)
      CALL PRNT(ESET(1),XSSET(1),VAREVD(1),ITHIN)
      IF(ITHIN.EQ.1)CALL THNLIN(ESET(1),XSSET(1),VAREVD(1))
      NSETIN=NSETF+1
      NSETF=NSETF+NPSET
      NEVP(NEV)=NSETF
      WRITE(IWE,220)NPSET,NSETF,NR
      WRITE(IWE,220)(NBT(I),INTP(I),I=1,NR)  .
220   FORMAT(1X,6I3)
      DO 225 I=1,NPSET
225   WRITE(IWE,230)ESET(I),XSSET(I)
230   FORMAT(1X,1PE11.4,1X,1PE11.4)
      WRITE(IWM,232)MAT,MT,Q,ZA,AWR
232   FORMAT(1X,I4,I3,1X,3(1PE11.4))
      IF(IXRN.NE.1)GO TO 240
      IF(NSETF.LE.500)GO TO 233
      TYPE 234,NSETF
234   FORMAT(1X,'STOP - INCREASE DIMENSIONS: NSETF = ',I3)
      STOP
C STORE ENERGIES AND CROSS SECTIONS OF EACH SET IN E & XSTN ARRAYS.
233   I=0
      DO 235 J=NSETIN,NSETF
      I=I+1
      E(J)=ESET(I)
235   XSTN(J)=XSSET(I)
240   CONTINUE
C WRITE OUT M MATRIX FOR SET NEV IN TEMPORARY FILE
      K=1
      DO 265 I=1,NPSET
      K=K+I-1
      L=K
      DO 245 LL=1,I
C STORE ROW OF VAREVD IN ESET
      ESET(LL)=VAREVD(L)
245   L=L+1
      IF((I+1).GT.NPSET)GO TO 255
      L=L+I-1
      DO 250 LL=I+1,NPSET
```

```
      ESET(LL)=VAREVD(L)
250   L=LL+L
255   WRITE(IWT,260)(ESET(J),J=1,NPSET)
260   FORMAT(1X,10(1PE11.4))
265   CONTINUE
270   CONTINUE
      NLGSET=NEVP(1)
      DO 274 I=2,NEVSET
      NST=NEVP(I)-NEVP(I-1)
274   NLGSET=MAXO(NLGSET,NST)
      WRITE(IWN,220)NSETF,NEVSET,NLGSET
      REWIND IWT
      IRD=0
      NEVCM1=0
      NCOL=NEVP(1)
      KOUNT=0
      ICOUNT=0
      IRECT=0
C NEVC REFERS TO COLUMNN OF THE BLOCKED MATRIX M.
      DO 420 NEVC=1,NEVSET
      M=0
      NEVCP1=NEVC+1
      IF(NEVCP1.GT.NEVSET)NEVCP1=NEVSET
      NROW=NEVP(1)
      NCF=NEVP(NEVC)
      IF(NEVC.EQ.1)GO TO 275
      NEVCM1=NEVP(NEVC-1)
      NCOL=NCF-NEVCM1
275   CONTINUE
C NEVR REFERS TO THE ROW OF THE BLOCKED MATRIX M.
      DO 410 NEVR=1,NEVCP1
      IF(NEVR.LT.NEVC)GO TO 385
      IF(NEVR.EQ.NEVC)GO TO 375
C THIS IS THE LOWER TRIANGULAR PART OF THE BLOCKED MATRIX M.
C SET RECTANGULAR BLOCKS (NEVR,NEVC) TO (NEVSET,NEVC) = 0.0 .
      MM=M
      NROWL=NSETF-NEVP(NEVR-1)
      NPTRCL=NROWL*NCOL
      DO 280 I=1,NPTRCL
      MM=MM+1
280   VAREVD(MM)=0.0
      IF(IXRN.NE.1)GO TO 410
      NCIN=NEVCM1+1
      KOUNT=KOUNT+NEVC
      ICOUNT=ICOUNT+1
      ICNT=ICOUNT
      KONT=KOUNT
      DO 370 NER=NEVR,NEVSET
      TYPE 285,NER,NEVC
285   FORMAT(1X,'IF READING CROSS-REACTION COVARIANCE FILE FOR SETS '
     >,I2,' AND ',I2,' TYPE 1 ',$)
      ACCEPT 10,IXREAC(KONT)
      IF(IXREAC(KONT).NE.1)GO TO 368
      NEVRM1=NEVP(NER-1)
```

```
      MIN=NCOL*NEVRM1+1
      NRF=NEVP(NER)
      MF=NRF*NCF
      NRIN=NEVRM1+1
      IRD=IRD+1
      TYPE 290
290   FORMAT(1X,'INPUT FILE (CROSS-REACTION COVARIANCE) = ',$)
      ACCEPT 20,IFILE
      OPEN(UNIT=IRD,FILE=IFILE)
      TYPE 295
295   FORMAT(1X,'TYPE MAT1,MT1 : ',$)
      ACCEPT 300,AMATL1,AMTN1
300   FORMAT(2F)
305   READ(IRD,310,END=72)AMAT1,AMT1
      IF(AMAT1.NE.AMATL1.OR.AMT1.NE.AMTN1)GO TO 305
310   FORMAT(22X,2F11.0)
      BACKSPACE IRD
      READ(IRD,75)MAT1,MT1,NC,NI
C IN FOLLOWING LOOP, PARAMETER DEFINITIONS DEPEND ON WHETHER LB=3 OR
C LB=6. IF LB=3, THERE ARE TWO SETS OF PAIRS OF NUMBERS WITH LT NUMBER OF PAIRS
C IN THE 2ND SET AND NR= TOTAL NUMBER OF PAIRS. IF LB=6, THEN THERE ARE
C NC (  COLUMNS) ENERGIES IN ONE SET AND NR (  ROWS) ENERGIES IN THE OTHER
C SET. THE OTHER PARAMETER DEFINITIONS ARE THE SAME AS ABOVE.
      DO 350 L=1,NI
      READ(IRD,75)LT,LB,NC,NR
      GO TO (311,311,313,311,311,316),LB
311   TYPE 312,LB
312   FORMAT(1X,'STOP: LB = ',I2,' (SHOULD BE 3 OR 6)')
      STOP
313   NP=NR
      IF(NP.LE.75)GO TO 315
      TYPE 314,NP
314   FORMAT(1X,'STOP - INCREASE DIMENSIONS: NP = ',I3)
      STOP
315   READ(IRD,55)(EKL(KL),F(KL),KL=1,NP)
      NPMLT=NP-LT
      NCP1=NPMLT+1
      NCM1=NPMLT-1
      GO TO 319
316   NP=NC+NR
      NCM1=NC-1
      NPF=NCM1*(NR-1)
      IF(NP.LE.75.AND.NPF.LE.1332)GO TO 318
      TYPE 317,NP,NPF
317   FORMAT(1X,'STOP, INCREASE DIMENSIONS: NP= ',I3,' NPF= ',I4)
      STOP
318   READ(IRD,55)(EKL(KL),KL=1,NP),(F(KL),KL=1,NPF)
      NCP1=NC+1
319   NPM1=NP-1
      M=MIN
      DO 345 J=NRIN,NRF
      MM=M
      DO 320 LK=NCP1,NPM1
320   IF(E(J).GE.EKL(LK).AND.E(J).LT.EKL(LK+1))GO TO 322
```

```
      LK=NPM1
322   IF(LB.EQ.3.AND.F(LK).EQ.0.0)GO TO 340
      DO 335 I=NCIN,NCF
      DO 325 KL=1,NCM1
325   IF(E(I).GE.EKL(KL).AND.E(I).LT.EKL(KL+1))GO TO 330
      KL=NCM1
330   IF(LB.EQ.3)FF=F(LK)*F(KL)
      IF(LB.EQ.6)FF=F((LK-NCP1)*NCM1+KL)
      VAREVD(MM)=VAREVD(MM)+FF*XSTN(I)*XSTN(J)
335   MM=MM+1
340   CONTINUE
      M=M+NCOL
345   CONTINUE
350   CONTINUE
      IF(IRD.EQ.1)GO TO 355
      DO 352 I=1,5000
352   READ(IWB,360,END=355)DUM
355   WRITE(IWB,360)NER,NEVC
360   FORMAT(1X,2I2)
      WRITE(IWB,365)(VAREVD(I),I=MIN,MF)
365   FORMAT(5X,10(1PE11.4))
C PRINT OUT COVARIANCE FOR RECTANGULAR BLOCK
      WRITE(IWP,366)NER,NEVC
366   FORMAT(//,1X,'CORRELATION MATRIX OF RECTANGULAR BLOCK ',I2,I2/)
      CALL PRNTRT(XSTN(NCIN),XSTN(NRIN),VAREVD(MIN))
368   KONT=KONT+ICNT
      ICNT=ICNT+1
370   CONTINUE
      REWIND IWB
      GO TO 410
375   M=M+1
C HERE NCOL=  OF COLUMNS =  OF ROWS IN A BLOCK OF DATA (DIAGONAL)
C OF MATRIX M.
      DO 380 I=1,NCOL
      MF=M-1+NCOL
      READ(IWT,260)(VAREVD(J),J=M,MF)
380   M=M+NCOL
      M=M-1
      GO TO 410
C HERE, WE ARE IN THE UPPER TRIANGLE PART OF THE BLOCK MATRIX M.
385   IRECT=IRECT+1
      IF(IXREAC(IRECT).NE.1)GO TO 395
      MF=M+NCOL
      M=M+1
      NPTRCU=NCOL*NEVP(NEVR)
390   READ(IWB,360)NCL,NRW
      IF(NEVR.NE.NRW.OR.NEVC.NE.NCL)GO TO 390
      READ(IWB,365)((VAREVD(J),J=I,NPTRCU,NCOL),I=M,MF)
      M=NPTRCU
      GO TO 410
395   IF(NEVR.NE.1)NROW=NEVP(NEVR)-NEVP(NEVR-1)
      NPTRC=NROW*NCOL
      DO 405 I=1,NPTRC
      M=M+1
```

```
 405     VAREVD(M)=0.0
 410     CONTINUE
C WRITE OUT (TRANSPOSE OF) COVARIANCE FOR COLUMN (BLOCK) NEVC OF MATRIX M
         NPTST=NCOL*NSETF
         DO 415 I=1,NCOL
 415     WRITE(IWV,260)(VAREVD(J),J=I,NPTST,NCOL)
 420     CONTINUE
         STOP
         END
C
         SUBROUTINE PRNTRT(XSC,XSR,VAR)
C THIS SUBROUTINE PRINTS OUT THE CORRELATION MATRIX OF THE RECTANGULAR
C OFF-DIAGONAL BLOCKS OF MATRIX M (ONLY IF ANY OF THE ELEMENTS ARE NON-
C ZERO). THE XSC ARRAY CONTAINS THE X-SECTION OF EACH COLUMN OF THE
C RECTANGULAR BLOCK & THE XSR ARRAY CONTAINS THE X-SECTION OF EACH ROW
C OF THE RECTANGULAR BLOCK. VAR CONTAINS THE COVARIANCE OF THE RECTANGULAR
C BLOCK, STDEV THE STANDARD DEVIATIONS.
         COMMON/STAND/IXRN,STDEV(500)
         COMMON/RECPAR/NER,NEVC,NCIN,NCF,NRIN,NRF
         DIMENSION XSC(1),XSR(1),VAR(1),ICOR(75)
         IWP=11
         IJ=0
         NROW=NRF-NRIN+1
         NCOL=NCF-NCIN+1
C NROW = # OF ROWS IN RECTANGULAR BLOCK & NCOL = # OF COLUMNS IN RECTANGULAR
C BLOCK. NRIN (NCIN) = # OF 1ST ROW (COLUMN) OF THE RECTANGULAR BLOCK
C IN MATRIX M & NRF (NCF) = # OF LAST ROW IN SAME BLOCK.
         WRITE(IWP,10)NEVC,(I,I=1,NCOL)
 10      FORMAT(2X,'SET',I2,28I4,1X/(19X,25I4))
         WRITE(IWP,20)NER
 20      FORMAT(1X,'SET',I2)
         NRINM1=NRIN-1
         NCINM1=NCIN-1
         DO 90 I=1,NROW
         STDR=STDEV(NRINM1+I)
         XSSTDR=XSR(I)*STDR
         IF(XSSTDR.EQ.0.0)GO TO 50
         DO 40 J=1,NCOL
         STDC=STDEV(NCINM1+J)
         IJ=IJ+1
         XSSTDC=XSC(J)*STDC
         IF(XSSTDC.EQ.0.0)GO TO 30
         ICOR(J)=IFIX(1.0E+6*VAR(IJ)/(XSSTDC*XSSTDR)+0.5)
         GO TO 40
 30      ICOR(J)=0.0
 40      CONTINUE
         GO TO 70
 50      IJ=IJ+NCOL
         DO 60 J=1,NCOL
 60      ICOR(J)=0.0
 70      WRITE(IWP,80)I,(ICOR(J),J=1,NCOL)
 80      FORMAT(1X,I3,4X,28I4,1X/(20X,25I4))
 90      CONTINUE
         RETURN
```

```
      END
C
      SUBROUTINE PRNT(E,XS,VAR,IJTHIN)
C THIS SUBROUTINE PRINTS OUT THE ENERGIES (E) & X-SECTIONS (XS) & STANDARD
C DEVIATIONS (STD) OF EACH EVALUATED SET ALONG WITH THE CORRESPONDING
C CORRELATION MATRIX (ICOR) CALCULATED FROM THE COVARIANCE (VAR) (DIAGONAL
C BLOCK OF MATRIX M). ALSO, THIS ROUTINE DETERMINES (IF IJTHIN = 1, IE IF
C THINNING THE DATA) WHICH ROWS (OR COLUMNS) OF THE CORRELATION MATRIX
C M (OF EACH SET) ARE CORRELATED TOGETHER. THE IJTHN ARRAY IS USED TO
C STORE THIS INFORMATION.
      COMMON/THIN/NBLK,IJTHN(75)
      COMMON/NUMBER/NPSET
      COMMON/STAND/IXRN,STDEV(500)
      DIMENSION E(1),XS(1),VAR(1),STD(75),ICOR(2850)
      DATA KOUNT/0/
      IWP=11
      WRITE(IWP,100)
 100  FORMAT(/,2X,'I',6X,'E(I)',6X,'X(E(I))',3X,'STD(X(E(I)))'/)
      NN=NPSET
      JJ=1
      DO 400 I=1,NN
      II=I*(I+1)/2
      IF(XS(I).EQ.0.0)GO TO 200
      STD(I)=SQRT(VAR(II))/XS(I)*100.
      GO TO 300
 200  STD(I)=0.0
 300  CONTINUE
 400  WRITE(IWP,500)I,E(I),XS(I),STD(I)
 500  FORMAT(1X,I3,2X,3(1PE11.4,1X))
      WRITE(IWP,600)
 600  FORMAT(/,5X,'CORRELATION MATRIX'/)
      NBLK=1
      IJTHN(1)=NN
      M=0
      N=1
C II = # OF ELEMENTS IN THE LOWER TRIANGULAR PART OF THE COVARIANCE MATRIX
C M OF THE CURRENT EVALUATED SET.
      DO 1000 I=1,II
      M=M+1
      IF(XS(N).EQ.0.0.OR.XS(M).EQ.0.0)GO TO 700
      IF(STD(M).EQ.0.0.OR.STD(N).EQ.0.0)GO TO 700
      ICOR(I)=IFIX(1.0E+6*VAR(I)/(XS(M)*STD(M)*XS(N)*STD(N))+0.5)
      GO TO 750
 700  ICOR(I)=0
 750  IF(IJTHIN.NE.1)GO TO 800
      IF((N-M).NE.1)GO TO 800
      IF(ICOR(I).EQ.100)GO TO 800
      IJTHN(JJ)=N-1
      JJ=JJ+1
      NBLK=JJ
      IJTHN(JJ)=NN
      GO TO 1000
 800  IF(N.NE.M)GO TO 1000
      N=N+1
```

```
      M=0
1000  CONTINUE
      ILMIN=0
      ILMAX=0
      DO 1100 I=1,NN
      ILMIN=ILMAX+1
      ILMAX=ILMAX+I
1100  WRITE(IWP,1200)I,(ICOR(L),L=ILMIN,ILMAX)
1200  FORMAT(1X,I3,1X,28I4,1X/(20X,25I4))
      IF(IXRN.NE.1)RETURN
      IF(IJTHIN.EQ.1)RETURN
C STORE STANDARD DEVIATIONS IN STDEV ARRAY FOR USE IN SUBROUTINE PRNTRT.
      DO 1300 I=1,NN
      KOUNT=KOUNT+1
1300  STDEV(KOUNT)=STD(I)
      RETURN
      END
C
      SUBROUTINE THNLIN(E,XS,VARNEW)
C THIS SUBROUTINE THINS THE ENERGIES & X-SECTIONS FROM ENDF FILE
C 3 AND THE COVARIANCE MATRIX M CALCULATED FROM ENDF FILE 33 WITH
C LINEAR INTERPOLATIONS (FOR THE CURRENT EVALUATED SET).
      COMMON/NUMBER/NPSET
      DIMENSION E(1),XS(1),VARNEW(1)
      DIMENSION NBTF(3),EN(75),XSN(75)
      COMMON/THIN/NBLK,IJTHN(75)
      COMMON/INTERP/NR,NBT(3)
      IWP=11
      TYPE 10
10    FORMAT(' EPS (0.<EPS<1.0) = '$)
      ACCEPT 20,EPS
20    FORMAT(E)
      EPSHI=1.+EPS
      EPSLO=1.-EPS
      NCOV=NPSET*(NPSET+1)/2.0
C NCOV = # OF ELEMENTS IN THE LOWER TRIANGULAR PART OF THE COVARIANCE MATRIX
C         OF THE CURRENT EVALUATED SET.
C NBLK = # OF CORRELATED BLOCKS OF MATRIX M.
      JPT=1
      IF(NBLK.EQ.NPSET)RETURN
      JA=0
      LCNT=0
      III=1
      DO 30 I=1,NR
30    NBTF(I)=NBT(I)
      DO 250 JBLK=1,NBLK
C JA & JB DETERMINE WHICH ROWS (OR COLUMNS) OF THE MATRIX ARE CORRELATED
C TOGETHER (IJTHN ARRAY IS FROM SUBROUTINE PRNT).
      IF(JBLK.NE.1)JA=IJTHN(JBLK-1)
      JB=IJTHN(JBLK)
      IF((JB-JA+1).GE.3)GO TO 50
      DO 40 L=JA+1,JB
      EN(JPT)=E(L)
      XSN(JPT)=XS(L)
```

```
 40     JPT=JPT+1
        III=JB
        J=JPT-1
        GO TO 250
 50     ISWT=0
        ISW=0
        NP=JB+1
        IF(JBLK.EQ.NBLK)NP=JB
        I=1+JA
        EN(JPT)=E(I)
        XSN(JPT)=XS(I)
        III=I
        J=JPT
 60     N=2
 70     K=0
 80     K=K+1
        IF((I+N).GT.NP)GO TO 230
C LINEAR INTERPOLATION IS DONE HERE.
        YC=(XS(I+N)-XSN(J))*(E(I+K)-EN(J))/(E(I+N)-EN(J))+XSN(J)
        IF(YC.LE.XS(I+K)*EPSHI.AND.YC.GT.XS(I+K)*EPSLO) GO TO 210
        J=J+1
        I=I+N-1
C ENERGIES & X-SECTIONS ARE THINNED HERE.
        EN(J)=E(I)
        XSN(J)=XS(I)
90      CONTINUE
        IF((I-III).LE.1)GO TO 150
        IM1=I-1
C THE FOLLOWING LOOP (140) DECREASES THE NUMBER VALUES SEPARATING THE
C INTERPOLATION RANGES (SINCE ENERGIES & X-SECTION PAIRS HAVE BEEN THINNED).
        DO 140 LL=III+1,IM1
        IF(LL.LE.NBT(1))GO TO 110
        IF(LL.LE.NBT(2))GO TO 120
        IF(LL.LE.NBT(3))GO TO 130
        TYPE 100
 100    FORMAT(1X,'MORE THAN 3 INTERPOLATION RANGES - STOP')
        STOP
 110    NBTF(1)=NBTF(1)-1
        IF(NR.LT.2)GO TO 140
 120    NBTF(2)=NBTF(2)-1
        IF(NR.LT.3)GO TO 140
 130    NBTF(3)=NBTF(3)-1
 140    CONTINUE
 150    III=I
        IF(ISWT.NE.0)GO TO 160
        IF(ISW.EQ.1)GO TO 240
        GO TO 60
 160    LCRIN=J+LCNT
        LCRF=I-1
C LCRIN & LCRF DENOTE # OF 1ST & LAST OF A STRING OF ROWS (COLUMNS) OF
C MATRIX M THAT WILL BE THINNED OUT.
        DO 200 L=LCRIN,LCRF
C 1ST, SET ROW (TO BE THINNED) OF COVARIANCE MATRIX TO 1.0E+5
        LCNT=LCNT+1
```

```
      LP1=L+1
      LM1=L-1
      IRIN=1+LM1*L/2
      IRF=L*LP1/2
      DO 170 M=IRIN,IRF
 170  VARNEW(M)=1.0E+5
C NOW, SET COLUMN (TO BE THINNED) TO 1.0E+5
      IROW=LP1
 180  IRIN=IRF+1
      IRF=IROW*(IROW+1)/2
      IF(IRF.GT.NCOV)GO TO 200
      IROW=IROW+1
      JCOL=0
      DO 190 M=IRIN,IRF
      JCOL=JCOL+1
      IF(JCOL.NE.L)GO TO 190
      VARNEW(M)=1.0E+5
      GO TO 180
 190  CONTINUE
 200  CONTINUE
      ISWT=0
      IF(ISW.EQ.1)GO TO 240
      GO TO 60
 210  IF(K.GE.N-1) GO TO 220
      ISWT=1
      GO TO 80
 220  N=N+1
      ISWT=1
      IF(I+N.LE.NP) GO TO 70
 230  J=J+1
      EN(J)=E(NP)
      XSN(J)=XS(NP)
      ISW=1
      IF(J.EQ.NP)GO TO 240
      IF(ISWT.EQ.0)GO TO 240
      I=NP
      GO TO 90
 240  CONTINUE
      JPT=J
 250  CONTINUE
C UPDATE NEW NUMBER VALUES FOR SEPARATING INTERPOLATION RANGES
      DO 260 LL=1,NR
 260  NBT(LL)=NBTF(LL)
C THIN 1.E+5 OUT OF VARNEW, IE THIN MATRIX M (VARNEW)
      ND=1
 270  DO 290 I=ND,NCOV
      II=I
      IF(VARNEW(I).NE.1.E+5)GO TO 290
      NCOVM1=NCOV-1
      IF((NCOVM1).LT.I)GO TO 310
      DO 280 L=I,NCOVM1
 280  VARNEW(L)=VARNEW(L+1)
      GO TO 300
 290  CONTINUE
```

```
      GO TO 320
300   NCOV=NCOVM1
      ND=II
      GO TO 270
310   NCOV=NCOVM1
320   CONTINUE
C CALL SUBROUTINE PRNT TO PRINT OUT THINNED DATA
      WRITE(IWP,330)
330   FORMAT(/,5X,'THINNED DATA'/)
      NPSET=J
      CALL PRNT(EN(1),XSN(1),VARNEW(1),0)
      DO 340 I=1,NPSET
      E(I)=EN(I)
340   XS(I)=XSN(I)
      RETURN
      END
```

```
C ** THIS PROGRAM (GLUCS.F4) EMPLOYES BAYES' THEOREM (OR EQUIVALENTLY
C ** LEAST SQUARES). EXPERIMENTAL DATA ARE READ FROM A FILE
C ** WITH UNIT NUMBER IRX (=1 IN THIS VERSION, IE FOR01.DAT) AND EVALUATED
C ** DATA ARE READ FROM FILES THAT WERE OUTPUT FROM THE UTILITY INPUT
C ** PROGRAM INPUT.F4 (UNIT NUMBERS IRN (=2), IRE (=3), AND IRV (=4)).
C ** IF THE DIMENSIONS OF THE PROGRAM NEED TO BE INCREASED FOR A PARTICULAR
C ** PROBLEM THE PROGRAM WILL STOP AND TYPE AN ERROR MESSAGE. THAT IS,
C ** THE USER IS TOLD TO INCREASE THE DIMENSIONS FOR EITHER THE REAL
C ** VARIABLES OR THE INTEGER VARIABLES. IN  OTHER WORDS, THE USER WILL
C ** NEED TO INCREASE THE DIMENSIONS OF ARRAY A AND INCREASE THE NUMBER
C ** NSIZER IN THE MAIN PROGRAM (REAL VARIABLES) OR INCREASE THE DIMENSION
C ** OF ARRAY IA AND INCREASE THE NUMBER NSIZEI IN THE MAIN PROGRAM (INTEGER
C ** VARIABLES). (THE DIMENSION OF ARRAY A AND THE NUMBER NSIZER MUST
C ** BE THE SAME NUMBER - LIKEWISE FOR ARRAY IA AND NSIZEI). IMPORTANT
C ** PARAMETERS THROUGHOUT THE PROGRAM ARE DEFINED.
C ** APRIL 21, 1980
C ** D. M. HETRICK
      COMMON/PARRL/NSIZER,A(35000)
      COMMON/PARIN/NSIZEI,IA(5000)
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      COMMON/WHERE/IEEX,IXSEX,ISTDEX,IVAREX,ICOREX,INEXP,IDEN,INUM,
     $  IEFIX,IEEV,IXSEV,INEVP,ITHY,IG,ITH,IGFIX,ITHFIX,IVAREV,IEMG,
     $  IXSNEW,IVARNW,IEN,IKPVT,IDUM
C ** NSIZER, NSIZEI BASED ON ? EXPERIMENTAL SETS, ? EVALUATED SETS
C ** WITH ? POINTS PER EXPRIMENTAL SET, ? POINTS PER EVALUATED SET
C ** (SEE USERS MANUAL)
      NSIZEI=5000
      NSIZER=35000
C ** GENERATE EXPERIMENTAL PARAMETERS (READ ENERGIES, X-SECTIONS, AND
C ** STANDARD DEVIATIONS, AND CALCULATE COVARIANCE)
      CALL PAREXP
C ** GENERATE EVALUATED PARAMETERS (READ ENERGIES, X-SECTIONS, AND
C ** CALCULATE G (PARTIAL DERIVATIVES), AND TH (THEORETICAL X-SECTION
C ** DATA))
      CALL PAREVL
C ** GENERATE COVARIANCE OF EVALUATED DATA
      CALL VAREVL
C ** GENERATE NEW X-SECTION XSNEW=T+MG(N+V)**-1(R-RT) AND NEW COVARIANCE VARNEW=
C ** M-M(G)TRANSPOSE(N+V)**-1(M(G)TRANSPOSE)TRANSPOSE  (AS DEFINED
C ** IN USERS MANUAL).
      CALL PARNEW
      STOP
      END
C
      FUNCTION IDIMR(MANY)
C ** PURPOSE - KEEP TRACK OF DIMENSIONS OF REAL ARRAYS
      COMMON/PARRL/NSIZER,A(1)
      DATA KOUNT/1/
      IDIMR=KOUNT
      KOUNT=KOUNT+MANY
      IF(KOUNT.LE.NSIZER)RETURN
      TYPE 10,NSIZER,KOUNT
 10   FORMAT(1X,'AVAILABLE SIZE (REAL) = ',I6,' BUT YOU NEED ',I6)
```

```
      STOP
      RETURN
      END
C
      FUNCTION IDIMI(MNY)
C ** PURPOSE - KEEP TRACK OF DIMENSIONS OF INTEGER ARRAYS
      COMMON/PARIN/NSIZEI,IA(1)
      DATA KONT/1/
      IDIMI=KONT
      KONT=KONT+MNY
      IF(KONT.LE.NSIZEI)RETURN
      TYPE 10,NSIZEI,KONT
 10   FORMAT(1X,'AVAILABLE SIZE (INTEGER) = ',I5,'BUT YOU NEED ',I5)
      STOP
      RETURN
      END
C
      SUBROUTINE PAREXP
C ** THIS ROUTINE SETS UP THE DIMENSIONS FOR THE EXPERIMENTAL VARIABLES
C ** AND CALLS SUBROUTINE PAREX TO READ AND GENERATE THE EXPERIMENTAL
C ** PARAMETERS.
      COMMON/PARRL/NSIZER,A(1)
      COMMON/PARIN/NSIZEI,IA(1)
      COMMON/WHERE/IEEX,IXSEX,ISTDEX,IVAREX,ICOREX,INEXP,IDEN,INUM,
     $  IEFIX,IEEV,IXSEV,INEVP,ITHY,IG,ITH,IGFIX,ITHFIX,IVAREV,IEMG,
     $  IXSNEW,IVARNW,IEN,IKPVT,IDUM
      COMMON 'NUMEX/NEXSET,NEXPTS,NCOVEX
      IRX=1
      READ(IRX,10)NEXPTS,NEXSET
C ** NEXSET= # OF EXPERIMENTAL SETS,
C ** NEXPTS= TOTAL # OF EXPERIMENTAL POINTS (ALL SETS).
 10   FORMAT(1X,2I3)
      IF(NEXPTS.GT.0.AND.NEXSET.GT.0)GO TO 30
      TYPE 20,NEXPTS,NEXSET
 20   FORMAT(1X,'STOP - NEXPTS = ',I,' NEXSET = ',I)
      STOP
 30   CONTINUE
C ** THE FOLLOWING PARAMETERS DETERMINE THE DIMENSIONS OF ARRAYS THAT ARE
C ** DEFINED IN SUBROUTINE PAREX. EG, IEEX IS THE DIMENSION SET ASIDE
C ** IN ARRAY A FOR  ARRAY EEX, THE EXPERIMENTAL ENERGY.
      IEEX=IDIMR(NEXPTS)
      IXSEX=IDIMR(NEXPTS)
      ISTDEX=IDIMR(NEXPTS)
      NCOVEX=NEXPTS*(NEXPTS+1)/2
C ** NCOVEX= # OF ELEMENTS IN LOWER TRIANGULAR COVARIANCE MATRIX OF
C ** EXPERIMENTAL DATA.
      IVAREX=IDIMR(NCOVEX)
      ICOREX=IDIMI(NCOVEX)
      INEXP=IDIMI(NEXSET)
      IDEN=IDIMI(NEXSET)
      INUM=IDIMI(NEXSET)
      IEFIX=IDIMR(NEXSET)
      CALL PAREX(A(IEEX),A(IXSEX),A(ISTDEX),A(IVAREX),IA(ICOREX),
     $          IA(INEXP),IA(IDEN),IA(INUM),A(IEFIX))
```

```
      RETURN
      END
C
      SUBROUTINE PAREX(EEX,XSEX,STDEX,VAREXD,ICOR,NEXP,ISTEVD,
     $          ISTEVN,EFIX)
C ** THIS SUBROUTINE READS THE EXPERIMENTAL DATA & FOR THE USERS CONVENIENCE
C ** PRINTS THESE DATA (UNIT # IWP (=14)) SO THAT IT CAN BE DETERMINED
C ** IF THEY WERE INPUT CORRECTLY.
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      DIMENSION EEX(1),XSEX(1),STDEX(1),VAREXD(1),EFIX(1)
      DIMENSION ICOR(1),NEXP(1),ISTEVD(1),ISTEVN(1)
      IRX=1
      IWP=14
      NINEX=1
      DO 60 NEX=1,NEXSET
      READ(IRX,10)NEXP(NEX),ISTEVN(NEX),ISTEVD(NEX)
C ** NEXP(NEX)= EG, IF NEXSET = 3, NEXPTS = 16, AND THERE ARE 4 POINTS IN
C ** EXPERIMENTAL SET 1 AND 6 POINTS IN EXPERIMENTAL SET 2, THEN
C ** NEXP(1)=4, NEXP(2)=10, AND NEXP(3)=16.
C ** ISTEVN(NEX) & ISTEVD(NEX) HANDLE THE RATIOS (NUMERATOR/DENOMINATOR);
C ** IE, ISTEVN(NEX) IS SET EQUAL TO THE # OF EVALUATED DATA SET THAT THE
C ** EXPERIMENTAL DATA SET NEX CORRESPONDS TO (NUMERATOR IN RATIO).
C ** ISTEVD(NEX) IS SET EQUAL TO THE # OF EVALUATED DATA SET THAT THE
C ** EXPERIMENTAL DATA SET NEX CORRESPONDS TO (DENOMINATOR IN RATIO).
C ** ISTEVD(NEX) IS SET TO 0 IF NO RATIOS INVOLVED IN EXPERIMENTAL DATA
C ** SET NEX. IF A FIXED SINGLE INPUT VALUE IS USED IN THE DENOMINATOR
C ** OF THE RATIO FOR EXPERIMENTAL SET NEX THEN ISTEVD(NEX) SHOULD
C ** BE NEGATIVE.
 10   FORMAT(1X,3I3)
      IF(NEXP(NEX).LE.NEXPTS.AND.NEXP(NEX).GE.NINEX)GO TO 18
      TYPE 14,NEXP(NEX),NINEX,NEXPTS
 14   FORMAT(1X,'ERROR IN DATA: NEXP(NEX) = ',I,' NINEX = ',I,
     $ ' NEXPTS = ',I)
      STOP
 18   DO 20 J=NINEX,NEXP(NEX)
      READ(IRX,30)EEX(J),XSEX(J),STDEX(J)
C ** EEX=EXPERIMENTAL ENERGY, XSEX=EXPERIMENTAL X-SECTION (OR RATIO),
C ** AND STDEX=EXPERIMENTAL STANDARD DEVIATION.
 20   EEX(J)=EEX(J)*1.0E+6
 30   FORMAT(1X,3(F10.4,1X))
      NINEX=NEXP(NEX)+1
      IF(ISTEVD(NEX).GE.0)GO TO 60
      IDEN=IABS(ISTEVD(NEX))
      TYPE 40,NEX,IDEN
 40   FORMAT(1X,'TYPE FIXED ENERGY (MEV)- (EXP. SET ',I2,' EVAL. SET  '
     $,I2,')',1X,$)
      ACCEPT 50,EFIX(NEX)
      EFIX(NEX)=EFIX(NEX)*1.0E+6
 50   FORMAT(F)
 60   CONTINUE
      IF(NEXP(NEXSET).EQ.NEXPTS)GO TO 68
      TYPE 64
 64   FORMAT(1X,'ERROR IN DATA: NEXP(NEXSET).NE.NEXPTS'
      STOP
```

```
 68     READ(IRX,10)IFLAG
C ** IF IFLAG = 0 READ CORRELATION MATRIX AND IF IFLAG = 1 READ
C ** RELATIVE COVARIANCE
        IF(IFLAG.EQ.1)GO TO 78
        READ(IRX,70)(ICOR(I),I=1,NCOVEX)
C ** ICOR = CORRELATION MATRIX (LOWER TRIANGULAR).
 70     FORMAT(20(I3,1X))
C ** THE FOLLOWING CALCULATES THE COVARIANCE OF THE EXPERIMENTAL DATA
C ** FROM THE CORRELATION MATRIX.
        J=0
        I=1
        DO 75 M=1,NCOVEX
        J=J+1
        VAREXD(M)=FLOAT(ICOR(M))*1.E-6*XSEX(I)*STDEX(I)*XSEX(J)*STDEX(J)
        IF(I.NE.J)GO TO 75
        I=I+1
        J=0
 75     CONTINUE
        GO TO 90
 78     READ(IRX,80)(VAREXD(I),I=1,NCOVEX)
C ** HERE, VAREXD IS THE RELATIVE COVARIANCE OF THE EXPERIMENTAL DATA
C ** (LOWER TRIANGULAR MATRIX).
 80     FORMAT(8F10.4)
C ** THE FOLLOWING CONVERTS VAREXD TO COVARIANCE & CALCULATES ICOR (THE
C ** CORRELATION MATRIX).
        J=0
        I=1
        DO 85 M=1,NCOVEX
        J=J+1
        ICOR(M)=IFIX((VAREXD(M)*100.0)/(STDEX(I)*STDEX(J))+0.5)
        VAREXD(M)=VAREXD(M)*1.E-4*XSEX(I)*XSEX(J)
        IF(I.NE.J)GO TO 85
        I=I+1
        J=0
 85     CONTINUE
 90     CONTINUE
C ** PRINT OUT EXPERIMENTAL DATA FOR USERS CONVENIENCE.
        WRITE(IWP,100)
 100    FORMAT(5X,'EXPERIMENTAL DATA'/)
        WRITE(IWP,110)
 110    FORMAT(2X,'I',6X,'E(I)',6X,'X(E(I))',3X,'STD(X(E( ,)')
        DO 120 I=1,NEXPTS
 120    WRITE(IWP,130)I,EEX(I),XSEX(I),STDEX(I)
 130    FORMAT(1X,I3,2X,3(1PE11.4,1X))
        WRITE(IWP,140)
 140    FORMAT(//,2X,'I',1X,'CORRELATION MATRIX'/)
        ICMIN=0
        ICMAX=0
        DO 150 I=1,NEXPTS
        ICMIN=ICMAX+1
        ICMAX=ICMAX+I
 150    WRITE(IWP,160)I,(ICOR(IC),IC=ICMIN,ICMAX)
 160    FORMAT(1X,I3,1X,28I4,1X/(20X,25I4))
        RETURN
```

```
      END
C
      SUBROUTINE PAREVL
C ** THIS ROUTINE SETS UP THE DIMENSIONS FOR THE EVALUATED DATA (READ
C ** FROM THE OUTPUT OF THE UTILITY PROGRAM INPUT.F4). SUBROUTINE PAREV
C ** IS CALLED TO READ THE EVALUATED ENERGIES AND X-SECTIONS AND GENERATE
C ** THE PARTIAL DERIVATIVES (G) AND THEORETICAL X-SECTION DATA (TH).
      COMMON/PARRL/NSIZER,A(1)
      COMMON/PARIN/NSIZEI,IA(1)
      COMMON/WHERE/IEEX,IXSEX,ISTDEX,IVAREX,ICOREX,INEXP,IDEN,INUM,
     $  IEFIX,IEEV,IXSEV,INEVP,ITHY,IG,ITH,IGFIX,ITHFIX,IVAREV,IEMG,
     $  IXSNEW,IVARNW,IEN,IKPVT,IDUM
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      IRN=2
      READ(IRN,10)NEVPTS,NEVSET,NLGEVS
C ** NEVPTS = TOTAL # OF EVALUATED POINTS (ALL SETS), NEVSET = # OF EVALUATED
C ** SETS, AND NLGEVS = # OF LARGEST EVALUATED SET.
 10   FORMAT(1X,6I3)
C ** THE FOLLOWING PARAMETERS DETERMINE THE DIMENSIONS OF ARRAYS THAT ARE
C ** DEFINED IN SUBROUTINE PAREV. EG, IEEV IS THE DIMENSION SET ASIDE
C ** IN ARRAY A FOR ARRAY EEV, THE EVALUATED ENERGIES.
      IEEV=IDIMR(NLGEVS)
      IXSEV=IDIMR(NLGEVS)
      INEVP=IDIMI(NEVSET)
      NXV=NEXPTS*NEVSET
      ITHY=IDIMR(NXV)
      NEXEVP=NEXPTS*NEVPTS
      IG=IDIMR(NEXEVP)
      ITH=IDIMR(NEXPTS)
      NEXLGE=NEXSET*NLGEVS
      IGFIX=IDIMR(NEXLGE)
      ITHFIX=IDIMR(NEXSET)
      CALL PAREV(A(IEEV),A(IXSEV),IA(INEVP),A(ITHY),A(IG),A(ITH),
     $          A(IEFIX),A(IGFIX),A(ITHFIX),IA(INUM),IA(IDEN),
     $          A(IEEX),A(IXSEX),IA(INEXP))
      RETURN
      END
C
      SUBROUTINE PAREV(EEV,XSEV,NEVP,THY,G,TH,EFIX,GFIX,THFIX,ISTEVN,
     $          ISTEVD,EEX,XSEX,NEXP)
C ** THIS SUBROUTINE READS THE FILE WITH UNIT # IRE (=3) THAT WAS OUTPUT
C ** FROM THE UTILITY PROGRAM INPUT.F4. (INTERPOLATION #'S, ENERGIES,
C ** AND X-SECTIONS FOR EACH SET). THE PARTIAL DERIVATIVES (G) & THE
C ** THEORETICAL X-SECTION DATA (TH) ARE CALCULATED (_PRINTED FOR THE
C ** USERS CONVENIENCE).
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      DIMENSION EEV(1),XSEV(1),THY(NEXPTS,1),G(NEXPTS,1),TH(1),
     $          EFIX(1),GFIX(NLGEVS,1),THFIX(1),EEX(1),XSEX(1)
      DIMENSION NEVP(1),ISTEVN(1),ISTEVD(1),NEXP(1)
      DIMENSION NBT(3),INTP(3)
      IWP=14
      IRE=3
```

```
      DO 10 J=1,NEVPTS
      DO 10 I=1,NEXPTS
 10   G(I,J)=0.0
      NFSET=0
      II=0
      WRITE(IWP,15)
 15   FORMAT(//,5X,'EVALUATED DATA'/)
      WRITE(IWP,18)
 18   FORMAT(2X,'I',6X,'E(I)',6X,'X(E(I))')
      DO 110 NEV=1,NEVSET
      READ(IRE,20)NPSET,NEVP(NEV),NR
C ** NPSET= # OF POINTS IN SET NEV,
C ** NEVP(NEV)= EG, IF NEVSET=3, NEVPTS=16, AND THERE ARE 4 POINTS IN
C ** EVALUATED SET 1 AND 6 POINTS IN EVALUATED SET 2, THEN NEVP(1)=4,
C ** NEVP(2)=10, AND NEVP(3)=16.
C ** NR=  NUMBER OF INTERPOLATION RANGES FOR SET NEV
      IF(NR.LE.3)GO TO 19
      TYPE 17,NR
 17   FORMAT(1X,'STOP - INCREASE DIMENSIONS: NR = ',I3)
      STOP
 19   READ(IRE,20)(NBT(I),INTP(I),I=1,NR)
C ** NBT(I)= NUMBER VALUE SEPARATING THE I-TH AND (I+1)-TH INTERPOLATION
C ** RANGES, AND INTP(I)= INTERPOLATION SCHEME IDENTIFICATION NUMBER
C ** USED IN THE I-TH RANGE.
 20   FORMAT(1X,6I3)
      DO 30 I=1,NPSET
      II=II+1
      READ(IRE,40)EEV(I),XSEV(I)
C ** EEV(I)= EVALUATED ENERGIES FOR SET NEV, AND XSEV(I)= EVALUATED X-SECTIONS.
 30   WRITE(IWP,35)II,EEV(I),XSEV(I)
 35   FORMAT(1X,I3,2X,3(1PE11.4,1X))
 40   FORMAT(1X,1PE11.4,1X,1PE11.4)
      NFT=NFSET+1
      NFSET=NFSET+NPSET
      NINEX=1
      DO 100 K=1,NEXSET
      IF(K.NE.1)NINEX=NEXP(K-1)+1
      NFEX=NEXP(K)
      INUM=ISTEVN(K)
      IDEN=IABS(ISTEVD(K))
      IF(INUM.NE.NEV.AND.IDEN.NE.NEV)GO TO 100
      IF(INUM.NE.IDEN)GO TO 60
C ** ISTEVD(K) SHOULD BE NEGATIVE HERE
C ** USED RELATIVE MEASUREMENT IN EXPERIMENT (IE, T(EI)/T(EF), EF FIXED
C ** ENERGY VALUE); CALCULATE FIXED VALUES FOR G AND TH HERE.
      EFX=EFIX(K)
      CALL GTHY(EFX,NPSET,EEV,XSEV,NBT,INTP,THFX,
     $          GFX1,GFX2,JJ)
      DO 50 J=1,NPSET
 50   GFIX(J,K)=0.0
      GFIX(JJ-1,K)=GFX1
      GFIX(JJ,K)=GFX2
      THFIX(K)=THFX
      GO TO 80
```

```
C **
 60     IF(INUM.EQ.NEV)GO TO 80
        IF(ISTEVD(K).GT.0)GO TO 80
C ** HAVE TP(EI)/T(EF), EF FIXED ENERGY VALUE; CALCULATE FIXED VALUES
C ** FOR G AND TH
        EFX=EFIX(K)
        CALL GTHY(EFX,NPSET,EEV,XSEV,NBT,INTP,THFX,
     $          GFX1,GFX2,JJ)
        JPNM2=JJ+NINSET-2
        JPNM1=JPNM2+1
        DO 70 I=NINEX,NFEX
        G(I,JPNM2)=GFX1
        G(I,JPNM1)=GFX2
 70     THY(I,NEV)=THFX
        GO TO 100
C ** HAVE TP(EI)/T(EI); CALCULATE REGULAR VALUES OF G AND TH
 80     DO 90 I=NINEX,NFEX
        ENEX=EEX(I)
        CALL GTHY(ENEX,NPSET,EEV,XSEV,NBT,INTP,THRY,
     $          G1,G2,JJ)
        JPNM2=JJ+NINSET-2
        JPNM1=JPNM2+1
        G(I,JPNM2)=G1
        G(I,JPNM1)=G2
 90     THY(I,NEV)=THRY
 100    CONTINUE
 110    CONTINUE
        REWIND IRE
C ** THE FOLLOWING CALCULATES FINAL VALUES FOR PARTIAL DERIVATIVES G
C ** AND THEORETICAL X-SECTION VALUES TH (ALL RATIOS ARE CONSIDERED).
        NINEX=1
        DO 220 K=1,NEXSET
        INUM=ISTEVN(K)
        IDEN=IABS(ISTEVD(K))
        IF(K.NE.1)NINEX=NEXP(K-1)+1
        NFEX=NEXP(K)
        IF(IDEN.EQ.0)GO TO 200
        NINEVD=1
        IF(IDEN.NE.1)NINEVD=NEVP(IDEN-1)+1
        NFEVD=NEVP(IDEN)
        IF(ISTEVD(K).LT.0.AND.IDEN.EQ.INUM)GO TO 180
        NINEVN=1
        IF(INUM.NE.1)NINEVN=NEVP(INUM-1)+1
        NFEVN=NEVP(INUM)
        DO 150 J=NINEVN,NFEVN
        DO 150 I=NINEX,NFEX
 150    G(I,J)=G(I,J)/THY(I,IDEN)
        DO 160 J=NINEVD,NFEVD
        DO 160 I=NINEX,NFEX
 160    G(I,J)=-THY(I,INUM)*G(I,J)/(THY(I,IDEN)*THY(I,IDEN))
        DO 170 I=NINEX,NFEX
 170    TH(I)=THY(I,INUM)/THY(I,IDEN)
        GO TO 220
 180    JJ=0
```

```
      THF=THFIX(K)
      THFSQ=THF*THF
      DO 190 J=NINEVD,NFEVD
      JJ=JJ+1
      GF=GFIX(JJ,K)
      DO 190 I=NINEX,NFEX
      G(I,J)=(G(I,J)/THF)-THY(I,INUM)*GF/THFSQ
190   TH(I)=THY(I,INUM)/THF
      GO TO 220
200   DO 210 I=NINEX,NFEX
210   TH(I)=THY(I,INUM)
220   CONTINUE
C ** PRINT OUT THEORETICAL X-SECTION DATA (TH) (COMPARE WITH EXPERIMENTAL
C ** DATA) AND PARTIAL DERIVATIVES (G) (NON-ZERO VALUES ONLY).
      WRITE(IWP,230)
230   FORMAT(//,5X,'THEORETICAL DATA'/)
      WRITE(IWP,240)
240   FORMAT(2X,'I',6X,'E(I)',4X,'REXP(E(I))',2X,'RTHEORY(E(I))')
      DO 250 I=1,NEXPTS
250   WRITE(IWP,35)I,EEX(I),XSEX(I),TH(I)
      WRITE(IWP,260)
260   FORMAT(//,5X,'SENSITIVITY MATRIX G-ONLY NONZERO VALUES GIVEN:'/)
      WRITE(IWP,270)
270   FORMAT(5X,'ROW I',5X,'COLUMN J',5X,'G(I,J)')
      DO 280 J=1,NEVPTS
      DO 280 I=1,NEXPTS
280   IF(G(I,J).NE.0.0)WRITE(IWP,290)I,J,G(I,J)
290   FORMAT(5X,I3,9X,I3,5X,1PE11.4)
      RETURN
      END
C
      SUBROUTINE GTHY(ENEX,NPSET,EEV,XSEV,NBT,INTP,THRY,G1,G2,JJ)
C ** THIS ROUTINE DETERMINES WHICH EVALUATED ENERGY INTERVAL (EEV ARRAY)
C ** THAT THE EXPERIMENTAL ENERGY ENEX LIES. THEN IT CALLS SUBROUTINE GTINT
C ** TO INTERPOLATE TO DETERMINE THE THEORETICAL VALUES OF G (I & (I+1)
C ** VALUES OF PARTIAL DERIVATIVES - CALLED G1, G2) AND THRY (THEORETICAL
C ** X-SECTION).
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      DIMENSION EEV(1),XSEV(1),NBT(1),INTP(1)
C ** ROUTINE ASSUMES THAT EXPERIMENTAL ENERGY ALWAYS .LE. EEV(NPSET)
      DO 10 J=2,NPSET
      JJ=J
      IF(ENEX.EQ.EEV(J))GO TO 50
      IF(ENEX.LT.EEV(J))GO TO 20
10    CONTINUE
      TYPE 15
15    FORMAT(1X,'ERROR - EXP. ENERGY.GT.EEV(NPSET)')
      STOP
20    XS1=XSEV(JJ-1)
      XS2=XSEV(JJ)
      EINT1=EEV(JJ-1)
      EINT2=EEV(JJ)
      DO 30 I=1,NR
30    IF(JJ.LE.NBT(I))INT=INTP(I)
```

```
      IF(JJ.LE.NBT(NR))GO TO 45
      TYPE 40
 40   FORMAT(1X,'ERROR - INTERPOLATION RANGE')
      STOP
C ** CALCULATE G1,G2, & THRY BY INTERPOLATION
 45   CALL GTINT(EINT1,EINT2,ENEX,XS1,XS2,INT,THRY,G1,G2)
      RETURN
C **
 50   G1=1.0
      G2=0.0
      THRY=XSEV(JJ)
      JJ=JJ+1
      RETURN
      END
C
      SUBROUTINE GTINT(EINT1,EINT2,ENEX,XS1,XS2,INT,THRY,G1,G2)
C ** THIS SUBROUTINE USES INTERPOLATION TO CALCULATE THRY, G1, G2. (SEE
C ** SUBROUTINE GTHY).
C ** X-SECTION CONSTANT IN ENERGY
      IF(INT.NE.1)GO TO 10
      THRY=XS1
      G1=1.0
      G2=0.0
 10   IF(INT.NE.2)GO TO 30
C ** X-SECTION LINEAR IN ENERGY
      THRY=XS1*EINT2-XS2*EINT1+ENEX*(XS2-XS1)
      E2ME1=EINT2-EINT1
      THRY=THRY/E2ME1
      DELTA=(ENEX-EINT1)/E2ME1
      G1=1.0-DELTA
      G2=DELTA
      RETURN
C **
30    IF(INT.NE.4)GO TO 40
C ** LN X-SECTION LINEAR IN ENERGY
      E2ME1=EINT2-EINT1
      THRY=XS1*EXP((ENEX-EINT1)*ALOG(XS2/XS1)/E2ME1)
      DELTA=(ENEX-EINT1)/E2ME1
      G1=THRY/XS1*(1.0-DELTA)
      G2=THRY/XS2*DELTA
      RETURN
C **
 40   IF(INT.NE.5)GO TO 50
C ** LN X-SECTION LINEAR IN LN ENERGY
      E2DE1=EINT2/EINT1
      THRY=XS1*EXP(ALOG(XS2/XS1)*ALOG(ENEX/EINT1)/ALOG(E2DE1))
      DELTA=ALOG(ENEX/EINT1)/ALOG(E2DE1)
      G1=THRY/XS1*(1.0-DELTA)
      G2=THRY/XS2*DELTA
      RETURN
C **
 50   TYPE 60,INT
 60   FORMAT(1X,'INTP = ',I1,' GTINT CALCULATES FOR INTP =
     $ 1,2,4, OR 5 ONLY')
```

```
      STOP
      RETURN
      END
C
      SUBROUTINE VAREVL
C ** THIS SUBROUTINE SETS UP THE DIMENSIONS FOR THE EVALUATED COVARIANCE (READ
C ** FROM THE OUTPUT OF UTILITY PROGRAM INPUT.F4 - UNIT # IRV (=4)).
C ** SUBROUTINE VAREV IS CALLED TO READ THE COVARIANCE VAREVD AND CALCULATE
C ** EMG = VAREVD*G(TRANSPOSE).
      COMMON/PARRL/NSIZER,A(1)
      COMMON/PARIN/NSIZEI,IA(1)
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      COMMON/WHERE/IEEX,IXSEX,ISTDEX,IVAREX,ICOREX,INEXP,IDEN,INUM,
     $  IEFIX,IEEV,IXSEV,INEVP,ITHY,IG,ITH,IGFIX,ITHFIX,IVAREV,IEMG,
     $  IXSNEW,IVARNW,IEN,IKPVT,IDUM
      NLGNEV=NLGEVS*NEVPTS
      IVAREV=IDIMR(NLGNEV)
      IEMG=IDIMR(NEXEVP)
      CALL VAREV(IA(INEVP),A(IG),A(IVAREV),A(IEMG))
      RETURN
      END
C
      SUBROUTINE VAREV(NEVP,G,VAREVD,EMG)
C ** VAREV READS THE COVARIANCE OF THE EVALUATED DATA (VAREVD - MATRIX M)
C ** FOR EACH SET FROM THE OUTPUT FILE (UNIT # IRV (=4)) OF THE UTILITY
C ** PROGRAM INPUT.F4. IT CALLS MULMG TO MULTIPLY VAREVD BY THE TRANSPOSE
C ** OF MATRIX G (PARTIAL DERIVATIVES).
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      DIMENSION NEVP(1),G(NEXPTS,1),VAREVD(NEVPTS,1),EMG(NEXPTS,1)
      IRV=4
      NPSET=NEVP(1)
      DO 40 NEV=1,NEVSET
      IF(NEV.NE.1)NPSET=NEVP(NEV)-NEVP(NEV-1)
C ** READ IN COVARIANCE EFFICIENTLY, IE (1,1), (2,1), (3,1) ETC.
      DO 20 J=1,NPSET
 20   READ(IRV,30)(VAREVD(I,J),I=1,NEVPTS)
 30   FORMAT(1X,10(1PE11.4))
C ** EMG=VAREVD*GT (GT IS G TRANSPOSE)
      CALL MULMG(NPSET,EMG,VAREVD,G)
 40   CONTINUE
      REWIND IRV
      RETURN
      END
C
      SUBROUTINE MULMG(NPSET,EMG,VAREVD,G)
C ** THIS SUBROUTINE MULTIPLIES COVARIANCE MATRIX M 'VAREVD) BY G TRANSPOSE
C ** AND STORES VALUES IN THE EMG ARRAY.
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      DIMENSION EMG(NEXPTS,1),VAREVD(NEVPTS,1),G(NEXPTS,1)
      DATA IROWI/0/
      NP=NPSET
      DO 20 IROW=1,NP
```

```
      IROWI=IROWI+1
      DO 20 JCOL=1,NEXPTS
      SUM=0.0
      DO 10 K=1,NEVPTS
 10   SUM=SUM+VAREVD(K,IROW)*G(JCOL,K)
 20   EMG(JCOL,IROWI)=SUM
      RETURN
      END
C
      SUBROUTINE PARNEW
C ** THIS SUBROUTINE SETS UP DIMENSIONS NEEDED TO CALCULATE THE NEW
C ** PARAMETERS (X-SECTIONS & COVARIANCE). SUBROUTINE PARNE IS CALLED TO
C ** PERFORM THESE CALCULATIONS.
      COMMON/PARRL/NSIZER,A(1)
      COMMON/PARIN/NSIZEI,IA(1)
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      COMMON/WHERE/IEEX,IXSEX,ISTDEX,IVAREX,ICOREX,INEXP,IDEN,INUM,
     $  IEFIX,IEEV,IXSEV,INEVP,ITHY,IG,ITH,IGFIX,ITHFIX,IVAREV,IEMG,
     $  IXSNEW,IVARNW,IEN,IKPVT,IDUM
C ** FOR XSNEW, USE SAME STORAGE THAT WAS USED FOR XSEV
      IXSNEW=IXSEV
C ** FOR VARNEW, USE SAME STORAGE THAT WAS USED FOR VAREVD
      IVARNW=IVAREV
      IEN=IDIMR(NCOVEX)
      N=NEXPTS
      IKPVT=IDIMI(N)
C ** FOR IDUM, USE SAME STORAGE THAT WAS USED FOR STDEX
      IDUM=ISTDEX
      CALL PARNE(IA(INEVP),A(IXSEX),A(IVAREX),A(ITH),A(IG),A(IEMG),
     $          A(IEEV),A(IXSNEW),A(IVARNW),A(IEN),IA(IKPVT),A(IDUM))
      RETURN
      END
C
      SUBROUTINE PARNE(NEVP,XSEX,VAREXD,TH,G,EMG,EEV,XSNEW,VARNEW,EN,
     $          KPVT,DUM)
C ** THIS ROUTINE MAKES SOME FINAL CALCULATIONS BEFORE CALLING SUBROUTINE
C ** XSVNEW TO CALCULATE THE NEW X-SECTIONS AND COVARIANCE MATRIX.
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      DIMENSION NEVP(1),XSEX(1),VAREXD(1),TH(1),G(NEXPTS,1),
     $          EMG(NEXPTS,1),EEV(1),XSNEW(1),VARNEW(NEVPTS,1),EN(1),
     $          KPVT(1),DUM(1)
C ** TRIANGULAR ARRAYS ARE USED WHERE POSSIBLE.
C **
C ** CALCULATE EN=G*EMG   (N=GM'G)TRANSPOSE)
      CALL MULGMG(EN,G,EMG)
C **
C ** CALCULATE EN=EN+VAREXD   (N+V)
      DO 10 I=1,NCOVEX
 10   EN(I)=EN(I)+VAREXD(I)
C **
C ** CALCULATE TH=XSEX-TH   (EXPERIMENTAL X-SECTION - THEORETICAL
C **                     X-SECTION OR (R-RT))
```

```
      DO 20 I=1,NEXPTS
 20   TH(I)=XSEX(I)-TH(I)
C **
C ** CALCULATE XSNEW=XSEV+EMG*(EN**-1)*TH AND VARNEW=VAREVD-EMG*(EN**-1)*EMGT
      CALL XSVNEW(NEVP,TH,EEV,XSNEW,VARNEW,EMG,EN,KPVT,DUM)
      RETURN
      END
C
      SUBROUTINE MULGMG(EN,G,EMG)
C ** MULGMG MULTIPLIES G BY M*G(TRANSPOSE) AND STORES RESULTS IN THE
C ** EN ARRAY (MATRIX N).
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      DIMENSION EN(1),G(NEXPTS,1),EMG(NEXPTS,1)
      DO 10 IL=1,NCOVEX
 10   EN(IL)=0.0
      DO 20 J=1,NEVPTS
      IL=0
      DO 20 I=1,NEXPTS
      DO 20 L=1,I
      IL=IL+1
 20   EN(IL)=EN(IL)+G(I,J)*EMG(L,J)
      RETURN
      END
C
      SUBROUTINE XSVNEW(NEVP,TH,EEV,XSNEW,VARNEW,EMG,EN,KPVT,DUM)
C ** THIS SUBROUTINE CALCULATES THE NEW X-SECTIONS (XSNEW ARRAY) & NEW
C ** COVARIANCE MATRIX M (VARNEW) AND WRITES THIS INFORMATION IN TWO FILES
C ** WITH UNIT #'S IWE (=15) & IWV (=16). THESE FILES WILL BE READ
C ** BY AN UTILITY OUTPUT PROGRAM (OUTPUT.F4) THAT WILL PUT THE DATA BACK
C ** INTO ENDF/B FORMAT.
      COMMON/NUMEX/NEXSET,NEXPTS,NCOVEX
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS,NEXEVP,NR
      DIMENSION NEVP(1),TH(1),EEV(1),XSNEW(1),VARNEW(NEVPTS,1),
     $          EMG(NEXPTS,1),EN(1),KPVT(1),DUM(1)
      DIMENSION NBT(3),INTP(3)
      IRE=3
      IRV=4
      IWE=15
      IWV=16
C **
C ** INVERSION IS HANDLED IN OPTIMUM FASHION.
C ** FACTORIZE EN
      CALL SSPCO(EN,NEXPTS,KPVT,RCOND,DUM)
      IF((1.0+RCOND).EQ.1.0)GO TO 80
C **
C ** CALCULATE EN**-1*TH   IE (N+V)**-1*(R-RT)
      CALL SSPSL(EN,NEXPTS,KPVT,TH)
C ** TH IS NOW (N+V)**-1*(R-RT)  (OR EN**-1*TH)
      II=0
      DO 20 NEV=1,NEVSET
      READ(IRE,22)NPSET,NEVPF,NR
      WRITE(IWE,22)NPSET,NEVPF,NR
      READ(IRE,22)(NBT(I),INTP(I),I=1,NR)
```

```
      WRITE(IWE,22)(NBT(I),INTP(I),I=1,NR)
      DO 15 I=1,NPSET
      II=II+1
C ** XSNEW HERE IS STILL SAME AS XSEV ABOVE.
      READ(IRE,24)EEV(I),XSNEW(I)
      DO 10 J=1,NEXPTS
 10   XSNEW(I)=XSNEW(I)+EMG(J,II)*TH(J)
C ** OUTPUT ENERGY & NEW X-SECTION (TO BE READ BY UTILITY OUTPUT PROGRAM).
 15   WRITE(IWE,24)EEV(I),XSNEW(I)
 20   CONTINUE
 22   FORMAT(1X,6I3)
 24   FORMAT(1X,1PE11.4,1X,1PE11.4)
 25   FORMAT(1X,10(1PE11.4))
      NINEV=1
      DO 70 NEV=1,NEVSET
      IF(NEV.NE.1)NINEV=NEVP(NEV-1)+1
      NFEV=NEVP(NEV)
      J=0
      DO 60 JJ=NINEV,NFEV
      DO 40 I=1,NEXPTS
 40   TH(I)=EMG(I,JJ)
C ** TH USED AS DUMMY  ARRAY HERE
      CALL SSPSL(EN,NEXPTS,KPVT,TH)
C ** TH = EN**-1*EMGT HERE   IE (N+V)**-1*(M(G)TRANSPOSE)TRANSPOSE
      J=J+1
      READ(IRV,25)(VARNEW(I,J),I=1,NEVPTS)
      DO 50 I=1,NEVPTS
      DO 50 K=1,NEXPTS
 50   VARNEW(I,J)=VARNEW(I,J)-EMG(K,I)*TH(K)
C ** OUTPUT NEW COVARIANCE (TO BE READ BY UTILITY OUTPUT PROGRAM).
      WRITE(IWV,25)(VARNEW(I,J),I=1,NEVPTS)
 60   CONTINUE
 70   CONTINUE
      RETURN
C
 80   TYPE 90,RCOND
 90   FORMAT(1X,'RCOND = ',1PE12.6/
     $ '  THE MATRIX IS ESSENTIALLY SINGULAR - STOP')
      STOP
      END
      SUBROUTINE SSPFA(AP,N,KPVT,INFO)
      INTEGER N,KPVT(1),INFO
      REAL AP(1)
C
C     SSPFA FACTORS A REAL SYMMETRIC MATRIX STORED IN
C     PACKED FORM BY ELIMINATION WITH SYMMETRIC PIVOTING.
C
C     TO SOLVE  A*X = B , FOLLOW SSPFA BY SSPSL.
C     TO COMPUTE  INVERSE(A)*C , FOLLOW SSPFA BY SSPSL.
C     TO COMPUTE  DETERMINANT(A) , FOLLOW SSPFA BY SSPDI.
C     TO COMPUTE  INVERSE(A) , FOLLOW SSPFA BY SSPDI.
C
C     ON ENTRY
C
```

```
C        AP       REAL (N*(N+1)/2)
C                 THE PACKED FORM OF A SYMMETRIC MATRIX  A .  THE
C                 COLUMNS OF THE UPPER TRIANGLE ARE STORED SEQUENTIALLY
C                 IN A ONE-DIMENSIONAL ARRAY OF LENGTH  N*(N+1)/2 .
C                 SEE COMMENTS BELOW FOR DETAILS.
C
C        N        INTEGER
C                 THE ORDER OF THE MATRIX  A .
C
C     OUTPUT
C
C        AP       A BLOCK DIAGONAL MATRIX AND THE MULTIPLIERS WHICH
C                 WERE USED TO OBTAIN IT STORED IN PACKED FORM.
C                 THE FACTORIZATION CAN BE WRITTEN  A = U*D*TRANS(U)
C                 WHERE  U  IS A PRODUCT OF PERMUTATION AND UNIT
C                 UPPER TRIANGULAR MATRICES , TRANS(U) IS THE
C                 TRANSPOSE OF  U , AND  D  IS BLOCK DIAGONAL
C                 WITH 1 BY 1 AND 2 BY 2 BLOCKS.
C
C        KPVT     INTEGER(N)
C                 AN INTEGER VECTOR OF PIVOT INDICES.
C
C        INFO     INTEGER
C                 = 0  NORMAL VALUE.
C                 = K  IF THE K-TH PIVOT BLOCK IS SINGULAR. THIS IS
C                      NOT AN ERROR CONDITION FOR THIS SUBROUTINE,
C                      BUT IT DOES INDICATE THAT SSPSL OR SSPDI MAY
C                      DIVIDE BY ZERO IF CALLED.
C
C     PACKED STORAGE
C
C          THE FOLLOWING PROGRAM SEGMENT WILL PACK THE UPPER
C          TRIANGLE OF A SYMMETRIC MATRIX.
C
C                K = 0
C                DO 20 J = 1, N
C                   DO 10 I = 1, J
C                      K = K + 1
C                      AP(K)  = A(I,J)
C             10    CONTINUE
C             20 CONTINUE
C
C     LINPACK. THIS VERSION DATED 03/11/78 .
C     JAMES BUNCH, UNIV. CALIF. SAN DIEGO, ARGONNE NAT. LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     BLAS SAXPY,SSWAP,ISAMAX
C     FORTRAN ABS,AMAX1,SQRT
C
C     INTERNAL VARIABLES
C
      REAL AK,AKM1,BK,BKM1,DENOM,MULK,MULKM1,T
      REAL ABSAKK,ALPHA,COLMAX,ROWMAX
```

```
      INTEGER ISAMAX,IJ,IJJ,IK,IKM1,IM,IMAX,IMAXP1,IMIM,IMJ,IMK
      INTEGER J,JJ,JK,JKM1,JMAX,JMIM,K,KK,KM1,KM1K,KM1KM1,KM2,KSTEP
      LOGICAL SWAP
C
C
C     INITIALIZE
C
C     ALPHA IS USED IN CHOOSING PIVOT BLOCK SIZE.
      ALPHA = (1.0E0 + SQRT(17.0E0))/8.0E0
C
      INFO = 0
C
C     MAIN LOOP ON K, WHICH GOES FROM N TO 1.
C
      K = N
      IK = (N*(N - 1))/2
   10 CONTINUE
C
C        LEAVE THE LOOP IF K=0 OR K=1.
C
C     ...EXIT
         IF (K .EQ. 0) GO TO 200
         IF (K .GT. 1) GO TO 20
            KPVT(1) = 1
            IF (AP(1) .EQ. 0.0E0) INFO = 1
C     ......EXIT
            GO TO 200
   20    CONTINUE
C
C        THIS SECTION OF CODE DETERMINES THE KIND OF
C        ELIMINATION TO BE PERFORMED.  WHEN IT IS COMPLETED,
C        KSTEP WILL BE SET TO THE SIZE OF THE PIVOT BLOCK, AND
C        SWAP WILL BE SET TO .TRUE. IF AN INTERCHANGE IS
C        REQUIRED.
C
         KM1 = K - 1
         KK = IK + K
         ABSAKK = ABS(AP(KK))
C
C        DETERMINE THE LARGEST OFF-DIAGONAL ELEMENT IN
C        COLUMN K.
C
         IMAX = ISAMAX(K-1,AP(IK+1),1)
         IMK = IK + IMAX
         COLMAX = ABS(AP(IMK))
         IF (ABSAKK .LT. ALPHA*COLMAX) GO TO 30
            KSTEP = 1
            SWAP = .FALSE.
         GO TO 90
   30    CONTINUE
C
C           DETERMINE THE LARGEST OFF-DIAGONAL ELEMENT IN
C           ROW IMAX.
C
```

```
            ROWMAX = 0.0E0
            IMAXP1 = IMAX + 1
            IM = IMAX*(IMAX - 1)/2
            IMJ = IM + 2*IMAX
            DO 40 J = IMAXP1, K
               ROWMAX = AMAX1(ROWMAX,ABS(AP(IMJ)))
               IMJ = IMJ + J
   40       CONTINUE
            IF (IMAX .EQ. 1) GO TO 50
               JMAX = ISAMAX(IMAX-1,AP(IM+1),1)
               JMIM = JMAX + IM
               ROWMAX = AMAX1(ROWMAX,ABS(AP(JMIM)))
   50       CONTINUE
            IMIM = IMAX + IM
            IF (ABS(AP(IMIM)) .LT. ALPHA*ROWMAX) GO TO 60
               KSTEP = 1
               SWAP = .TRUE.
            GO TO 80
   60       CONTINUE
            IF (ABSAKK .LT. ALPHA*COLMAX*(COLMAX/ROWMAX)) GO TO 70
               KSTEP = 1
               SWAP = .FALSE.
            GO TO 80
   70       CONTINUE
               KSTEP = 2
               SWAP = IMAX .NE. KM1
   80       CONTINUE
   90    CONTINUE
         IF (AMAX1(ABSAKK,COLMAX) .NE. 0.0E0) GO TO 100
C
C           COLUMN K IS ZERO.  SET INFO AND ITERATE THE LOOP.
C
            KPVT(K) = K
            INFO = K
         GO TO 190
  100    CONTINUE
         IF (KSTEP .EQ. 2) GO TO 140
C
C           1 X 1 PIVOT BLOCK.
C
            IF (.NOT.SWAP) GO TO 120
C
C              PERFORM AN INTERCHANGE.
C
               CALL SSWAP(IMAX,AP(IM+1),1,AP(IK+1),1)
               IMJ = IK + IMAX
               DO 110 JJ = IMAX, K
                  J = K + IMAX - JJ
                  JK = IK + J
                  T = AP(JK)
                  AP(JK) = AP(IMJ)
                  AP(IMJ) = T
                  IMJ = IMJ - (J - 1)
  110          CONTINUE
```

```
  120       CONTINUE
C
C           PERFORM THE ELIMINATION.
C
            IJ = IK - (K - 1)
            DO 130 JJ = 1, KM1
               J = K - JJ
               JK = IK + J
               MULK = -AP(JK)/AP(KK)
               T = MULK
               CALL SAXPY(J,T,AP(IK+1),1,AP(IJ+1),1)
               IJJ = IJ + J
               AP(JK) = MULK
               IJ = IJ - (J - 1)
  130       CONTINUE
C
C           SET THE PIVOT ARRAY.
C
            KPVT(K) = K
            IF (SWAP) KPVT(K) = IMAX
         GO TO 190
  140    CONTINUE
C
C           2 X 2 PIVOT BLOCK.
C
            KM1K = IK + K - 1
            IKM1 = IK - (K - 1)
            IF (.NOT.SWAP) GO TO 160
C
C              PERFORM AN INTERCHANGE.
C
               CALL SSWAP(IMAX,AP(IM+1),1,AP(IKM1+1),1)
               IMJ = IKM1 + IMAX
               DO 150 JJ = IMAX, KM1
                  J = KM1 + IMAX - JJ
                  JKM1 = IKM1 + J
                  T = AP(JKM1)
                  AP(JKM1) = AP(IMJ)
                  AP(IMJ) = T
                  IMJ = IMJ - (J - 1)
  150          CONTINUE
               T = AP(KM1K)
               AP(KM1K) = AP(IMK)
               AP(IMK) = T
  160       CONTINUE
C
C           PERFORM THE ELIMINATION.
C
            KM2 = K - 2
            IF (KM2 .EQ. 0) GO TO 180
               AK = AP(KK)/AP(KM1K)
               KM1KM1 = IKM1 + K - 1
               AKM1 = AP(KM1KM1)/AP(KM1K)
               DENOM = 1.0E0 - AK*AKM1
```

```
                  IJ = IK - (K - 1) - (K - 2)
                  DO 170 JJ = 1, KM2
                     J = KM1 - JJ
                     JK = IK + J
                     BK = AP(JK)/AP(KM1K)
                     JKM1 = IKM1 + J
                     BKM1 = AP(JKM1)/AP(KM1K)
                     MULK = (AKM1*BK - BKM1)/DENOM
                     MULKM1 = (AK*BKM1 - BK)/DENOM
                     T = MULK
                     CALL SAXPY(J,T,AP(IK+1),1,AP(IJ+1),1)
                     T = MULKM1
                     CALL SAXPY(J,T,AP(IKM1+1),1,AP(IJ+1),1)
                     AP(JK) = MULK
                     AP(JKM1) = MULKM1
                     IJJ = IJ + J
                     IJ = IJ - (J - 1)
  170             CONTINUE
  180          CONTINUE
C
C              SET THE PIVOT ARRAY.
C
               KPVT(K) = 1 - K
               IF (SWAP) KPVT(K) = -IMAX
               KPVT(K-1) = KPVT(K)
  190       CONTINUE
            IK = IK - (K - 1)
            IF (KSTEP .EQ. 2) IK = IK - (K - 2)
            K = K - KSTEP
         GO TO 10
  200 CONTINUE
      RETURN
      END
C
C
C
      INTEGER FUNCTION ISAMAX(N,SX,INCX)
C
C     FINDS THE INDEX OF ELEMENT HAVING MAX. ABSOLUTE VALUE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1),SMAX
      INTEGER I,INCX,IX,N
C
      ISAMAX = 0
      IF( N .LT. 1 ) RETURN
      ISAMAX = 1
      IF(N.EQ.1)RETURN
      IF(INCX.EQ.1)GO TO 20
C
C        CODE FOR INCREMENT NOT EQUAL TO 1
C
      IX = 1
      SMAX = ABS(SX(1))
```

```
      IX = IX + INCX
      DO 10 I = 2,N
         IF(ABS(SX(IX)).LE.SMAX) GO TO 5
         ISAMAX = I
         SMAX = ABS(SX(IX))
    5    IX = IX + INCX
   10 CONTINUE
      RETURN
C
C        CODE FOR INCREMENT EQUAL TO 1
C
   20 SMAX = ABS(SX(1))
      DO 30 I = 2,N
         IF(ABS(SX(I)).LE.SMAX) GO TO 30
         ISAMAX = I
         SMAX = ABS(SX(I))
   30 CONTINUE
      RETURN
      END
C
C
C
      REAL FUNCTION SASUM(N,SX,INCX)
C
C     TAKES THE SUM OF THE ABSOLUTE VALUES.
C     USES UNROLLED LOOPS FOR INCREMENT EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1),STEMP
      INTEGER I,INCX,M,MP1,N,NINCX
C
      SASUM = 0.0E0
      STEMP = 0.0E0
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1)GO TO 20
C
C        CODE FOR INCREMENT NOT EQUAL TO 1
C
      NINCX = N*INCX
      DO 10 I = 1,NINCX,INCX
        STEMP = STEMP + ABS(SX(I))
   10 CONTINUE
      SASUM = STEMP
      RETURN
C
C        CODE FOR INCREMENT EQUAL TO 1
C
C
C        CLEAN-UP LOOP
C
   20 M = MOD(N,6)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        STEMP = STEMP + ABS(SX(I))
```

```
   30 CONTINUE
      IF( N .LT. 6 ) GO TO 60
   40 MP1 = M + 1
      DO 50 I = MP1,N,6
        STEMP = STEMP + ABS(SX(I)) + ABS(SX(I + 1)) + ABS(SX(I + 2))
     *  + ABS(SX(I + 3)) + ABS(SX(I + 4)) + ABS(SX(I + 5))
   50 CONTINUE
   60 SASUM = STEMP
      RETURN
      END
C
C
C
      SUBROUTINE SAXPY(N,SA,SX,INCX,SY,INCY)
C
C     CONSTANT TIMES A VECTOR PLUS A VECTOR.
C     USES UNROLLED LOOP FOR INCREMENTS EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1),SY(1),SA
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
C
      IF(N.LE.0)RETURN
      IF (SA .EQ. 0.0) RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
C
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
C          NOT EQUAL TO 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        SY(IY) = SY(IY) + SA*SX(IX)
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      RETURN
C
C        CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C
C        CLEAN-UP LOOP
C
   20 M = MOD(N,4)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        SY(I) = SY(I) + SA*SX(I)
   30 CONTINUE
      IF( N .LT. 4 ) RETURN
   40 MP1 = M + 1
      DO 50 I = MP1,N,4
        SY(I) = SY(I) + SA*SX(I)
```

```
        SY(I + 1) = SY(I + 1) + SA*SX(I + 1)
        SY(I + 2) = SY(I + 2) + SA*SX(I + 2)
        SY(I + 3) = SY(I + 3) + SA*SX(I + 3)
   50 CONTINUE
      RETURN
      END
C
C
C
      REAL FUNCTION SDOT(N,SX,INCX,SY,INCY)
C
C     FORMS THE DOT PRODUCT OF TWO VECTORS.
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1),SY(1),STEMP
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
C
      STEMP = 0.0E0
      SDOT = 0.0E0
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
C
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
C          NOT EQUAL TO 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        STEMP = STEMP + SX(IX)*SY(IY)
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      SDOT = STEMP
      RETURN
C
C        CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C
C        CLEAN-UP LOOP
C
   20 M = MOD(N,5)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        STEMP = STEMP + SX(I)*SY(I)
   30 CONTINUE
      IF( N .LT. 5 ) GO TO 60
   40 MP1 = M + 1
      DO 50 I = MP1,N,5
        STEMP = STEMP + SX(I)*SY(I) + SX(I + 1)*SY(I + 1) +
     *   SX(I + 2)*SY(I + 2) + SX(I + 3)*SY(I + 3) + SX(I + 4)*SY(I + 4)
   50 CONTINUE
```

```
   60 SDOT = STEMP
      RETURN
      END
C
C
C
      SUBROUTINE  SSCAL(N,SA,SX,INCX)
C
C     SCALES A VECTOR BY A CONSTANT.
C     USES UNROLLED LOOPS FOR INCREMENT EQUAL TO 1.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SA,SX(1)
      INTEGER I,INCX,M,MP1,N,NINCX
C
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1)GO TO 20
C
C        CODE FOR INCREMENT NOT EQUAL TO 1
C
      NINCX = N*INCX
      DO 10 I = 1,NINCX,INCX
        SX(I) = SA*SX(I)
   10 CONTINUE
      RETURN
C
C        CODE FOR INCREMENT EQUAL TO 1
C
C
C        CLEAN-UP LOOP
C
   20 M = MOD(N,5)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        SX(I) = SA*SX(I)
   30 CONTINUE
      IF( N .LT. 5 ) RETURN
   40 MP1 = M + 1
      DO 50 I = MP1,N,5
        SX(I) = SA*SX(I)
        SX(I + 1) = SA*SX(I + 1)
        SX(I + 2) = SA*SX(I + 2)
        SX(I + 3) = SA*SX(I + 3)
        SX(I + 4) = SA*SX(I + 4)
   50 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE  SSWAP (N,SX,INCX,SY,INCY)
C
C     INTERCHANGES TWO VECTORS.
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO 1.
```

```
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1),SY(1),STEMP
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
C
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
C
C       CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS NOT EQUAL
C         TO 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        STEMP = SX(IX)
        SX(IX) = SY(IY)
        SY(IY) = STEMP
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      RETURN
C
C       CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C
C       CLEAN-UP LOOP
C
   20 M = MOD(N,3)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        STEMP = SX(I)
        SX(I) = SY(I)
        SY(I) = STEMP
   30 CONTINUE
      IF( N .LT. 3 ) RETURN
   40 MP1 = M + 1
      DO 50 I = MP1,N,3
        STEMP = SX(I)
        SX(I) = SY(I)
        SY(I) = STEMP
        STEMP = SX(I + 1)
        SX(I + 1) = SY(I + 1)
        SY(I + 1) = STEMP
        STEMP = SX(I + 2)
        SX(I + 2) = SY(I + 2)
        SY(I + 2) = STEMP
   50 CONTINUE
      RETURN
      END
C
C
C
```

```
      SUBROUTINE SSPSL(AP,N,KPVT,B)
      INTEGER N,KPVT(1)
      REAL AP(1),B(1)
C
C     SSISL SOLVES THE REAL SYMMETRIC SYSTEM
C     A * X = B
C     USING THE FACTORS COMPUTED BY SSPFA.
C
C     ON ENTRY
C
C        AP       REAL(N*(N+1)/2)
C                 THE OUTPUT FROM SSPFA.
C
C        N        INTEGER
C                 THE ORDER OF THE MATRIX  A .
C
C        KPVT     INTEGER(N)
C                 THE PIVOT VECTOR FROM SSPFA.
C
C        B        REAL(N)
C                 THE RIGHT HAND SIDE VECTOR.
C
C     ON RETURN
C
C        B        THE SOLUTION VECTOR  X .
C
C     ERROR CONDITION
C
C        A DIVISION BY ZERO MAY OCCUR IF  SSPCO  HAS SET RCOND .EQ. 0.0
C        OR  SSPFA  HAS SET INFO .NE. 0  .
C
C     TO COMPUTE  INVERSE(A) * C  WHERE  C  IS A MATRIX
C     WITH  P  COLUMNS
C           CALL SSPFA(AP,N,KPVT,INFO)
C           IF (INFO .NE. 0) GO TO ...
C           DO 10 J = 1, P
C              CALL SSPSL(AP,N,KPVT,C(1,J))
C        10 CONTINUE
C
C     LINPACK. THIS VERSION DATED 03/11/78 .
C     JAMES BUNCH, UNIV. CALIF. SAN DIEGO, ARGONNE NAT. LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     BLAS SAXPY,SDOT
C     FORTRAN IABS
C
C     INTERNAL VARIABLES.
C
      REAL AK,AKM1,BK,BKM1,SDOT,DENOM,TEMP
      INTEGER IK,IKM1,IKP1,K,KK,KM1K,KM1KM1,KP
C
C     LOOP BACKWARD APPLYING THE TRANSFORMATIONS AND
C     D INVERSE TO B.
```

```
C
      K = N
      IK = (N*(N - 1))/2
   10 IF (K .EQ. 0) GO TO 80
         KK = IK + K
         IF (KPVT(K) .LT. 0) GO TO 40
C
C           1 X 1 PIVOT BLOCK.
C
            IF (K .EQ. 1) GO TO 30
               KP = KPVT(K)
               IF (KP .EQ. K) GO TO 20
C
C                 INTERCHANGE.
C
                  TEMP = B(K)
                  B(K) = B(KP)
                  B(KP) = TEMP
   20          CONTINUE
C
C              APPLY THE TRANSFORMATION.
C
               CALL SAXPY(K-1,B(K),AP(IK+1),1,B(1),1)
   30       CONTINUE
C
C           APPLY D INVERSE.
C
            B(K) = B(K)/AP(KK)
            K = K - 1
            IK = IK - K
         GO TO 70
   40    CONTINUE
C
C           2 X 2 PIVOT BLOCK.
C
            IKM1 = IK - (K - 1)
            IF (K .EQ. 2) GO TO 60
               KP = IABS(KPVT(K))
               IF (KP .EQ. K - 1) GO TO 50
C
C                 INTERCHANGE.
C
                  TEMP = B(K-1)
                  B(K-1) = B(KP)
                  B(KP) = TEMP
   50          CONTINUE
C
C              APPLY THE TRANSFORMATION.
C
               CALL SAXPY(K-2,B(K),AP(IK+1),1,B(1),1)
               CALL SAXPY(K-2,B(K-1),AP(IKM1+1),1,B(1),1)
   60       CONTINUE
C
C           APPLY D INVERSE.
```

```
C
            KM1K = IK + K - 1
            KK = IK + K
            AK = AP(KK)/AP(KM1K)
            KM1KM1 = IKM1 + K - 1
            AKM1 = AP(KM1KM1)/AP(KM1K)
            BK = B(K)/AP(KM1K)
            BKM1 = B(K-1)/AP(KM1K)
            DENOM = AK*AKM1 - 1.0E0
            B(K) = (AKM1*BK - BKM1)/DENOM
            B(K-1) = (AK*BKM1 - BK)/DENOM
            K = K - 2
            IK = IK - (K + 1) - K
   70    CONTINUE
      GO TO 10
   80 CONTINUE
C
C     LOOP FORWARD APPLYING THE TRANSFORMATIONS.
C
      K = 1
      IK = 0
   90 IF (K .GT. N) GO TO 160
         IF (KPVT(K) .LT. 0) GO TO 120
C
C           1 X 1 PIVOT BLOCK.
C
            IF (K .EQ. 1) GO TO 110
C
C              APPLY THE TRANSFORMATION.
C
               B(K) = B(K) + SDOT(K-1,AP(IK+1),1,B(1),1)
               KP = KPVT(K)
               IF (KP .EQ. K) GO TO 100
C
C                 INTERCHANGE.
C
                  TEMP = B(K)
                  B(K) = B(KP)
                  B(KP) = TEMP
  100          CONTINUE
  110       CONTINUE
            IK = IK + K
            K = K + 1
         GO TO 150
  120    CONTINUE
C
C           2 X 2 PIVOT BLOCK.
C
            IF (K .EQ. 1) GO TO 140
C
C              APPLY THE TRANSFORMATION.
C
               B(K) = B(K) + SDOT(K-1,AP(IK+1),1,B(1),1)
               IKP1 = IK + K
```

```
                B(K+1) = B(K+1) + SDOT(K-1,AP(IKP1+1),1,B(1),1)
                KP = IABS(KPVT(K))
                IF (KP .EQ. K) GO TO 130
C
C                  INTERCHANGE.
C
                   TEMP = B(K)
                   B(K) = B(KP)
                   B(KP) = TEMP
  130           CONTINUE
  140        CONTINUE
             IK = IK + K + K + 1
             K = K + 2
  150     CONTINUE
          GO TO 90
  160     CONTINUE
          RETURN
          END
C
C
C
          SUBROUTINE SSPCO(AP,N,KPVT,RCOND,Z)
          INTEGER N,KPVT(1)
          REAL AP(1),Z(1)
          REAL RCOND
C
C         SSPCO FACTORS A REAL SYMMETRIC MATRIX STORED IN PACKED
C         FORM BY ELIMINATION WITH SYMMETRIC PIVOTING AND ESTIMATES
C         THE CONDITION OF THE MATRIX.
C
C         IF  RCOND  IS NOT NEEDED, SSPFA IS SLIGHTLY FASTER.
C         TO SOLVE  A*X = B , FOLLOW SSPCO BY SSPSL.
C         TO COMPUTE  INVERSE(A)*C , FOLLOW SSPCO BY SSPSL.
C         TO COMPUTE  INVERSE(A) , FOLLOW SSPCO BY SSPDI.
C         TO COMPUTE  DETERMINANT(A) , FOLLOW SSPCO BY SSPDI.
C         TO COMPUTE  INERTIA(A), FOLLOW SSPCO BY SSPDI.
C
C         ON ENTRY
C
C            AP       REAL (N*(N+1)/2)
C                     THE PACKED FORM OF A SYMMETRIC MATRIX  A .  THE
C                     COLUMNS OF THE UPPER TRIANGLE ARE STORED SEQUENTIALLY
C                     IN A ONE-DIMENSIONAL ARRAY OF LENGTH  N*(N+1)/2 .
C                     SEE COMMENTS BELOW FOR DETAILS.
C
C            N        INTEGER
C                     THE ORDER OF THE MATRIX  A .
C
C         OUTPUT
C
C            AP       A BLOCK DIAGONAL MATRIX AND THE MULTIPLIERS WHICH
C                     WERE USED TO OBTAIN IT STORED IN PACKED FORM.
C                     THE FACTORIZATION CAN BE WRITTEN  A = U*D*TRANS(U)
C                     WHERE  U  IS A PRODUCT OF PERMUTATION AND UNIT
```

```
C               UPPER TRIANGULAR MATRICES , TRANS(U) IS THE
C               TRANSPOSE OF  U , AND  D  IS BLOCK DIAGONAL
C               WITH 1 BY 1 AND 2 BY 2 BLOCKS.
C
C        KPVT   INTEGER(N)
C               AN INTEGER VECTOR OF PIVOT INDICES.
C
C        RCOND  REAL
C               AN ESTIMATE OF THE RECIPROCAL CONDITION OF  A .
C               FOR THE SYSTEM  A*X = B , RELATIVE PERTURBATIONS
C               IN  A  AND  B  OF SIZE  EPSILON  MAY CAUSE
C               RELATIVE PERTURBATIONS IN  X  OF SIZE  EPSILON/RCOND .
C               IF  RCOND  IS SO SMALL THAT THE LOGICAL EXPRESSION
C                         1.0 + RCOND .EQ. 1.0
C               IS TRUE, THEN  A  MAY BE SINGULAR TO WORKING
C               PRECISION.  IN PARTICULAR,  RCOND  IS ZERO  IF
C               EXACT SINGULARITY IS DETECTED OR THE ESTIMATE
C               UNDERFLOWS.
C
C        Z      REAL(N)
C               A WORK VECTOR WHOSE CONTENTS ARE USUALLY UNIMPORTANT.
C               IF  A  IS CLOSE TO A SINGULAR MATRIX, THEN  Z  IS
C               AN APPROXIMATE NULL VECTOR IN THE SENSE THAT
C               NORM(A*Z) = RCOND*NORM(A)*NORM(Z) .
C
C     PACKED STORAGE
C
C          THE FOLLOWING PROGRAM SEGMENT WILL PACK THE UPPER
C          TRIANGLE OF A SYMMETRIC MATRIX.
C
C                K = 0
C                DO 20 J = 1, N
C                   DO 10 I = 1, J
C                      K = K + 1
C                      AP(K) = A(I,J)
C             10    CONTINUE
C             20 CONTINUE
C
C     LINPACK. THIS VERSION DATED 03/11/78 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     LINPACK SSPFA
C     BLAS SAXPY,SDOT,SSCAL,SASUM
C     FORTRAN ABS,AMAX1,IABS,SIGN
C
C     INTERNAL VARIABLES
C
      REAL AK,AKM1,BK,BKM1,SDOT,DENOM,EK,T
      REAL ANORM,S,SASUM,YNORM
      INTEGER I,IJ,IK,IKM1,IKP1,INFO,J,JM1,J1
      INTEGER K,KK,KM1K,KM1KM1,KP,KPS,KS
C
```

```
C
C     FIND NORM OF A USING ONLY UPPER HALF
C
      J1 = 1
      DO 30 J = 1, N
         Z(J) = SASUM(J,AP(J1),1)
         IJ = J1
         J1 = J1 + J
         JM1 = J - 1
         IF (JM1 .LT. 1) GO TO 20
         DO 10 I = 1, JM1
            Z(I) = Z(I) + ABS(AP(IJ))
            IJ = IJ + 1
   10    CONTINUE
   20    CONTINUE
   30 CONTINUE
      ANORM = 0.0E0
      DO 40 J = 1, N
         ANORM = AMAX1(ANORM,Z(J))
   40 CONTINUE
C
C     FACTOR
C
      CALL SSPFA(AP,N,KPVT,INFO)
C
C     RCOND = 1/(NORM(A)*(ESTIMATE OF NORM(INVERSE(A)))) .
C     ESTIMATE = NORM(Z)/NORM(Y) WHERE  A*Z = Y  AND  A*Y = E .
C     THE COMPONENTS OF  E  ARE CHOSEN TO CAUSE MAXIMUM LOCAL
C     GROWTH IN THE ELEMENTS OF W  WHERE  D*W = U*E .
C     THE VECTORS ARE FREQUENTLY RESCALED TO AVOID OVERFLOW.
C
C     SOLVE D*W = U*E
C
      EK = 1.0E0
      DO 50 J = 1, N
         Z(J) = 0.0E0
   50 CONTINUE
      K = N
      IK = (N*(N - 1))/2
   60 IF (K .EQ. 0) GO TO 120
         KK = IK + K
         IKM1 = IK - (K - 1)
         KS = 1
         IF (KPVT(K) .LT. 0) KS = 2
         KP = IABS(KPVT(K))
         KPS = K + 1 - KS
         IF (KP .EQ. KPS) GO TO 70
            T = Z(KPS)
            Z(KPS) = Z(KP)
            Z(KP) = T
   70    CONTINUE
         IF (Z(K) .NE. 0.0E0) EK = SIGN(EK,Z(K))
         Z(K) = Z(K) + EK
         CALL SAXPY(K-KS,Z(K),AP(IK+1),1,Z(1),1)
```

```
      IF (KS .EQ. 1) GO TO 80
         IF (Z(K-1) .NE. 0.0E0) EK = SIGN(EK,Z(K-1))
         Z(K-1) = Z(K-1) + EK
         CALL SAXPY(K-KS,Z(K-1),AP(IKM1+1),1,Z(1),1)
   80    CONTINUE
      IF (KS .EQ. 2) GO TO 100
         IF (ABS(Z(K)) .LE. ABS(AP(KK))) GO TO 90
            S = ABS(AP(KK))/ABS(Z(K))
            CALL SSCAL(N,S,Z,1)
            EK = S*EK
   90       CONTINUE
         IF (AP(KK) .NE. 0.0E0) Z(K) = Z(K)/AP(KK)
         IF (AP(KK) .EQ. 0.0E0) Z(K) = 1.0E0
      GO TO 110
  100 CONTINUE
         KM1K = IK + K - 1
         KM1KM1 = IKM1 + K - 1
         AK = AP(KK)/AP(KM1K)
         AKM1 = AP(KM1KM1)/AP(KM1K)
         BK = Z(K)/AP(KM1K)
         BKM1 = Z(K-1)/AP(KM1K)
         DENOM = AK*AKM1 - 1.0E0
         Z(K) = (AKM1*BK - BKM1)/DENOM
         Z(K-1) = (AK*BKM1 - BK)/DENOM
  110 CONTINUE
      K = K - KS
      IK = IK - K
      IF (KS .EQ. 2) IK = IK - (K + 1)
   GO TO 60
  120 CONTINUE
   S = 1.0E0/SASUM(N,Z,1)
   CALL SSCAL(N,S,Z,1)
C
C     SOLVE TRANS(U)*Y = W
C
   K = 1
   IK = 0
  130 IF (K .GT. N) GO TO 160
      KS = 1
      IF (KPVT(K) .LT. 0) KS = 2
      IF (K .EQ. 1) GO TO 150
         Z(K) = Z(K) + SDOT(K-1,AP(IK+1),1,Z(1),1)
         IKP1 = IK + K
         IF (KS .EQ. 2)
     *      Z(K+1) = Z(K+1) + SDOT(K-1,AP(IKP1+1),1,Z(1),1)
         KP = IABS(KPVT(K))
         IF (KP .EQ. K) GO TO 140
            T = Z(K)
            Z(K) = Z(KP)
            Z(KP) = T
  140    CONTINUE
  150 CONTINUE
      IK = IK + K
      IF (KS .EQ. 2) IK = IK + (K + 1)
```

```
            K = K + KS
         GO TO 130
  160 CONTINUE
         S = 1.0E0/SASUM(N,Z,1)
         CALL SSCAL(N,S,Z,1)
C
         YNORM = 1.0E0
C
C        SOLVE D*V = U*Y
C
         K = N
         IK = N*(N - 1)/2
  170 IF (K .EQ. 0) GO TO 230
            KK = IK + K
            IKM1 = IK - (K - 1)
            KS = 1
            IF (KPVT(K) .LT. 0) KS = 2
            IF (K .EQ. KS) GO TO 190
               KP = IABS(KPVT(K))
               KPS = K + 1 - KS
               IF (KP .EQ. KPS) GO TO 180
                  T = Z(KPS)
                  Z(KPS) = Z(KP)
                  Z(KP) = T
  180          CONTINUE
               CALL SAXPY(K-KS,Z(K),AP(IK+1),1,Z(1),1)
               IF (KS .EQ. 2) CALL SAXPY(K-KS,Z(K-1),AP(IKM1+1),1,Z(1),1)
  190       CONTINUE
            IF (KS .EQ. 2) GO TO 210
               IF (ABS(Z(K)) .LE. ABS(AP(KK))) GO TO 200
                  S = ABS(AP(KK))/ABS(Z(K))
                  CALL SSCAL(N,S,Z,1)
                  YNORM = S*YNORM
  200          CONTINUE
               IF (AP(KK) .NE. 0.0E0) Z(K) = Z(K)/AP(KK)
               IF (AP(KK) .EQ. 0.0E0) Z(K) = 1.0E0
            GO TO 220
  210       CONTINUE
               KM1K = IK + K - 1
               KM1KM1 = IKM1 + K - 1
               AK = AP(KK)/AP(KM1K)
               AKM1 = AP(KM1KM1)/AP(KM1K)
               BK = Z(K)/AP(KM1K)
               BKM1 = Z(K-1)/AP(KM1K)
               DENOM = AK*AKM1 - 1.0E0
               Z(K) = (AKM1*BK - BKM1)/DENOM
               Z(K-1) = (AK*BKM1 - BK)/DENOM
  220       CONTINUE
            K = K - KS
            IK = IK - K
            IF (KS .EQ. 2) IK = IK - (K + 1)
         GO TO 170
  230 CONTINUE
         S = 1.0E0/SASUM(N,Z,1)
```

```
      CALL SSCAL(N,S,Z,1)
      YNORM = S*YNORM
C
C     SOLVE TRANS(U)*Z = V
C
      K = 1
      IK = 0
  240 IF (K .GT. N) GO TO 270
         KS = 1
         IF (KPVT(K) .LT. 0) KS = 2
         IF (K .EQ. 1) GO TO 260
            Z(K) = Z(K) + SDOT(K-1,AP(IK+1),1,Z(1),1)
            IKP1 = IK + K
            IF (KS .EQ. 2)
     *         Z(K+1) = Z(K+1) + SDOT(K-1,AP(IKP1+1),1,Z(1),1)
            KP = IABS(KPVT(K))
            IF (KP .EQ. K) GO TO 250
               T = Z(K)
               Z(K) = Z(KP)
               Z(KP) = T
  250       CONTINUE
  260    CONTINUE
         IK = IK + K
         IF (KS .EQ. 2) IK = IK + (K + 1)
         K = K + KS
      GO TO 240
  270 CONTINUE
C     MAKE ZNORM = 1.0
      S = 1.0E0/SASUM(N,Z,1)
      CALL SSCAL(N,S,Z,1)
      YNORM = S*YNORM
C
      IF (ANORM .NE. 0.0E0)   :OND = YNORM/ANORM
      IF (ANORM .EQ. 0.0E0) .  'D = 0.0E0
      RETURN
      END
```

```
C ** THIS PROGRAM (OUTPUT.F4) IS A UTILITY PROGRAM THAT READS DATA THAT
C ** WERE OUTPUT FROM BOTH INPUT.F4 (UNIT #'S IRN (=2), AND IRM (=10))
C ** AND GLUCS.F4 (UNIT #'S IRE (=15), AND IRV (=16)).
C ** IF THE DIMENSIONS OF THE PROGRAM NEED TO BE INCREASED FOR A PARTICULAR
C ** PROBLEM THE PROGRAM WILL STOP AND TYPE AN ERROR MESSAGE. THAT IS,
C ** THE USER IS TOLD TO INCREASE THE DIMENSIONS FOR EITHER THE REAL
C ** VARIABLES OR THE INTEGER VARIABLES. IN  OTHER WORDS, THE USER WILL
C ** NEED TO INCREASE THE DIMENSIONS OF ARRAY A AND INCREASE THE NUMBER
C ** NSIZER IN THE MAIN PROGRAM (REAL VARIABLES) OR INCREASE THE DIMENSION
C ** OF ARRAY IA AND INCREASE THE NUMBER NSIZEI IN THE MAIN PROGRAM (INTEGER
C ** VARIABLES). (THE DIMENSION OF ARRAY A AND THE NUMBER NSIZER MUST
C ** BE THE SAME NUMBER - LIKEWISE FOR ARRAY IA AND NSIZEI). IMPORTANT
C ** PARAMETERS THROUGHOUT THE PROGRAM ARE DEFINED.
      COMMON/PARRL/NSIZER,A(25000)
      COMMON/PARIN/NSIZEI,IA(7000)
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
      COMMON/WHERE/INEVP,IEEV,IXSNEW,IMAT,IMT,IQ,IZA,IAWR,ISTD,IVARNW,
     $                  IVARBK,IIJTHN,IEEVTN,IXSTHN,ISTDTN,IICOR,
     $                  INPTHN,IENF
C ** NSIZER, NSIZEI BASED ON ? EVALUATED SETS
C ** ? POINTS PER EVALUATED SET  (SEE USERS MANUAL)
      NSIZEI=7000
      NSIZER=25000
C ** GENERATE ENDF FILE 3 PARAMETERS (ENERGIES, X-SECTIONS)
      CALL PAREXS
C ** GENERATE ENDF FILE 33 USING LB = 5 FOR DIAGONAL BLOCKS OF MATRIX M'
      CALL PARDIA
C ** GENERATE ENDF FILE 33 USING LB = 6 FOR OFF-DIAGONAL
C ** RECTANGULAR BLOCKS OF MATRIX M' (DONE ONLY IF # OF EVALUATED
C ** SETS (NEVSET) > 1)
      IF(NEVSET.EQ.1)STOP
      CALL PARREC
      STOP
      END
C
      FUNCTION IDIMR(MANY)
C ** PURPOSE - KEEP TRACK OF DIMENSIONS OF REAL ARRAYS
      COMMON/PARRL/NSIZER,A(1)
      DATA KOUNT/1/
      IDIMR=KOUNT
      KOUNT=KOUNT+MANY
      IF(KOUNT.LE.NSIZER)RETURN
      TYPE 10,NSIZER,KOUNT
 10   FORMAT(1X,'AVAILABLE SIZE (REAL) = ',I6,' BUT YOU NEED ',I6)
      STOP
      RETURN
      END
C
      FUNCTION IDIMI(MNY)
C ** PURPOSE - KEEP TRACK OF DIMENSIONS OF INTEGER ARRAYS
      COMMON/PARIN/NSIZEI,IA(1)
      DATA KONT/1/
      IDIMI=KONT
      KONT=KONT+MNY
```

```
      IF(KONT.LE.NSIZEI)RETURN
      TYPE 10,NSIZEI,KONT
 10   FORMAT(1X,'AVAILABLE SIZE (INTEGER) = ',I5,'BUT YOU NEED ',I5)
      STOP
      RETURN
      END
C
      SUBROUTINE PAREXS
C ** THIS SUBROUTINE SETS UP THE DIMENSIONS FOR WRITING THE ENDF/B FILE
C ** 3 (ENERGIES, X-SECTIONS, MATERIAL #'S, ETC.). SUBROUTINE RDMEXS
C ** IS CALLED TO READ THE MATERIAL #'S (OUTPUT FROM INPUT.F4), AND THE
C ** ENERGIES AND X-SECTIONS (OUTPUT FROM GLUCS.F4).
      COMMON/PARRL/NSIZER,A(1)
      COMMON/PARIN/NSIZEI,IA(1)
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
      COMMON/WHERE/INEVP,IEEV,IXSNEW,IMAT,IMT,IQ,IZA,IAWR,ISTD,IVARNW,
     $                  IVARBK,IIJTHN,IEEVTN,IXSTHN,ISTDTN,IICOR,
     $                  INPTHN,IENF
      IRN=2
C ** READ FILE FROM THE OUTPUT OF INPUT.F4 WITH UNIT # IRN (=2).
      READ(IRN,10)NEVPTS,NEVSET,NLGEVS
C ** NEVPTS= TOTAL # OF EVALUATED POINTS IN ALL SETS, NEVSET= # OF EVALUATED
C ** SETS, AND NLGEVS= # IN LARGEST EVALUATED SET.
 10   FORMAT(1X,3I3)
C ** THE FOLLOWING PARAMETERS DETERMINE THE DIMENSIONS OF ARRAYS THAT
C ** ARE DEFINED IN SUBROUTINE RDMEXS. EG, IEEV IS THE DIMENSION SET
C ** ASIDE FOR ARRAY EEV (EVALUATED ENERGY).
      INEVP=IDIMI(NEVSET)
      IEEV=IDIMR(NEVPTS)
      IXSNEW=IDIMR(NEVPTS)
      IMAT=IDIMI(NEVSET)
      IMT=IDIMI(NEVSET)
      IQ=IDIMR(NEVSET)
      IZA=IDIMR(NEVSET)
      IAWR=IDIMR(NEVSET)
      CALL RDMEXS(IA(INEVP),A(IEEV),A(IXSNEW),IA(IMAT),IA(IMT),
     $          A(IQ),A(IZA),A(IAWR))
      RETURN
      END
C
      SUBROUTINE RDMEXS(NEVP,EEV,XSNEW,MAT,MT,Q,ZA,AWR)
C ** THIS ROUTINE READS DATA FROM FILES THAT WERE OUTPUT FROM  THE INPUT.F4
C ** (UNIT # IRM (=10)) AND GLUCS.F4 (UNIT # IRE (=15)) PROGRAMS AND
C ** CALLS SUBROUTINE WRIEXS TO WRITE THE ENERGIES AND NEW X-SECTIONS
C ** IN ENDF/B FORMAT (FILE 3).
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
      DIMENSION NEVP(1),EEV(1),XSNEW(1),MAT(1),MT(1),Q(1),ZA(1),AWR(1)
      DIMENSION NBT(3),INTP(3)
      IRM=10
      IRE=15
      II=0
      DO 50 NEV=1,NEVSET
      NINEV=II+1
      READ(IRE,10)NPSET,NEVP(NEV),NR
```

```
C ** NPSET= # OF POINTS IN SET NEV
C ** NEVP(NEV)= EG, IF NEVSET=3, NEVPTS=16, AND THERE ARE 4 POINTS IN
C ** EVALUATED SET 1, AND 6 POINTS IN EVALUATED SET 2, THEN NEVP(1)=4,
C ** NEVP(2)=10, AND NEVP(3)=16.
C ** NR= # OF INTERPOLATION RANGES IN SET NEV.
 10     FORMAT(1X,6I3)
        IF(NR.LE.3)GO TO 15
        TYPE 12,NR
 12     FORMAT(1X,'STOP: NR = ',I2,' INCREASE DIMENSIONS')
        STOP
 15     READ(IRE,10)(NBT(I),INTP(I),I=1,3)
C ** NBT(I)= # VALUE SEPARATING I-TH AND (I+1)-TH INTERPOLATION RANGES,
C ** INTP(I)= INTERPOLATION SCHEME ID # USED IN I-TH RANGE.
        DO 20 I=1,NPSET
        II=II+1
 20     READ(IRE,30)EEV(II),XSNEW(II)
C ** EEV= EVALUATED ENERGY, XSNEW= NEW X-SECTION CALCULATED BY BAYES THEOREM.
 30     FORMAT(1X,1PE11.4,1X,1PE11.4)
        READ(IRM,40)MAT(NEV),MT(NEV),Q(NEV),ZA(NEV),AWR(NEV)
C ** FOR SET NEV, MAT(NEV)= MATERIAL #, MT(NEV)= REACTION TYPE IDENTIFIER,
C ** Q(NEV)= REACTION Q-VALUE, ZA(NEV)= THE (Z,A) DESIGNATION FOR MATERIAL,
C ** AND AWR(NEV)= THE RATIO OF THE MASS OF AN ATOM (OR MOLECULE) TO
C ** THAT OF THE NEUTRON.
 40     FORMAT(1X,I4,I3,1X,3(1PE11.4))
        CALL WRIEXS(NPSET,NR,NBT,INTP,EEV(NINEV),XSNEW(NINEV),MAT(NEV),
     $          MT(NEV),Q(NEV),ZA(NEV),AWR(NEV))
 50     CONTINUE
        RETURN
        END
C
        SUBROUTINE WRIEXS(NPSET,NR,NBT,INTP,EEV,XSNEW,MAT,MT,Q,ZA,AWR)
C ** THIS ROUTINE WRITES THE EVALUATED ENERGIES AND NEW X-SECTIONS (FROM
C ** BAYES THEOREM) INTO ENDF/B FORMAT (FILE 3).
        DIMENSION NBT(1),INTP(1),EEV(1),XSNEW(1)
        DIMENSION EZERO(2),XSZERO(2)
        DATA IWR/17/
C ** DATA ARE WRITTEN IN FILES WITH UNIT #'S IWR (=18,19,...,(17+NEVSET)).
        IWR=IWR+1
        MF=3
        LO=0
        B=0.0
        L1=1
        L2=2
        NP=NPSET
        WRITE(IWR,10)ZA,AWR,LO,LO,LO,LO,MAT,MF,MT
        WRITE(IWR,10)B,Q,LO,LO,NR,NP,MAT,MF,MT
 10     FORMAT(1P2E11.4,4I11,I4,I2,I3)
        WRITE(IWR,20)(NBT(I),INTP(I),I=1,3),MAT,MF,MT
 20     FORMAT(6I11,I4,I2,I3)
        DO 30 I=1,NP,3
        II=I
        IP2=I+2
        IF(IP2.GT.NP)GO TO 50
 30     WRITE(IWR,40)(EEV(K),XSNEW(K),K=I,IP2),MAT,MF,MT
```

```
 40     FORMAT(3(1PE11.5,E11.4),I4,I2,I3)
        GO TO 70
 50     IP2MNP=IP2-NP
        DO 60 I=1,IP2MNP
        EZERO(I)=0.0
 60     XSZERO(I)=0.0
        WRITE(IWR,40)(EEV(K),XSNEW(K),K=II,NP),(EZERO(K),XSZERO(K),
     $          K=1,IP2MNP),MAT,MF,MT
 70     WRITE(IWR,10)B,B,L0,L0,L0,L0,MAT,MF,L0
        WRITE(IWR,10)B,B,L0,L0,L0,L0,MAT,L0,L0
        RETURN
        END
C
        SUBROUTINE PARDIA
C ** THIS SUBROUTINE SETS UP THE DIMENSIONS NEEDED FOR WRITING THE COVARIANCE
C ** OF THE DIAGONAL BLOCKS OF MATRIX M' INTO ENDF/B FORMAT (FILE 33, LB=5).
C ** SUBROUTINE PARDI IS CALLED TO READ THE COVARIANCE (OUTPUT FROM GLUCS.F4).
        COMMON/PARRL/NSIZER,A(1)
        COMMON/PARIN/NSIZEI,IA(1)
        COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
        COMMON/WHERE/INEVP,IEEV,IXSNEW,IMAT,IMT,IQ,IZA,IAWR,ISTD,IVARNW,
     $                  IVARBK,IIJTHN,IEEVTN,IXSTHN,ISTDTN,IICOR,
     $                  INPTHN,IENF
C ** THE FOLLOWING PARAMETERS DETERMINE THE DIMENSIONS OF ARRAYS THAT ARE
C ** USED IN SUBROUTINE PARDI, CORBLK, THNDIA, PNTDIA, & WRILB5 BELOW.
C ** EG, ISTD IS THE DIMENSION SET ASIDE  IN ARRAY A FOR ARRAY STD (STANDARD
C ** DEVIATION).
        ISTD=IDIMR(NLGEVS)
        NLGNEV=NLGEVS*NEVPTS
        IVARNW=IDIMR(NLGNEV)
        NLGSQR=NLGEVS*NLGEVS
        IVARBK=IDIMR(NLGSQR)
        IIJTHN=IDIMI(NLGEVS)
        NLGSET=NLGEVS*NEVSET
        IEEVTN=IDIMR(NLGSET)
        IXSTHN=IDIMR(NLGSET)
        ISTDTN=IDIMR(NLGSET)
        IICOR=IDIMI(NLGSQR)
        INPTHN=IDIMI(NEVSET)
        NUM=2*NLGEVS+NLGSQR+7
        IENF=IDIMR(NUM)
C ** NOTE THAT IENF REPRESENTS STORAGE FOR ENF ARRAY IN SUBROUTINE
C ** WRILB6 AS WELL AS WRILB5
        CALL PARDI(IA(INEVP),A(IEEV),A(IXSNEW),A(ISTD),A(IVARNW),
     $          A(IVARBK),IA(IIJTHN),A(IEEVTN),A(IXSTHN),A(ISTDTN),
     $          IA(IICOR),IA(INPTHN),IA(IMAT),IA(IMT),A(IZA),
     $          A(IAWR),A(IENF))
        RETURN
        END
C
        SUBROUTINE PARDI(NEVP,EEV,XSNEW,STD,VARNEW,VARBLK,IJTHN,EEVTHN,
     $          XSTHN,STDTHN,ICOR,NPTHN,MAT,MT,ZA,AWR,ENF)
C ** PARDI READS THE COVARIANCE THAT IS OUTPUT FROM GLUCS.F4 (READS FILE
C ** WITH UNIT # IRV (=16)). VARBLK IS SET TO THE LOWER TRIANGULAR ELEMENTS
```

```
C ** OF EACH OF THE DIAGONAL BLOCKS OF MATRIX M'. SUBROUTINE CORBLK IS
C ** CALLED TO DETERMINE WHICH ROWS (COLUMNS) OF EACH DIAGONAL BLOCK
C ** OF M' ARE CORRELATED, AND SUBROUTINE THNDIA IS CALLED TO THIN
C ** EACH BLOCK.
      COMMON/NUMFV/NEVSET,NEVPTS,NLGEVS
      DIMENSION NEVP(1),EEV(1),XSNEW(1),STD(1),VARNEW(NEVPTS,1),
     $          VARBLK(1),IJTHN(1),EEVTHN(NLGEVS,1),XSTHN(NLGEVS,1),
     $          STDTHN(NLGEVS,1),ICOR(1),NPTHN(1),MAT(1),MT(1),
     $          ZA(1),AWR(1),ENF(1)
      IRV=16
      NPSET=NEVP(1)
      NINEV=1
      DO 60 NEV=1,NEVSET
      IF(NEV.EQ.1)GO TO 10
      NPSET=NEVP(NEV)-NEVP(NEV-1)
      NINEV=NEVP(NEV-1)+1
 10   CONTINUE
      DO 20 J=1,NPSET
 20   READ(IRV,30)(VARNEW(I,J),I=1,NEVPTS)
 30   FORMAT(1X,10(1PE11.4))
C ** SET VARBLK TO LOWER TRIANGULAR ELEMENTS OF BLOCK (NEV,NEV)
C ** OF THE MATRIX M'
      I=NINEV
      J=1
      L=0
      NPLTR=NPSET*(NPSET+1)/2
      DO 50 IJ=1,NPLTR
      VARBLK(IJ)=VARNEW(I,J)
      L=L+1
      IF(L.NE.J)GO TO 40
      I=NINEV
      J=J+1
      L=0
      GO TO 50
 40   I=I+1
 50   CONTINUE
      CALL CORBLK(NPSET,XSNEW(NINEV),STD,VARBLK,NBLK,IJTHN)
      CALL THNDIA(NEV,NPSET,EEV(NINEV),XSNEW(NINEV),STD,VARBLK,NBLK,
     $          IJTHN,EEVTHN,XSTHN,STDTHN,ICOR,NPTHN,MAT,MT,
     $          ZA,AWR,ENF)
 60   CONTINUE
      RETURN
      END
C
      SUBROUTINE CORBLK(NPSET,XS,STD,VARBLK,NBLK,IJTHN)
C ** THIS SUBROUTINE DETERMINES WHICH ROWS (COLUMNS) OF EACH DIAGONAL
C ** BLOCK OF MATRIX M' ARE CORRELATED TOGETHER AND STORES THIS INFORMATION
C ** IN ARRAY IJTHN.
      DIMENSION XS(1),STD(1),VARBLK(1),IJTHN(1)
      NP=NPSET
      DO 20 I=1,NP
      II=I*(I+1)/2
      IF(XS(I).EQ.0.0)GO TO 10
      STD(I)=100.0*SQRT(VARBLK(II))/XS(I)
```

```
      GO TO 20
10    STD(I)=0.0
20    CONTINUE
      NBLK=1
      IJTHN(1)=NP
      IBLK=1
      DO 40 I=2,NP
      J=I-1
      IDIA=I*(I+1)/2
      IDIAM1=IDIA-1
      IF(XS(I).EQ.0.0.OR.XS(J).EQ.0.0.OR.STD(I).EQ.0.0.OR.
     $        STD(J).EQ.0.0)GO TO 30
      ICOR=IFIX(1.0E+6*VARBLK(IDIAM1)/(XS(I)*XS(J)*STD(I)*STD(J))+0.5)
      IF(ICOR.EQ.100)GO TO 40
30    IJTHN(IBLK)=J
      IBLK=IBLK+1
      NBLK=IBLK
      IJTHN(IBLK)=NP
40    CONTINUE
      RETURN
      END
C
      SUBROUTINE THNDIA(NEV,NPSET,EEV,XSNEW,STD,VARBLK,NBLK,IJTHN,
     $        EEVTHN,XSTHN,STDTHN,ICOR,NPTHN,MAT,MT,ZA,AWR,ENF)
C ** THNDIA THINS OUT THE CORRELATED ROWS (COLUMNS) FOR EACH DIAGONAL
C ** BLOCK OF COVARIANCE MATRIX M'. IT THEN CALLS PNTDIA TO PRINT THESE
C ** RESULTS IN READABLE FORM AND CALLS WRILB5 TO WRITE THE COVARIANCE
C ** INTO ENDF/B FORMAT (LB=5).
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
      DIMENSION EEV(1),XSNEW(1),STD(1),VARBLK(1),IJTHN(1),
     $        EEVTHN(NLGEVS,1),XSTHN(NLGEVS,1),STDTHN(NLGEVS,1),
     $        NPTHN(1),MAT(1),MT(1),ZA(1),AWR(1),ENF(1),ICOR(1)
      NVRBLK=NPSET*(NPSET+1)/2.0
      LCNT=0
      ITHN=0
      IA=0
      DO 60 IBLK=1,NBLK
      IF(IBLK.NE.1)IA=IJTHN(IBLK-1)
      IB=IJTHN(IBLK)
      ITHN=ITHN+1
      IAP1=IA+1
C ** STORE THINNED ENERGIES, X-SECTIONS, AND STANDARD DEVIATIONS IN ARRAYS
C ** EEVTHN, XSTHN, AND STDTHN RESPECTIVELY (FOR EACH SET NEV)
      EEVTHN(ITHN,NEV)=EEV(IAP1)
      XSTHN(ITHN,NEV)=XSNEW(IAP1)
      STDTHN(ITHN,NEV)=STD(IAP1)
      IF((IB-IA).EQ.1)GO TO 60
      LCRIN=IA+2
      LCRF=IB
      IF(IB.NE.NPSET)GO TO 10
      ITHN=ITHN+1
      EEVTHN(ITHN,NEV)=EEV(IB)
      XSTHN(ITHN,NEV)=XSNEW(IB)
      STDTHN(ITHN,NEV)=STD(IB)
```

```
      LCRF=LCRF-1
      IF(LCRF.LT.LCRIN)GO TO 60
C ** SET ROWS AND COLUMNS OF COVARIANCE MATRIX M' (VARBLK) THAT NEED TO
C ** BE THINNED TO 1.0E+5.
 10   DO 50 L=LCRIN,LCRF
C ** 1ST, SET ROW OF COVARIANCE MATRIX TO 1.0E+5
      LP1=L+1
      LM1=L-1
      IRIN=1+LM1*L/2
      IRF=L*LP1/2
      DO 20 M=IRIN,IRF
 20   VARBLK(M)=1.0E+5
C ** NOW, SET COLUMN TO 1.0E+5
      IROW=LP1
 30   IRIN=IRF+1
      IRF=IROW*(IROW+1)/2
      IF(IRF.GT.NVRBLK)GO TO 50
      IROW=IROW+1
      JCOL=0
      DO 40 M=IRIN,IRF
      JCOL=JCOL+1
      IF(JCOL.NE.L)GO TO 40
      VARBLK(M)=1.0E+5
      GO TO 30
 40   CONTINUE
 50   CONTINUE
 60   CONTINUE
      NPTHN(NEV)=ITHN
C ** THIN OUT COVARIANCE MATRIX
      IJ=1
 70   DO 90 M=IJ,NVRBLK
      IF(VARBLK(M).NE.1.0E+5)GO TO 90
      NVRM1=NVRBLK-1
      IF(NVRM1.LT.M)GO TO 110
      DO 80 L=M,NVRM1
 80   VARBLK(L)=VARBLK(L+1)
      GO TO 100
 90   CONTINUE
      GO TO 120
 100  NVRBLK=NVRM1
      IJ=M
      GO TO 70
 110  NVRBLK=NVRM1
 120  CONTINUE
C ** CALL PNTDIA TO PRINT THINNED PARAMETERS OF DIAGONAL BLOCKS OF M'
      CALL PNTDIA(NEV,NPTHN(NEV),EEVTHN(1,NEV),XSTHN(1,NEV),
     $          STDTHN(1,NEV),VARBLK,ICOR)
C ** CALL WRILB5 TO WRITE FILE 33 IN ENDF FORMAT FOR SET NEV
      CALL WRILB5(NPTHN(NEV),EEVTHN(1,NEV),XSTHN(1,NEV),VARBLK,
     $          MAT(NEV),MT(NEV),ZA(NEV),AWR(NEV),ENF)
      RETURN
      END
C
      SUBROUTINE PNTDIA(NEV,NPTHN,EN,XS,STD,VAR,ICOR)
```

```
C ** THIS SUBROUTINE PRINTS THE THINNED ENERGIES (EN), X-SECTIONS (XS),
C ** AND STANDARD DEVIATIONS (STD) IN A FILE WITH UNIT # IWP (=17). ALSO,
C ** IT CALCULATES THE CORRELATION MATRIX ELEMENTS (ICOR) FROM THE
C ** COVARIANCE (VAR) & PRINTS THE RESULTS (FOR EACH DIAGONAL BLOCK OF
C ** MATRIX M').
      DIMENSION EN(1),XS(1),STD(1),VAR(1),ICOR(1)
      IWP=17
      WRITE(IWP,10)NEV
 10   FORMAT(1X,'FINAL THINNED DATA OF # ',I2,' SET')
      WRITE(IWP,20)
 20   FORMAT(2X,'I',6X,'E(I)',6X,'X(E(I))',3X,'STD(X(E(I)))')
      NPT=NPTHN
      DO 30 I=1,NPT
 30   WRITE(IWP,40)I,EN(I),XS(I),STD(I)
 40   FORMAT(1X,I3,2X,3(1PE11.4,1X))
      WRITE(IWP,50)
 50   FORMAT(5X,'CORRELATION MATRIX')
      NVR=NPT*(NPT+1)/2.0
      M=0
      N=1
      DO 80 I=1,NVR
      M=M+1
      IF(XS(M).EQ.0.0.OR.XS(N).EQ.0.0.OR.STD(M).EQ.0.0.OR.STD(N)
     $        .EQ.0.0)GO TO 60
      ICOR(I)=IFIX(1.0E+6*VAR(I)/(XS(M)*STD(M)*XS(N)*STD(N))+0.5)
      GO TO 70
 60   ICOR(I)=0.0
 70   IF(N.NE.M)GO TO 80
      N=N+1
      M=0
 80   CONTINUE
      IRMIN=0
      IRMAX=0
      DO 90 I=1,NPT
      IRMIN=IRMAX+1
      IRMAX=IRMAX+I
 90   WRITE(IWP,100)I,(ICOR(IR),IR=IRMIN,IRMAX)
 100  FORMAT(1X,I3,1X,28I4,1X/(20X,25I4))
      RETURN
      END
C
      SUBROUTINE WRILB5(NPTHN,EN,XS,VAR,MAT,MT,ZA,AWR,ENF)
C ** THIS SUBROUTINE WRITES THE COVARIANCE OF THE DIAGONAL BLOCKS OF MATRIX
C ** M' IN ENDF/B FORMAT (FILE 33) USING THE LB=5 FLAG.
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
      DIMENSION EN(1),XS(1),VAR(1),ENF(1)
      DATA IWR/17/
C ** DATA ARE WRITTEN IN FILES WITH UNIT #'S IWR (=18,19,...,(17+NEVSET)).
      IWR=IWR+1
      NPT=NPTHN
      NVR=(NPT-1)*NPT/2.0
      M=0
      N=1
      DO 20 I=1,NVR
```

```
      M=M+1
      IF(XS(M).EQ.0.0.OR.XS(N).EQ.0.0)GO TO 10
C ** CALCULATE RELATIVE COVARIANCE MATRIX COMPONENTS (LB=5)
      VAR(I)=VAR(I)/(XS(M)*XS(N))
 10   IF(N.NE.M)GO TO 20
      N=N+1
      M=0
 20   CONTINUE
C ** SET 1ST ENERGY TO 1.0E-5 & STORE ENERGIES IN ENF ARRAY
      ENE=EN(1)
      ENF(1)=1.0E-5
      DO 30 I=2,NPT
      ENF(I)=ENE
 30   ENE=EN(I)
      NPT=NPT+1
      ENF(NPT)=ENE
C ** IN THE FOLLOWING, ENF REPRESENTS F, THE FRACTIONAL COMPONENT
C ** CORRELATED OVER THE ENERGY INTERVAL. THE CALCULATIONS ARE STORED
C ** AFTER THE ENERGIES IN ARRAY ENF.
      IR=NPT-1
      IJ=0
      ENF(2*NPT-1)=0.0
      NPTM2=NPT-2
      DO 50 I=1,NPTM2
      ENF(I+NPT)=0.0
      IJPI=IJ+I
      IJ=IJPI
      J=I
      IL=IR+1
      IR=IL+NPT-I-2
      DO 40 II=IL,IR
C ** VAR IS LOWER TRIANGULAR; ENF MUST CONTAIN THE UPPER TRIANGULAR VALUES
      ENF(II+NPT)=VAR(IJPI)
      IJPI=IJPI+J
 40   J=J+1
 50   CONTINUE
      NTOT=NPT+IR
      LO=0
      L1=1
      LB=5
      DUM=0.0
      NL=NEVSET
      MF=33
      WRITE(IWR,70)ZA,AWR,LO,LO,LO,NL,MAT,MF,MT
 70   FORMAT(1P2E11.4,4I11,I4,I2,I3)
      WRITE(IWR,70)DUM,DUM,LO,MT,LO,L1,MAT,MF,MT
      WRITE(IWR,70)DUM,DUM,L1,LB,NTOT,NPT,MAT,MF,MT
      DO 90 I=1,NTOT,6
      IP5=I+5
      IF(IP5.LE.NTOT)GO TO 90
      DO 80 J=NTOT+1,IP5
 80   ENF(J)=0.0
 90   WRITE(IWR,100)(ENF(K),K=I,IP5),MAT,MF,MT
 100  FORMAT(3(1PE11.4,E11.4),I4,I2,I3)
```

```
      IF(NEVSET.NE.1)RETURN
      WRITE(IWR,70)DUM,DUM,L0,L0,L0,L0,MAT,MF,L0
      WRITE(IWR,70)DUM,DUM,L0,L0,L0,L0,MAT,L0,L0
      RETURN
      END
C
      SUBROUTINE PARREC
C ** PARREC SETS UP THE DIMENSIONS (OR ARRAYS) NEEDED FOR WRITING THE
C ** COVARIANCE OF THE OFF-DIAGONAL BLOCKS OF MATRIX M' INTO ENDF/B FORMAT
C ** (FILE 33) USING THE LB=6 FLAG. SUBROUTINE THNREC IS CALLED
C ** TO READ AND THIN THE COVARIANCE.
      COMMON/PARRL/NSIZER,A(1)
      COMMON/PARIN/NSIZEI,IA(1)
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
      COMMON/WHERE/INEVP,IEEV,IXSNEW,IMAT,IMT,IQ,IZA,IAWR,ISTD,IVARNW,
     $                  IVARBK,IIJTHN,IEEVTN,IXSTHN,ISTDTN,IICOR,
     $                  INPTHN,IENF
      IRV=16
C ** REWIND FILE CONTAINING COVARIANCE MATRIX M'.
      REWIND IRV
      CALL THNREC(IA(INEVP),A(IEEV),A(IVARNW),A(IVARBK),A(IEEVTN),
     $          A(IXSTHN),A(ISTDTN),IA(IICOR),IA(INPTHN),IA(IMAT),
     $          IA(IMT),A(IZA),A(IAWR),A(IENF))
C ** A(IENF) WILL BE STORAGE USED FOR CALCULATING & WRITING
C ** FINAL ENERGIES E AND COMPONENT VALUES F IN LB = 6 FORMAT (SUBROUTINE
C ** WRILB6).
      RETURN
      END
C
      SUBROUTINE THNREC(NEVP,EEV,VARNEW,VARBLK,EEVTHN,XSTHN,STDTHN,
     $          ICOR,NPTHN,MAT,MT,ZA,AWR,ENF)
C ** THNREC READS THE COVARIANCE MATRIX M' (OR VARNEW) AND SETS VARBLK TO
C ** THE OFF-DIAGONAL BLOCKS OF M' (ONE BLOCK AT A TIME). IT THINS
C ** VARBLK, CALLS PNTREC TO PRINT THE CORRELATION MATRIX OF THE THINNED
C ** OFF-DIAGONAL BLOCKS (VARBLK) AND CALLS WRILB6 TO WRITE VARBLK IN
C ** ENDF/B FORMAT (FILE 33) USING THE LB=6 FLAG.
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
      DIMENSION NEVP(1),EEV(1),VARNEW(NEVPTS,1),VARBLK(1),
     $          EEVTHN(NLGEVS,1),XSTHN(NLGEVS,1),STDTHN(NLGEVS,1),
     $          ICOR(1),NPTHN(1),MAT(1),MT(1),ZA(1),AWR(1),ENF(1)
C ** THE COVARIANCE IS READ FROM FILE IRV (=16) (OUTPUT FROM GLUCS.F4). THE
C ** COVARIANCE OF THE OFF-DIAGONAL RECTANGULAR BLOCKS OF M' ARE WRITTEN
C ** IN FILES WITH UNIT #'S IWT AND IWF (18, 19,...,(17+NEVSET)).
      IW=17
      IWT=17
      IWF=17
      IRV=16
      NEVM1=NEVSET-1
      NPSET=NEVP(1)
      NROW=0
      DO 140 NEVR=1,NEVM1
      IWT=IWT+1
      IW=IW+1
      IWR=IW
```

```
      IF(NEVR.EQ.1)GO TO 10
      NROW=NEVP(NEVR-1)
      NPSET=NEVP(NEVR)-NROW
10    DO 20 J=1,NPSET
20    READ(IRV,30)(VARNEW(I,J),I=1,NEVPTS)
30    FORMAT(1X,10(1PE11.4))
C ** SET VARBLK TO THINNED (RECTANGULAR) DATA BLOCK (NEVR,NEVC) OF MATRIX M'
      DO 130 NEVC=NEVR+1,NEVSET
      IWR=IWR+1
      IJ=0
      IR=1
      NINEVC=NEVP(NEVC-1)+1
      NFEVC=NEVP(NEVC)
      DO 100 J=1,NPSET
      IF(IR.LE.NPTHN(NEVR))GO TO 50
      TYPE 40,NEVR
40    FORMAT(1X,'IR > NPTHN(',I2,') - ERROR')
      STOP
50    IF(EEV(J+NROW).NE.EEVTHN(IR,NEVR))GO TO 100
      IR=IR+1
      IC=1
      DO 80 I=NINEVC,NFEVC
      IF(IC.LE.NPTHN(NEVC))GO TO 70
      TYPE 60,NEVC
60    FORMAT(1X,'IC > NPTHN(',I2,') - ERROR')
      STOP
70    IF(EEV(I).NE.EEVTHN(IC,NEVC))GO TO 80
      IJ=IJ+1
      VARBLK(IJ)=VARNEW(I,J)
      IC=IC+1
80    CONTINUE
      IF((IC-1).EQ.NPTHN(NEVC))GO TO 100
      TYPE 90,NEVC
90    FORMAT(1X,'IC .NE. NPTHN(',I2,') - ERROR')
      STOP
100   CONTINUE
      IF((IR-1).EQ.NPTHN(NEVR))GO TO 120
      TYPE 110,NEVR
110   FORMAT(1X,'IR.NE.NPTHN(',I2,') - ERROR')
      STOP
120   CONTINUE
      CALL PNTREC(NEVR,NEVC,NPTHN(NEVR),NPTHN(NEVC),XSTHN(1,NEVR),
     $          XSTHN(1,NEVC),STDTHN(1,NEVR),STDTHN(1,NEVC),
     $          VARBLK,ICOR)
      CALL WRILB6(IWR,IWT,NEVR,NEVC,NPTHN(NEVR),NPTHN(NEVC),
     $          EEVTHN(1,NEVR),XSTHN(1,NEVR),EEVTHN(1,NEVC),
     $          XSTHN(1,NEVC),VARBLK,
     $          MAT(NEVR),MAT(NEVC),MT(NEVR),MT(NEVC),ZA(NEVR),
     $          ZA(NEVC),AWR(NEVR),AWR(NEVC),ENF)
130   CONTINUE
140   CONTINUE
      LO=0
      DUM=0.0
      DO 150 I=1,NEVSET
```

```
      IWF=IWF+1
 150  WRITE(IWF,160)DUM,DUM,LO,LO,LO,LO,MAT(I),LO,LO
 160  FORMAT(1P2E11.4,4I11,I4,I2,I3)
      RETURN
      END
C
      SUBROUTINE PNTREC(NEVR,NEVC,NROW,NCOL,XSR,XSC,STDR,STDC,
     $          VAR,ICOR)
C ** PNTREC PRINTS THE CORRELATION ELEMENTS (CALCULATED FROM COVARIANCE VAR)
C ** FOR EACH OF THE RECTANGULAR BLOCKS OF MATRIX M' (UNIT # IWP (=17)).
      DIMENSION XSR(1),XSC(1),STDR(1),STDC(1),VAR(1),ICOR(1)
      IWP=17
      WRITE(IWP,10)NEVC,NEVR
 10   FORMAT(1X,'FINAL THINNED CORRELATION MATRIX OF RECTANGULAR
     $ BLOCK ',I2,I2)
      WRITE(IWP,20)NEVR,(I,I=1,NROW)
 20   FORMAT(2X,'SET',I2,28I4,1X/(19X,25I4))
      WRITE(IWP,30)NEVC
 30   FORMAT(1X,'SET',I2)
      IJ=0
      DO 80 I=1,NROW
      XSSTDR=XSR(I)*STDR(I)
      IF(XSSTDR.EQ.0.0)GO TO 60
      DO 50 J=1,NCOL
      XSSTDC=XSC(J)*STDC(J)
      IJ=IJ+1
      IF(XSSTDC.EQ.0.0)GO TO 40
      ICOR(IJ)=IFIX(1.0E+6*VAR(IJ)/(XSSTDC*XSSTDR)+0.5)
      GO TO 50
 40   ICOR(IJ)=0.0
 50   CONTINUE
      GO TO 80
 60   IJ=IJ+NCOL
      DO 70 J=1,NCOL
 70   ICOR(J)=0.0
 80   CONTINUE
C ** TO BE CONSISTENT W/ UTILITY INPUT PROGRAM, WRITE TRANSPOSE OF ICOR(IJ)
      NPTS=NCOL*NROW
      DO 90 I=1,NCOL
 90   WRITE(IWP,100)I,(ICOR(J),J=I,NPTS,NCOL)
 100  FORMAT(1X,I3,4X,28I4,1X/(20X,25I4))
      RETURN
      END
C
      SUBROUTINE WRILB6(IWR,IWT,NEVR,NEVC,NR,NC,ENR,XSR,ENC,XSC,VAR,
     $          MATR,MATC,MTR,MTC,ZAR,ZAC,AWRR,AWRC,ENF)
C ** THIS SUBROUTINE WRITES THE RECTANGULAR BLOCKS OF MATRIX M' INTO
C ** ENDF/B FORMAT (FILE 33) USING THE LB=6 FLAG. WRILB6 TAKES ADVANTAGE
C ** OF THE SYMMETRY OF THE COVARIANCE MATRIX M' (VAR HERE), FIRST WRITING
C ** THE CURRENT RECTANGULAR BLOCK OF THE MATRIX IN A FILE WITH UNIT
C ** # IWR AND THEN WRITING ITS TRANSPOSE IN A FILE WITH UNIT # IWT.
C ** THESE UNIT #'S WILL BE 18,OR 19,...,OR (17+NEVSET).
      COMMON/NUMEV/NEVSET,NEVPTS,NLGEVS
      DIMENSION ENR(1),XSR(1),ENC(1),XSC(1),VAR(1),ENF(1)
```

```
      NCP1=NC+1
      NRP1=NR+1
C ** 1ST ENERGY IN AN E TABLE MUST BE 1.0E-5
      ENF(1)=1.0E-5
      ENF(NCP1+1)=1.0E-5
      DO 10 I=1,NC
      J=I+1
 10   ENF(J)=ENC(I)
      J=J+1
      DO 20 I=1,NR
      J=J+1
 20   ENF(J)=ENR(I)
      NRPNC=J
      MF=33
      LB=6
      L0=0
      L1=1
      DUM=0.0
      WRITE(IWR,30)DUM,DUM,MATR,MTR,L0,L1,MATC,MF,MTC
 30   FORMAT(1P2E11.4,4I11,I4,I2,I3)
C ** IT IS ASSUMED HERE THAT THE LAST ENERGY IN EACH ENDF EVALUATED
C ** SET IS 20.0E+6 (ENDF LAW)
      IF(ENR(NR).NE.20.E+6.OR.ENC(NC).NE.20.E+6)TYPE 40
 40   FORMAT(1X,'ERROR - FINAL ENERGY .NE. 20.0 MEV')
      WRITE(IWR,30)DUM,DUM,L0,LB,NCP1,NRP1,MATC,MF,MTC
      MM=0
      M=NRPNC
      DO 60 I=1,NR
      IF(I.GT.2)MM=MM+1
      DO 60 J=1,NC
      M=M+1
      IF(I.EQ.1.OR.J.EQ.1)GO TO 50
      MM=MM+1
      IF(XSR(I-1).EQ.0.0.OR.XSC(J-1).EQ.0.0)GO TO 50
      ENF(M)=VAR(MM)/(XSR(I-1)*XSC(J-1))
      GO TO 60
 50   ENF(M)=0.0
 60   CONTINUE
      MM=0
      DO 70 I=NRPNC+1,M
      MM=MM+1
 70   VAR(MM)=ENF(I)
      DO 90 I=1,M,6
      IP5=I+5
      IF(IP5.LE.M)GO TO 90
      DO 80 J=M+1,IP5
 80   ENF(J)=0.0
 90   WRITE(IWR,100)(ENF(K),K=I,IP5),MATC,MF,MTC
 100  FORMAT(6(1PE11.4,E11.4),I4,I2,I3)
      NEVM1=NEVSET-1
      IF(NEVR.EQ.NEVM1)WRITE(IWR,30)DUM,DUM,L0,L0,L0,L0,MATC,MF,L0
C ** NOW WRITE TRANSPOSE OF ABOVE
      WRITE(IWT,30)DUM,DUM,MATC,MTC,L0,L1,MATR,MF,MTR
      ENF(NRP1+1)=1.0E-5
```

```
      DO 110 I=1,NR
      J=I+1
110   ENF(J)=ENR(I)
      J=J+1
      DO 120 I=1,NC
      J=J+1
120   ENF(J)=ENC(I)
      M=J
      NRNC=NR*NC
      WRITE(IWT,30)DUM,DUM,LO,LB,NRP1,NCP1,MATR,MF,MTR
      DO 130 J=1,NC
      DO 130 I=J,NRNC,NC
      M=M+1
      ENF(M)=VAR(I)
130   CONTINUE
      DO 150 I=1,M,6
      IP5=I+5
      IF(IP5.LE.M)GO TO 150
      DO 140 J=M+1,IP5
140   ENF(J)=0.0
150   WRITE(IWT,100)(ENF(K),K=I,IP5),MATR,MF,MTR
      IF(NEVC.EQ.NEVSET)WRITE(IWT,30)DUM,DUM,LO,LO,LO,LO,MATR,MF,LO
      RETURN
      END
```

Appendix B

Teletype Output for Sample Problem Execution

In the following, underlined quantities are typed in by the user. The symbol ⤸ means a carriage return.

EXECUTION OF PROGRAM INPUT.F4
(FIRST CYCLE)

```
EX INPUT.F4 ⤸
FØRTRAN:  INPUT
MAIN.
PRNTRT
PRNT
THNLIN
LINK:  LØADING
[LNKXCT INPUT EXECUTIØN]
# EVAL SETS - (0.LT.#.LE.10) 2 ⤸
IF WILL BE READING CRØSS-REACTIØN CØVARIANCE FILE(S), TYPE 1 ⤸
INPUT FILE  1 = FE56.DAT ⤸
MT = 103 ⤸
TØ THIN #  1 SET, TYPE 1 1 ⤸
EPS (0.<EPS<1.0) = .9 ⤸
INPUT FILE 2 = CU65.DAT ⤸
MT = 16 ⤸
TO THIN # 2 SET, TYPE 1 1 ⤸
EPS (0.<EPS<1.0) = .9 ⤸
STØP

END ØF EXECUTIØN
CPU TIME: 27.78 ELAPSED TIME: 1:34.67
EXIT
```

EXECUTION OF PROGRAM GLUCS.F4
(FIRST CYCLE)

```
EX GLUCS.F4 ⤸
FØRTRAN:  GLUCS
MAIN.
IDIMR
IDIMI
PAREXP
PAREX
PAREVL
PAREV
GTHY
GTINT
VAREVL
VAREV
MULMG
PARNEW
```

```
PARNE
MULGMG
XSVNEW
SSPFA
ISAMAX
SASUM
SAXPY
SDØT
SSCAL
SSWAP
SSPSL
SSPCØ
LINK:  LØADING
[LNKXCT GLUCS EXECUTIØN]
STØP

END ØF EXECUTIØN
CPU TIME:  18.33 ELAPSED TIME:  22.05
```

EXECUTION OF PROGRAM ØUTPUT.F4
(FIRST CYCLE)

```
EX ØUTPUT.F4 ♪
FØRTRAN:  ØUTPUT
MAIN.
IDIMR
IDIMI
PAREXS
RDMEXS
WRIEXS
PARDIA
PARDI
CØRBLK
THNDIA
PNTDIA
WRILB5
PARREC
THNREC
PNTREC
WRILB3
LINK:  LØADING
[LNKXCT ØUTPUT EXECUTIØN]
STØP

END ØF EXECUTIØN
CPU TIME:  12.68 ELAPSED TIME:  14.43
EXIT
```

```
DEL FØR12.DAT ↵
FILES DELETED:
FØR12.DAT
16 BLØCKS FREED

REN FØR??.ØNE=FØR??.DAT ↵
FILES RENAMED:
FØRO1.DAT
FØR11.DAT
FØRO3.DAT
FØR10.DAT
FØRO2.DAT
FØRO4.DAT
FØR14.DAT
FØR15.DAT
FØR16.DAT
FØR18.DAT
FØR19.DAT
FØR17.DAT

REN FØRO1.DAT=FØRO1.TWØ ↵
FILES RENAMED:
FØRO1.TWØ
```

EXECUTION OF PROGRAM INPUT.F4
(SECOND CYCLE)

```
EX INPUT.F4 ↵
LINK:  LØADING
[LNKXCT INPUT EXECUTIØN]
# EVAL SETS = (0.LT.#.LE.10) 3 ↵
IF WILL BE READING CRØSS-REACTIØN CØVARIANCE FILE(S), TYPE 1 1 ↵
INPUT FILE 1 = FØR18.ONE ↵
MT = 103 ↵
TØ THIN #  1 SET, TYPE 1 ↵
INPUT FILE 2 = FØR19.ONE ↵
MT = 16 ↵
TØ THIN #  2 SET, TYPE 1 ↵
INPUT FILE  3 = S32.DAT ↵
MT = 103 ↵
TØ THIN #  3 SET, TYPE 1 1 ↵
EPS (0.<EPS<1.0) = .9 ↵
IF READING CRØSS-REACTIØN CØVARIANCE FILE FØR SETS 2 AND 1 TYPE 1 1 ↵
INPUT FILE (CRØSS-REACTIØN CØVARIANCE) = FØR18.ONE ↵
TYPE MAT1,MT1 : 651 16 ↵
```

```
IF READING CRØSS-REACTIØN CØVARIANCE FILE FØR SETS  3 AND  1 TYPE 1 ↵
IF READING CRØSS-REACTIØN CØVARIANCE FILE FØR SETS  3 AND  2 TYPE 1 ↵
STØP

END ØF EXECUTIØN
CPU TIME:  1:30.18        ELAPSED TIME:  6:57.13
EXIT
```

EXECUTION OF PROGRAM GLUCS.F4
(SECOND CYCLE)

```
EX GLUCS.F4 ↵
LINK:  LOADING
[LNKXCT GLUCS EXECUTIØN]
TYPE FIXED ENERGY (MEV)- (EXP. SET  3 EVAL. SET  1)  14.5 ↵
TYPE FIXED ENERGY (MEV)- (EXP. SET  4 EVAL. SET  2)  14.5 ↵
STØP

END ØF EXECUTIØN
CPU TIME:  1:41.18        ELAPSED TIME:  5:51.37
EXIT
```

EXECUTION OF PROGRAM ØUTPUT.F4
(SECOND CYCLE)

```
EX ØUTPUT.F4 ↵
LINK:  LØADING
[LNKXCT ØUTPUT EXECUTIØN]
STØP

END ØF EXECUTIØN
CPU TIME:  33.48          ELAPSED TIME:  1:30.73
EXIT
```

Appendix C

Input Data for the Model Programs

Note that part of the input data for one program may be the output from the previous program.

```
THIS IS FE56.DAT (INPUT FOR INPUT.F4).
2.6056E+04 5.5454E+01          0          0          0          1 561 1451
0.0        0.0                 0          0          0          0 561 1451
0.0        0.0                 0          0         39          4 561 1451
26-FE-56  ORNL        EVAL-JUL78   C.Y.FU                          561 1451
    EXPERIMENTAL DATA AVAILABLE SINCE THE EVALUATION OF FE-56     561 1451
(N,P) CROSS SECTIONS BY DUDEY AND KENNERLEY(1) WERE REVIEWED.     561 1451
SMITH AND MEADOWS(2) REPORTED NEW DATA FROM 4 TO 10 MEV, AN ENERGY 561 1451
RANGE MOST IMPORTANT FOR APPLICATION OF THIS REACTION TO FISSION  561 1451
REACTOR DOSIMETRY.  VONACH(3) HAS RECENTLY CALLED FOR ATTENTION   561 1451
TO HIS DATA(4) FROM 13 TO 15 MEV.  FURTHER CHECKS(3) RESULTED     561 1451
IN A REDUCTION OF THE DATA ERRORS FROM 4% TO 2.5%.  THE DATA HAD  561 1451
NOT BEEN USED IN THE DUDEY AND KENNERLEY EVALUATION BUT ARE       561 1451
2 TO 3% LOWER THAN THE EVALUATED VALUES.  THREE SINGLE-ENERGY     561 1451
MEASUREMENTS(5-7) NEAR 14.6 MEV HAVE BEEN REPORTED.  THE ONE      561 1451
BY DYER(5) IS 1% HIGHER THAN THE ENDF/B-IV VALUE, ROBERTSON 7%    561 1451
HIGHER, AND QAIM 4% LOWER.                                        561 1451
    INCLUSION OF THE NEW DATA IN A SUBJECTIVE ANALYSIS RESULTS    561 1451
IN UP TO 10% REDUCTION FROM THE ENDF/B-IV VALUES FROM THRESHOLD TO 561 1451
6 MEV, 2-3% REDUCTION FORM 6 TO 10 MEV, AND 1-2% REDUCTION FROM   561 1451
13-15 MEV. THE DATA OF SMITH AND MEADOWS ARE RELATIVE TO ENDF/B-IV 561 1451
U-238(N,F) CROSS SECTIONS AND HAD BEEN RENORMALIZED TO ENDF/B-V   561 1451
VALUES BEFORE BEING USED IN THE PRESENT EVALUATION.  THE EFFECT   561 1451
ON THE CALCULATED FISSION SPECTRUM AVERAGE IS ESTIMATED TO BE     561 1451
A REDUCTION OF ABOUT 3% FROM ENDF/B-IV VALUE.                     561 1451
    ESTIMATED LONG RANGE CORRELATED ERROR IS 2% FROM THRESHOLD    561 1451
TO 20 MEV.  SHORT RANGE CORRELATED ERRORS ARE GIVEN FOR EVERY 2   561 1451
MEV INERVAL.  THESE ARE 7% UP TO 6 MEV, 2% PLUS 0.5 MB FROM       561 1451
6 TO 20 MEV.                                                      561 1451
    REFERENCES:                                                   561 1451
 1. N. D. DUDEY AND R. KENNERLEY, "EVALUATION OF FE-56(N,P)MN-56  561 1451
    CROSS SECTIONS FOR ENDF/B-III," IN ENDF/B-IV DOSIMETRY FILE,  561 1451
    EDITED BY B. A. MAGURNO, BNL-NCS-50446(1975).                 561 1451
 2. D. L. SMITH AND J. W. MEADOWS, NUCL. SCI. ENG. 58,314(1975).  561 1451
 3. H. VONACH,"STATUS OF SOME ACTIVATION CROSS SECTIONS FOR       561 1451
    REACTOR NEUTRON DOSIMETRY IN THE RANGE 13-15 MEV," P. 361,    561 1451
    IAEA-208(1975).                                               561 1451
 4. H. VONACH ET AL., CONF. NUCL. CROSS SECTIONS AND TECH.,       561 1451
    WASHINGTON, NBS-SP-299, VOL. 1, P. 885(1968).                 561 1451
 5. N. C. DYER AND J. H. HAMILTON, J. INORG. NUCL. CHEM. 34, 1119 561 1451
    (1972).                                                       561 1451
 6. J. C. ROBERTSON ET AL., J. NUCL. EN. 27, 139(1972).           561 1451
 7. S. M. QAIM ET AL., CONF. ON CHEM. NUCL. DATA, CANTERBURY,     561 1451
    P. 121(1971).                                                 561 1451
0.0        0.0                 1        451         46          1 561 1451
0.0        0.0                 2        151          4          1 561 1451
0.0        0.0                 3        103         19          1 561 1451
0.0        0.0                33        103         10          1 561 1451
0.0        0.0                 0          0          0          0 561 1  0
0.0        0.0                 0          0          0          0 561 0  0
2.6056E+04 5.5454E+01          0          0          1          0 561 2151
2.6056E+04 1.0                 0          0          1          0 561 2151
1.0000E-05 2.0000E+07          0          0          0          0 561 2151
0.0        5.0000E-01          0          0          0          0 561 2151
```

```
 0.0        0.0                 0          0          0          0 561 2  0
 0.0        0.0                 0          0          0          0 561 0  0
 2.6056E+04 5.5454E+01          0         99          0          0 561 3103
 0.0000E-01-2.9130E+06          0          0          3         47 561 3103
          2          2         12          4         47          2 561 3103
2.96550E+06 0.0000E-013.00000E+06 7.0000E-094.00000E+06 6.0000E-06 561 3103
4.25000E+06 3.2000E-054.50000E+06 1.4000E-044.60000E+06 2.4000E-04 561 3103
4.80000E+06 5.6500E-045.00000E+06 1.1000E-035.25000E+06 2.5000E-03 561 3103
5.50000E+06 4.7000E-035.75000E+06 8.0000E-036.00000E+06 1.2500E-02 561 3103
6.20000E+06 1.5900E-026.40000E+06 1.9200E-026.60000E+06 2.2400E-02 561 3103
6.80000E+06 2.5600E-027.00000E+06 2.8700E-027.50000E+06 3.6100E-02 561 3103
8.00000E+06 4.2700E-028.50000E+06 4.8900E-029.00000E+06 5.5000E-02 561 3103
9.50000E+06 6.1400E-021.00000E+07 6.9300E-021.05000E+07 7.7500E-02 561 3103
1.10000E+07 8.7000E-021.15000E+07 9.5600E-021.20000E+07 1.0300E-01 561 3103
1.22000E+07 1.0600E-011.24000E+07 1.0800E-011.26000E+07 1.1000E-01 561 3103
1.28000E+07 1.1100E-011.30000E+07 1.1200E-011.34000E+07 1.1250E-01 561 3103
1.38000E+07 1.1140E-011.42000E+07 1.0880E-011.46000E+07 1.0500E-01 561 3103
1.50000E+07 9.9400E-021.55000E+07 9.0700E-021.60000E+07 8.1800E-02 561 3103
1.65000E+07 7.6200E-021.70000E+07 6.9200E-021.75000E+07 6.3500E-02 561 3103
1.80000E+07 5.8900E-021.85000E+07 5.4700E-021.90000E+07 5.1300E-02 561 3103
1.95000E+07 4.9200E-022.00000E+07 4.8800E-020.00000E-01 0.0000E-01 561 3103
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 561 3  0
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 561 0  0
 2.6056E+04 5.5454E+01          0          0          0          1 56133103
 0.0000E-01 0.0000E-01          0        103          0          1 56133103
 0.0000E-01 0.0000E-01          1          5        300         24 56133103
1.00000E-05 2.9655E+063.00000E+06 4.5000E+065.00000E+06 5.5000E+06 56133103
6.00000E+06 6.4000E+067.00000E+06 8.0000E+069.00000E+06 1.0000E+07 56133103
1.10000E+07 1.2000E+071.24000E+07 1.2800E+071.34000E+07 1.4200E+07 56133103
1.50000E+07 1.6000E+071.70000E+07 1.8000E+071.90000E+07 2.0000E+07 56133103
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 56133103
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 56133103
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 56133103
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 56133103
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 56133103
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 56133103
0.00000E-01 0.0000E-010.00000E-01 0.0000E-010.00000E-01 0.0000E-01 56133103
0.00000E-01 0.0000E-010.00000E-01 1.0000E-022.25000E-03 1.2000E-03 56133103
7.35000E-04 5.0400E-044.03200E-04 2.6880E-041.72032E-04 1.3763E-04 56133103
1.10100E-04 6.6060E-055.28482E-05 4.2279E-052.81857E-05 1.8039E-05 56133103
1.62350E-05 1.4611E-051.31503E-05 1.1835E-051.33147E-05 1.4380E-05 56133103
8.10000E-03 2.1600E-031.32300E-03 9.0720E-047.25760E-04 4.8384E-04 56133103
3.09658E-04 2.4773E-041.98181E-04 1.1891E-049.51268E-05 7.6101E-05 56133103
5.07343E-05 3.2470E-052.92230E-05 2.6301E-052.36706E-05 2.1304E-05 56133103
2.39665E-05 2.5884E-056.40000E-03 2.3520E-031.61280E-03 1.2902E-03 56133103
8.60160E-04 5.5050E-044.40402E-04 3.5232E-042.11393E-04 1.6911E-04 56133103
1.35291E-04 9.0194E-055.77244E-05 5.1952E-054.67567E-05 4.2081E-05 56133103
3.78730E-05 4.2607E-054.60156E-05 4.9000E-032.35200E-03 1.8816E-03 56133103
1.25440E-03 8.0282E-046.42253E-04 5.1380E-043.08281E-04 2.4663E-04 56133103
1.97300E-04 1.3153E-048.41814E-05 7.5763E-056.81869E-05 6.1368E-05 56133103
5.52314E-05 6.2135E-056.71061E-05 3.6000E-032.30400E-03 1.5360E-03 56133103
9.83040E-04 7.8643E-046.29146E-04 3.7749E-043.01990E-04 2.4159E-04 56133103
1.61061E-04 1.0308E-049.27713E-05 8.3494E-057.51447E-05 6.7630E-05 56133103
7.60841E-05 8.2171E-053.60000E-03 1.9200E-031.22880E-03 9.8304E-04 56133103
```

```
7.86432E-04 4.7186E-043.77487E-04 3.0199E-042.01327E-04 1.2885E-04 56133103
1.15964E-04 1.0437E-049.39309E-05 8.4538E-059.51051E-05 1.0271E-04 56133103
2.50000E-03 1.2800E-031.02400E-03 8.1920E-044.91520E-04 3.9322E-04 56133103
3.14573E-04 2.0972E-041.34218E-04 1.2080E-041.08716E-04 9.7845E-05 56133103
8.80602E-05 9.9068E-051.06993E-04 1.6000E-031.02400E-03 8.1920E-04 56133103
4.91520E-04 3.9322E-043.14573E-04 2.0972E-041.34218E-04 1.2080E-04 56133103
1.08716E-04 9.7845E-058.80602E-05 9.9068E-051.06993E-04 1.6000E-03 56133103
1.02400E-03 6.1440E-044.91520E-04 3.9322E-042.62144E-04 1.6777E-04 56133103
1.50995E-04 1.3590E-041.22306E-04 1.1008E-041.23835E-04 1.3374E-04 56133103
1.60000E-03 7.6800E-046.14400E-04 4.9152E-043.27680E-04 2.0972E-04 56133103
1.88744E-04 1.6987E-041.52882E-04 1.3759E-041.54793E-04 1.6718E-04 56133103
9.00000E-04 5.7600E-044.60800E-04 3.0720E-041.96608E-04 1.7695E-04 56133103
1.59252E-04 1.4333E-041.28995E-04 1.4512E-041.56728E-04 9.0000E-04 56133103
5.76000E-04 3.8400E-042.45760E-04 2.2118E-041.99066E-04 1.7916E-04 56133103
1.61243E-04 1.8140E-041.95910E-04 9.0000E-044.80000E-04 3.0720E-04 56133103
2.76480E-04 2.4883E-042.23949E-04 2.0155E-042.26748E-04 2.4489E-04 56133103
6.25000E-04 3.2000E-042.88000E-04 2.5920E-042.33280E-04 2.0995E-04 56133103
2.36196E-04 2.5509E-044.00000E-04 2.8800E-042.59200E-04 2.3328E-04 56133103
2.09952E-04 2.3620E-042.55092E-04 4.0000E-043.24000E-04 2.9160E-04 56133103
2.62440E-04 2.9525E-043.18865E-04 4.0000E-043.24000E-04 2.9160E-04 56133103
3.28050E-04 3.5429E-044.00000E-04 3.2400E-043.64500E-04 3.9366E-04 56133103
4.00000E-04 4.0500E-044.37400E-04 6.2500E-046.07500E-04 9.0000E-04 56133103
 0.0000E-01 0.0000E-01          0          0          0          0 56133  0
 0.0000E-01 0.0000E-01          0          0          0          0 561 0  0
 0.0000E-01 0.0000E-01          0          0          0          0   0 0  0
 0.0000E-01 0.0000E-01          0          0          0          0  -1 0  0
```

THIS IS THE FOR01.DAT THAT WAS INPUT FOR GLUCS.F4 FOR THE FIRST RUN.

```
 16  2
  6  1  0
   14.6700      0.1081      1.0607
   15.3000      0.1000      1.8426
   16.5500      0.0801      1.4349
   17.0000      0.0721      1.3903
   17.8600      0.0631      1.3042
   18.9500      0.0532      1.5369
 16  2  1
   14.6700      8.8300      1.0803
   15.3000      9.9300      2.4825
   15.5200     10.5200      1.5166
   15.9000     11.4600      1.6435
   16.3500     12.4800      2.0130
   16.4600     12.9700      1.6453
   16.5500     13.3500      1.3813
   17.0000     14.7300      1.2849
   17.8600     17.1600      1.1996
   18.9500     19.2400      0.9762
  0
100  42 100  54  61 100  56  59  84 100  59  58  82  80 100  50  53  76  73  72
100 -19   2   2   2   3   2 100   1 -80 -32 -28 -25 -25  14 100   2 -30 -49 -44
-48 -39  23  44 100   2 -32 -51 -46 -40 -41  22  45  71 100   2 -24 -38 -35 -30
-31  18  35  54  55 100   2 -28 -45 -40 -36 -36  21  41  64  66  50 100   2 -32
-52 -47 -41 -42  26  48  75  77  59  70 100   3 -30 -49 -44 -39 -39  28  47  73
 75  57  68  79 100   3 -28 -45 -40 -35 -36  29  45  70  71  55  65  76  75 100
  3 -18 -29 -26 -22 -23  36  37  59  58  46  55  64  64  64 100
```

THIS IS FOR02.DAT. THESE DATA WERE OUTPUT FROM INPUT.F4 (DURING THE FIRST RUN) AND USED AS INPUT FOR BOTH GLUCS.F4 AND OUTPUT.F4.

41 2 25

THIS IS FOR03.DAT. THESE DATA WERE OUTPUT FROM INPUT.F4 (DURING THE FIRST RUN) AND USED AS INPUT FOR GLUCS.F4.

```
25 25  3
 2  2  8  4 25  2
2.9655E+06  0.0000E-01
3.0000E+06  7.0000E-09
4.0000E+06  6.0000E-06
4.2500E+06  3.2000E-05
4.5000E+06  1.4000E-04
5.0000E+06  1.1000E-03
5.5000E+06  4.7000E-03
6.0000E+06  1.2500E-02
6.4000E+06  1.9200E-02
7.0000E+06  2.8700E-02
8.0000E+06  4.2700E-02
9.0000E+06  5.5000E-02
1.0000E+07  6.9300E-02
1.1000E+07  8.7000E-02
1.2000E+07  1.0300E-01
1.2400E+07  1.0800E-01
1.2800E+07  1.1100E-01
1.3400E+07  1.1250E-01
1.4200E+07  1.0880E-01
1.5000E+07  9.9400E-02
1.6000E+07  8.1800E-02
1.7000E+07  6.9200E-02
1.8000E+07  5.8900E-02
1.9000E+07  5.1300E-02
2.0000E+07  4.8800E-02
16 41  1
16  2
1.0063E+07  0.0000E-01
1.0500E+07  3.4580E-02
1.1000E+07  1.5440E-01
1.1500E+07  3.1340E-01
1.2000E+07  4.7540E-01
1.2500E+07  6.0000E-01
1.3000E+07  6.9800E-01
1.3500E+07  7.8300E-01
1.4500E+07  9.0800E-01
1.5500E+07  9.9100E-01
1.6500E+07  1.0470E+00
1.7500E+07  1.0810E+00
1.8500E+07  1.1050E+00
1.9200E+07  1.1030E+00
1.9600E+07  1.0560E+00
2.0000E+07  9.7960E-01
```

THIS IS FOR04.DAT. THESE DATA WERE OUTPUT FROM INPUT.F4 (DURING THE FIRST RUN) AND USED AS INPUT FOR GLUCS.F4. TO SAVE SPACE, ONLY THE FIRST TWO AND LAST TWO ROWS OF THE MATRIX M ARE GIVEN HERE.

```
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01
 0.0000E-01 4.9000E-19 4.2000E-16 2.2400E-15 2.2050E-15 9.2400E-15 2.4181E-14 4.4100E-14 5.4190E-14 5.4002E-14
5.1420E-14 5.2988E-14 5.3410E-14 4.0231E-14 3.8104E-14 3.1963E-14 2.1900E-14 1.4206E-14 1.2365E-14 1.0166E-14
7.5299E-15 5.7329E-15 5.4897E-15 5.1639E-15 4.9122E-15 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01
```

.
.
.
.
.
.

```
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 1.5392E-04 2.4545E-04 3.9630E-04 5.7251E-04
7.3599E-04 9.3401E-04 1.1426E-03 1.3947E-03 1.5722E-03 1.7484E-03 1.9002E-03 2.0446E-03 2.2677E-03 3.0153E-03
2.7972E-03
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 1.4278E-04 2.2769E-04 3.6763E-04 5.3109E-04
6.8274E-04 8.6643E-04 1.0599E-03 1.2938E-03 1.4584E-03 1.6219E-03 1.7627E-03 1.8967E-03 2.1036E-03 2.7972E-03
2.5948E-03
```

THIS IS FOR10.DAT. THESE DATA WERE OUTPUT FROM INPUT.F4 (DURING THE FIRST RUN) AND USED AS INPUT FOR OUTPUT.F4.

```
 561103 -2.9130E+06 2.6056E+04 5.5454E+01
 651 16 -9.9100E+06 2.9063E+04 6.4928E+01
```

THIS IS FOR15.DAT. THESE DATA WERE OUTPUT FROM GLUCS.F4 (DURING THE FIRST RUN) AND USED AS INPUT FOR OUTPUT.F4.

```
 25 25  3
  2  2  8  4 25  2
2.9655E+06  0.0000E-01
3.0000E+06  7.0107E-09
4.0000E+06  6.0092E-06
4.2500E+06  3.2049E-05
4.5000E+06  1.4039E-04
5.0000E+06  1.1054E-03
5.5000E+06  4.7336E-03
6.0000E+06  1.2609E-02
6.4000E+06  1.9410E-02
7.0000E+06  2.9027E-02
8.0000E+06  4.3187E-02
9.0000E+06  5.5784E-02
1.0000E+07  7.0535E-02
1.1000E+07  8.8454E-02
1.2000E+07  1.0515E-01
1.2400E+07  1.1082E-01
1.2800E+07  1.1402E-01
1.3400E+07  1.1556E-01
1.4200E+07  1.1250E-01
1.5000E+07  1.0319E-01
1.6000E+07  8.5812E-02
1.7000E+07  7.1227E-02
1.8000E+07  6.0431E-02
1.9000E+07  5.2429E-02
2.0000E+07  4.9874E-02
16 41  1
16  2
1.0063E+07  0.0000E-01
1.0500E+07  3.6601E-02
1.1000E+07  1.5762E-01
1.1500E+07  3.1860E-01
1.2000E+07  4.8292E-01
1.2500E+07  6.0966E-01
1.3000E+07  7.1026E-01
1.3500E+07  7.9800E-01
1.4500E+07  9.3112E-01
1.5500E+07  1.0026E+00
1.6500E+07  1.0456E+00
1.7500E+07  1.0769E+00
1.8500E+07  1.0599E+00
1.9200E+07  1.0088E+00
1.9600E+07  1.0003E+00
2.0000E+07  9.2790E-01
```

THIS IS FOR16.DAT. THESE DATA WERE OUTPUT FROM GLUCS.F4 (DURING THE FIRST RUN) AND USED AS INPUT FOR OUTPUT.F4. TO SAVE SPACE, ONLY THE FIRST TWO AND LAST TWO ROWS OF THE MATRIX M' ARE GIVEN HERE.

```
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01
0.0000E-01
 0.0000E-01 4.8998E-19 4.1998E-16 2.2399E-15 2.2041E-15 9.2280E-15 2.4106E-14 4.3856E-14 5.3721E-14 5.3271E-14
5.0333E-14 5.1238E-14 5.0654E-14 3.6987E-14 3.3303E-14 2.5671E-14 1.5163E-14 7.3785E-15 3.7155E-15 1.9856E-15
1.7404E-15 1.3698E-15 1.1740E-15 1.0215E-15 9.7160E-16 0.0000E-01 2.0388E-15 3.2511E-15 5.2492E-15 7.5836E-15
9.7483E-15 1.2371E-14 1.5135E-14 1.9010E-14 1.9677E-14 2.1021E-14 2.1021E-14 2.2175E-14 2.1622E-14 1.5902E-14
1.4751E-14

.
.
.
.
.
.

 0.0000E-01 1.5902E-14 1.3630E-11 7.2691E-11 5.7247E-10 7.9964E-09 4.9825E-08 1.6226E-07 3.1155E-07 4.8510E-07
7.2172E-07 1.1621E-06 1.8302E-06 2.1540E-06 3.1879E-06 4.1782E-06 4.4732E-06 4.5336E-06 5.4113E-06 5.8012E-06
5.3229E-06 4.5493E-06 3.9790E-06 5.7418E-06 5.4621E-06 0.0000E-01 8.0361E-06 1.2817E-05 2.0693E-05 2.9861E-05
3.8441E-05 4.8754E-05 5.9611E-05 7.4500E-05 7.6436E-05 8.2980E-05 1.0009E-04 7.9308E-05 2.1477E-04 1.5421E-03
1.4306E-03
 0.0000E-01 1.4751E-14 1.2644E-11 6.7432E-11 5.3105E-10 7.4179E-09 4.6221E-08 1.5052E-07 2.8901E-07 4.5000E-07
6.6951E-07 1.0780E-06 1.6978E-06 1.9982E-06 2.9572E-06 3.8759E-06 4.1496E-06 4.2056E-06 5.0198E-06 5.3814E-06
4.9379E-06 4.2202E-06 3.6914E-06 5.3264E-06 5.0670E-06 0.0000E-01 7.4517E-06 1.1890E-05 1.9200E-05 2.7705E-05
3.5663E-05 4.5228E-05 5.5272E-05 6.9117E-05 7.0899E-05 7.6978E-05 9.2834E-05 7.3600E-05 1.9921E-04 1.4306E-03
1.3271E-03
```

THIS IS THE FOR01.DAT THAT WAS INPUT FOR GLUCS.F4 FOR THE SECOND RUN.

```
 64  4
 25  1  3
    6.7000      0.0884      3.0000
    7.0100      0.0923      2.9000
    7.5000      0.1105      2.9000
    7.8000      0.1401      2.9000
    8.1000      0.1421      3.0000
    8.2700      0.1401      3.0000
    8.4000      0.1507      3.0000
    8.6100      0.1615      3.0000
    8.8100      0.1666      3.0000
    9.0200      0.1792      2.9000
    9.6300      0.1797      2.8000
   10.0300      0.1956      2.8000
   10.1300      0.1977      2.7000
   10.4000      0.2068      2.7000
   10.6200      0.2135      2.7000
   10.8100      0.2115      2.7000
   11.0000      0.2273      2.7000
   11.2200      0.2318      2.6000
   11.5000      0.2468      3.1000
   11.6100      0.2617      3.0000
   12.1000      0.2833      2.9000
   12.5900      0.3333      2.7000
   13.0800      0.3738      3.5000
   13.5800      0.4152      3.5000
   14.5000      0.4867      3.6000
 40  2  3
   10.1300      0.0178     57.0000
   10.4000      0.0455     11.0000
   10.5000      0.0907     14.0000
   10.6200      0.1930      6.5000
   10.8100      0.2130      5.9000
   11.0000      0.4210      4.2000
   11.2200      0.5180      6.8000
   11.5000      0.7840      5.9000
   11.6100      0.9770      5.3000
   12.0000      1.2460      5.3000
   12.1000      1.3610      5.1000
   12.6000      1.9550      4.6000
   13.0800      2.5410      6.5000
   13.5800      3.1160      6.9000
   14.5000      4.3050      3.1000
 52  1 -1
   12.5300      1.0090      5.4000
   13.5800      1.0820      3.4000
   13.8900      1.0730      3.4000
   14.2400      1.0000      3.6000
   14.6800      0.9820      3.7000
   14.7600      0.9730      3.7000
   16.1000      0.7490      5.5000
   16.6000      0.6930      5.6000
   17.5000      0.6130      5.9000
```

```
  18.5000     0.5130     5.5000
  19.1000     0.4810     5.7000
  19.8000     0.4390     5.4000
64  2 -2
  13.5800     0.8980     3.0000
  13.8900     0.93 0     3.0000
  14.2500     0.9690     3.0000
  14.6800     1.0080     3.1000
  14.7400     1.0300     3.0000
  16.3000     1.0430     5.9000
  16.5000     1.1190     5.5000
  16.6000     1.0500     5.9000
  17.9000     1.104      6.1000
  18.3000     1.1170     6.0000
  18.5000     1.1600     5.8000
  19.8000     1.0590     5.8000
 1
  9.0000    4.0000    8.4100    4.0000    4.0000    8.4100    4.0000    4.0000
  4.0000    8.4100    4.0000    4.0000    4.0000    4.0000    9.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    9.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    9.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    9.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    9.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    8.4100    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    7.8400    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    7.8400    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    7.2900    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  7.2900    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    7.2900
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    7.2900
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  7.2900    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    6.7600    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    9.6100    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    9.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    8.4100    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    7.2900    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000   12.2500    4.0000    4.0000    4.0000    4.0000
  4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
```

4.0000	4.0000	4.0000	12.2500	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	4.0000	12.9600	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	3249.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
4.0000	121.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	4.0000	4.0000	196.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	4.0000
4.0000	4.0000	42.2500	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	4.0000	4.0000	4.0000	4.0000
34.8100	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	4.0000	4.0000	4.0000	4.0000	4.0000	17.6400
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	46.2400
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000
34.8100	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	28.0900	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	4.0000	4.0000	28.0900	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	4.0000
4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000
4.0000	26.0100	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	4.0000	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	21.1600	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000

4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	4.0000	42.2500	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	4.0000	4.0000
4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	47.6100	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	2.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000	4.0000
4.0000	4.0000	4.0000	9.6100	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	29.1600	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	11.5600	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	11.5600	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	12.9600	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	13.6900	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13.6900	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	30.2500
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	31.3600
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
34.8100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	30.2500	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	32.4900	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	29.1600	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	9.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	9.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	9.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

0.0000	0.0000	0.0000	9.6100	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	9.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	34.8100	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	30.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	34.8100	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0 0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	37.2100	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
36.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	33.6400
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	33.6400

Appendix D

Sample Program Output

THIS IS FOR11.DAT. THESE DATA WERE OUTPUT FROM INPUT.F4 DURING THE FIRST RUN.
EVAL. DATA OF # 1 SET

I	E(I)	X(E(I))	STD(X(E(I)))
1	2.9655E+06	0.0000E-01	0.0000E-01
2	3.0000E+06	7.0000E-09	1.0000E+01
3	4.0000E+06	6.0000E-06	1.0000E+01
4	4.2500E+06	3.2000E-05	1.0000E+01
5	4.5000E+06	1.4000E-04	9.0000E+00
6	4.6000E+06	2.4000E-04	9.0000E+00
7	4.8000E+06	5.6500E-04	9.0000E+00
8	5.0000E+06	1.1000E-03	8.0000E+00
9	5.2500E+06	2.5000E-03	8.0000E+00
10	5.5000E+06	4.7000E-03	7.0000E+00
11	5.7500E+06	8.0000E-03	7.0000E+00
12	6.0000E+06	1.2500E-02	6.0000E+00
13	6.2000E+06	1.5900E-02	6.0000E+00
14	6.4000E+06	1.9200E-02	6.0000E+00
15	6.6000E+06	2.2400E-02	6.0000E+00
16	6.8000E+06	2.5600E-02	6.0000E+00
17	7.0000E+06	2.8700E-02	5.0000E+00
18	7.5000E+06	3.6100E-02	5.0000E+00
19	8.0000E+06	4.2700E-02	4.0000E+00
20	8.5000E+06	4.8900E-02	4.0000E+00
21	9.0000E+06	5.5000E-02	4.0000E+00
22	9.5000E+06	6.1400E-02	4.0000E+00
23	1.0000E+07	6.9300E-02	4.0000E+00
24	1.0500E+07	7.7500E-02	4.0000E+00
25	1.1000E+07	8.7000E-02	3.0000E+00
26	1.1500E+07	9.5600E-02	3.0000E+00
27	1.2000E+07	1.0300E-01	3.0000E+00
28	1.2200E+07	1.0600E-01	3.0000E+00
29	1.2400E+07	1.0800E-01	3.0000E+00
30	1.2600E+07	1.1000E-01	3.0000E+00
31	1.2800E+07	1.1100E-01	2.5000E+00
32	1.3000E+07	1.1200E-01	2.5000E+00
33	1.3400E+07	1.1250E-01	2.0000E+00
34	1.3800E+07	1.1140E-01	2.0000E+00
35	1.4200E+07	1.0880E-01	2.0000E+00
36	1.4600E+07	1.0500E-01	2.0000E+00
37	1.5000E+07	9.9400E-02	2.0000E+00
38	1.5500E+07	9.0700E-02	2.0000E+00
39	1.6000E+07	8.1800E-02	2.0000E+00
40	1.6500E+07	7.6200E-02	2.0000E+00
41	1.7000E+07	6.9200E-02	2.0000E+00
42	1.7500E+07	6.3500E-02	2.0000E+00
43	1.8000E+07	5.8900E-02	2.5000E+00
44	1.8500E+07	5.4700E-02	2.5000E+00
45	1.9000E+07	5.1300E-02	3.0000E+00
46	1.9500E+07	4.9200E-02	3.0000E+00
47	2.0000E+07	4.8800E-02	3.0000E+00

CORRELATION MATRIX

1	0
2	0 100
3	0 100 100

4	0	100	100	100																									
5	0	25	25	25	100																								
6	0	25	25	25	100	100																							
7	0	25	25	25	100	100	100																						
8	0	15	15	15	30	30	30	100																					
9	0	15	15	15	30	30	30	100	100																				
10	0	10	10	11	21	21	21	42	42	100																			
11	0	11	10	10	21	21	21	42	42	100	100																		
12	0	8	8	8	17	17	17	34	34	56	56	100																	
13	0	8	8	8	17	17	17	34	34	56	56	100	100																
14	0	7	7	7	13	13	13	27	27	45	45	64	64	100															
15	0	7	7	7	13	13	13	27	27	45	45	64	64	100	100														
16	0	7	7	7	13	13	13	27	27	45	45	64	64	100	100	100													
17	0	5	5	5	11	11	11	22	22	36	36	51	51	64	64	64	100												
18	0	5	5	5	11	11	11	22	22	36	36	51	51	64	64	64	100	100											
19	0	4	4	4	9	9	9	17	17	29	29	41	41	51	51	51	64	64	100										
20	0	4	4	4	9	9	9	17	17	29	29	41	41	51	51	51	64	64	100	100									
21	0	3	3	3	7	7	7	14	14	23	23	33	33	41	41	41	51	51	64	64	100								
22	0	3	3	3	7	7	7	14	14	23	23	33	33	41	41	41	51	51	64	64	100	100							
23	0	3	3	3	6	6	6	11	11	18	18	26	26	33	33	33	41	41	51	51	64	64	100						
24	0	3	3	3	6	6	6	11	11	18	18	26	26	33	33	33	41	41	51	51	64	64	100	100					
25	0	2	2	2	4	4	4	9	9	15	15	21	21	26	26	26	33	33	41	41	51	51	64	64	100				
26	0	2	2	2	4	4	4	9	9	15	15	21	21	26	26	26	33	33	41	41	51	51	64	64	100	100			
27	0	2	2	2	4	4	4	7	7	12	12	17	17	21	21	21	26	26	33	33	41	41	51	51	64	64	100		
28	0	2	2	2	4	4	4	7	7	12	12	17	17	21	21	21	26	26	33	33	41	41	51	51	64	64	100	100	
29	0	1	1	1	3	3	3	6	6	9	9	13	13	17	17	17	21	21	26	26	33	33	41	41	51	51	64	64	100
30	0	1	1	1	3	3	3	6	6	9	9	13	13	17	17	17	21	21	26	26	33	33	41	41	51	51	64	64	100 100
31	0	1	1	1	2	2	2	5	5	8	8	11	11	13	13	13	17	17	21	21	26	26	33	33	41	41	51	51	64 64 100
32	0	1	1	1	2	2	2	5	5	8	8	11	11	13	13	13	17	17	21	21	26	26	33	33	41	41	51	51	64 64 100 100
33	0	1	1	1	2	2	2	4	4	6	6	9	9	11	11	11	13	13	17	17	21	21	26	26	33	33	41	41	51 51 64 64 100
34	0	1	1	1	2	2	2	4	4	6	6	9	9	11	11	11	13	13	17	17	21	21	26	26	33	33	41	41	51 51 64 64 100 100
35	0	1	1	1	2	2	2	3	3	5	5	8	8	10	10	10	12	12	15	15	19	19	24	24	29	29	37	37	46 46 58 58 72 72 100
36	0	1	1	1	2	2	2	3	3	5	5	8	8	10	10	10	12	12	15	15	19	19	24	24	29	29	37	37	46 46 58 58 72 72 100 100
37	0	1	1	1	1	1	1	3	3	5	5	7	7	9	9	9	11	11	14	14	17	17	21	21	27	27	33	33	41 41 52 52 65 65 81 81 100
38	0	1	1	1	1	1	1	3	3	5	5	7	7	9	9	9	11	11	14	14	17	17	21	21	27	27	33	33	41 41 52 52 65 65 81 81 100 100
39	0	1	1	1	1	1	1	3	3	4	4	6	6	8	8	8	10	10	12	12	15	15	19	19	24	24	30	30	37 37 47 47 58 58 73 73 81 81 100
40	0	1	1	1	1	1	1	3	3	4	4	6	6	8	8	8	10	10	12	12	15	15	19	19	24	24	30	30	37 37 47 47 58 58 73 73 81 81 100 100
41	0	1	1	1	1	1	1	2	2	4	4	6	6	7	7	7	9	9	11	11	14	14	17	17	21	21	27	27	34 34 42 42 52 52 66 66 73 73 81 81 100
42	0	1	1	1	1	1	1	2	2	4	4	6	6	7	7	7	9	9	11	11	14	14	17	17	21	21	27	27	34 34 42 42 52 52 66 66 73 73 81 81 100 100

```
43   0   1   1   1   1   1   1   2   2   4   4   5   5   6   6   6   8   8  10  10  12  12  15  15  19  19  24  24
                    30  30  38  38  47  47  59  59  66  66  73  73  81  81 100
44   0   1   1   1   1   1   1   2   2   4   4   5   5   6   6   6   8   8  10  10  12  12  15  15  19  19  24  24
                    30  30  38  38  47  47  59  59  66  66  73  73  81  81 100 100
45   0   0   0   0   1   1   1   2   2   3   3   5   5   6   6   6   7   7   9   9  11  11  14  14  17  17  22  22
                    27  27  34  34  43  43  53  53  59  59  66  66  73  73  81  81 100
46   0   0   0   0   1   1   1   2   2   3   3   5   5   6   6   6   7   7   9   9  11  11  14  14  17  17  22  22
                    27  27  34  34  43  43  53  53  59  59  66  66  73  73  81  81 100 100
47   0   0   0   0   1   1   1   2   2   3   3   5   5   6   6   6   7   7   9   9  11  11  14  14  17  17  22  22
                    27  27  34  34  43  43  53  53  59  59  66  66  73  73  81  81 100 100 100
```

THINNED DATA

I	E(I)	X(E(I))	STD(X(E(I)))
1	2.9655E+06	0.0000E-01	0.0000E-01
2	3.0000E+06	7.0000E-09	1.0000E+01
3	4.0000E+06	6.0000E-06	1.0000E+01
4	4.2500E+06	3.2000E-05	1.0000E+01
5	4.5000E+06	1.4000E-04	9.0000E+00
6	5.0000E+06	1.1000E-03	8.0000E+00
7	5.5000E+06	4.7000E-03	7.0000E+00
8	6.0000E+06	1.2500E-02	6.0000E+00
9	6.4000E+06	1.9200E-02	6.0000E+00
10	7.0000E+06	2.8700E-02	5.0000E+00
11	8.0000E+06	4.2700E-02	4.0000E+00
12	9.0000E+06	5.5000E-02	4.0000E+00
13	1.0000E+07	6.9300E-02	4.0000E+00
14	1.1000E+07	8.7000E-02	3.0000E+00
15	1.2000E+07	1.0300E-01	3.0000E+00
16	1.2400E+07	1.0800E-01	3.0000E+00
17	1.2800E+07	1.1100E-01	2.5000E+00
18	1.3400E+07	1.1250E-01	2.0000E+00
19	1.4200E+07	1.0880E-01	2.0000E+00
20	1.5000E+07	9.9400E-02	2.0000E+00
21	1.6000E+07	8.1800E-02	2.0000E+00
22	1.7000E+07	6.9200E-02	2.0000E+00
23	1.8000E+07	5.8900E-02	2.5000E+00
24	1.9000E+07	5.1300E-02	3.0000E+00
25	2.0000E+07	4.8800E-02	3.0000E+00

CORRELATION MATRIX

```
 1   0
 2   0 100
 3   0 100 100
 4   0 100 100 100
 5   0  25  25  25 100
 6   0  15  15  15  30 100
 7   0  10  10  11  21  42 100
 8   0   8   8   8  17  34  56 100
 9   0   7   7   7  13  27  45  64 100
10   0   5   5   5  11  22  36  51  64 100
11   0   4   4   4   9  17  29  41  51  64 100
12   0   3   3   3   7  14  23  33  41  51  64 100
13   0   3   3   3   6  11  18  26  33  41  51  64 100
14   0   2   2   2   4   9  15  21  26  33  41  51  64 100
15   0   2   2   2   4   7  12  17  21  26  33  41  51  64 100
16   0   1   1   1   3   6   9  13  17  21  26  33  41  51  64 100
```

```
 17    0   1   1   1   2   5   8  11  13  17  21  26  33  41  51  64 100
 18    0   1   1   1   2   4   6   9  11  13  17  21  26  33  41  51  64 100
 19    0   1   1   1   2   3   5   8  10  12  15  19  24  29  37  46  58  72 100
 20    0   1   1   1   1   3   5   7   9  11  14  17  21  27  33  41  52  65  81 100
 21    0   1   1   1   1   3   4   6   8  10  12  15  19  24  30  37  47  58  73  81 100
 22    0   1   1   1   1   2   4   6   7   9  11  14  17  21  27  34  42  52  66  73  81 100
 23    0   1   1   1   1   2   4   5   6   8  10  12  15  19  24  30  38  47  59  66  73  81 100
 24    0   0   0   0   1   2   3   5   6   7   9  11  14  17  22  27  34  43  53  59  66  73  81 100
 25    0   0   0   0   1   2   3   5   6   7   9  11  14  17  22  27  34  43  53  59  66  73  81 100 100
EVAL. DATA OF  #  2 SET
 I       E(I)          X(E(I))      STD(X(E(I)))
  1   1.0063E+07   0.0000E-01   0.0000E-01
  2   1.0500E+07   3.4580E-02   4.4000E+01
  3   1.1000E+07   1.5440E-01   1.1000E+01
  4   1.1500E+07   3.1340E-01   7.0000E+00
  5   1.2000E+07   4.7540E-01   6.0000E+00
  6   1.2500E+07   6.0000E-01   5.5000E+00
  7   1.3000E+07   6.9800E-01   5.4000E+00
  8   1.3500E+07   7.8300E-01   5.3000E+00
  9   1.4000E+07   8.5300E-01   5.3000E+00
 10   1.4500E+07   9.0800E-01   5.3000E+00
 11   1.5000E+07   9.5300E-01   5.3000E+00
 12   1.5500E+07   9.9100E-01   5.2000E+00
 13   1.6000E+07   1.0230E+00   5.2000E+00
 14   1.6500E+07   1.0470E+00   5.2000E+00
 15   1.7000E+07   1.0640E+00   5.2000E+00
 16   1.7500E+07   1.0810E+00   5.2000E+00
 17   1.8000E+07   1.0950E+00   5.2000E+00
 18   1.8500E+07   1.1050E+00   5.2000E+00
 19   1.9000E+07   1.1090E+00   5.2000E+00
 20   1.9200E+07   1.1030E+00   5.2000E+00
 21   1.9400E+07   1.0850E+00   5.2000E+00
 22   1.9600E+07   1.0560E+00   5.2000E+00
 23   2.0000E+07   9.7960E-01   5.2000E+00
    CORRELATION MATRIX
  1    0
  2    0 100
  3    0  56 100
  4    0  50  72 100
  5    0  45  65  81 100
  6    0  41  58  73  81 100
  7    0  37  52  66  73  81 100
  8    0  35  50  62  69  77  85 100
  9    0  35  50  62  69  77  85 100 100
 10    0  33  47  59  66  73  81  90  90 100
 11    0  33  47  59  66  73  81  90  90 100 100
 12    0  32  45  56  63  69  77  86  86  90  90 100
 13    0  32  45  56  63  69  77  86  86  90  90 100 100
 14    0  30  43  53  59  66  73  81  81  86  86  90  90 100
 15    0  30  43  53  59  66  73  81  81  86  86  90  90 100 100
 16    0  28  41  51  56  63  70  77  77  81  81  86  86  90  90 100
 17    0  28  41  51  56  63  70  77  77  81  81  86  86  90  90 100 100
 18    0  26  37  46  51  56  63  70  70  73  73  77  77  81  81  86  85 100
 19    0  26  37  46  51  56  63  70  70  73  73  77  77  81  81  86  85 100 100
```

```
20    0  23  33  41  46  51  56  63  63  66  66  69  69  73  73  77  77  81  81 100
21    0  23  33  41  46  51  56  63  63  66  66  69  69  73  73  77  77  81  81 100 100
22    0  18  26  33  37  41  45  50  50  53  53  56  56  58  58  62  62  65  65  72  72 100
23    0  18  26  33  37  41  45  50  50  53  53  56  56  58  58  62  62  65  65  72  72 100 100
  THINNED DATA
I      E(I)        X(E(I))     STD(X(E(I)))
 1  1.0063E+07  0.0000E-01  0.0000E-01
 2  1.0500E+07  3.4580E-02  4.4000E+01
 3  1.1000E+07  1.5440E-01  1.1000E+01
 4  1.1500E+07  3.1340E-01  7.0000E+00
 5  1.2000E+07  4.7540E-01  6.0000E+00
 6  1.2500E+07  6.0000E-01  5.5000E+00
 7  1.3000E+07  6.9800E-01  5.4000E+00
 8  1.3500E+07  7.8300E-01  5.3000E+00
 9  1.4500E+07  9.0800E-01  5.3000E+00
10  1.5500E+07  9.9100E-01  5.2000E+00
11  1.6500E+07  1.0470E+00  5.2000E+00
12  1.7500E+07  1.0810E+00  5.2000E+00
13  1.8500E+07  1.1050E+00  5.2000E+00
14  1.9200E+07  1.1030E+00  5.2000E+00
15  1.9600E+07  1.0560E+00  5.2000E+00
16  2.0000E+07  9.7960E-01  5.2000E+00
  CORRELATION MATRIX
 1    0
 2    0 100
 3    0  56 100
 4    0  50  72 100
 5    0  45  65  81 100
 6    0  41  58  73  81 100
 7    0  37  52  66  73  81 100
 8    0  35  50  62  69  77  85 100
 9    0  33  47  59  66  73  81  90 100
10    0  32  45  56  63  69  77  86  90 100
11    0  30  43  53  59  66  73  81  86  90 100
12    0  28  41  51  56  63  70  77  81  86  90 100
13    0  26  37  46  51  56  63  70  73  77  81  86 100
14    0  23  33  41  46  51  56  63  66  69  73  77  81 100
15    0  18  26  33  37  41  45  50  53  56  58  62  65  72 100
16    0  18  26  33  37  41  45  50  53  56  58  62  65  72 100 100
```

THIS IS FOR14.DAT. THESE DATA WERE OUTPUT FROM GLUCS.F4 DURING THE FIRST RUN.

EXPERIMENTAL DATA

I	E(I)	X(E(I))	STD(X(E(I)))
1	1.4670E+07	1.0810E-01	1.0607E+00
2	1.5300E+07	1.0000E-01	1.8426E+00
3	1.6550E+07	8.0100E-02	1.4349E+00
4	1.7000E+07	7.2100E-02	1.3903E+00
5	1.7860E+07	6.3100E-02	1.3042E+00
6	1.8950E+07	5.3200E-02	1.5369E+00
7	1.4670E+07	8.8300E+00	1.0803E+00
8	1.5300E+07	9.9300E+00	2.4825E+00
9	1.5520E+07	1.0520E+01	1.5166E+00
10	1.5900E+07	1.1460E+01	1.6435E+00
11	1.6350E+07	1.2480E+01	2.0130E+00
12	1.6460E+07	1.2970E+01	1.6453E+00
13	1.6550E+07	1.3350E+01	1.3813E+00
14	1.7000E+07	1.4730E+01	1.2849E+00
15	1.7860E+07	1.7160E+01	1.1996E+00
16	1.8950E+07	1.9240E+01	9.7620E-01

I CORRELATION MATRIX

1	100													
2	42	100												
3	54	61	100											
4	56	59	84	100										
5	59	58	82	80	100									
6	50	53	76	73	72	100								
7	-19	2	2	2	3	2	100							
8	1	-80	-32	-28	-25	-25	14	100						
9	2	-30	-49	-44	-48	-39	23	44	100					
10	2	-32	-51	-46	-40	-41	22	45	71	100				
11	2	-24	-38	-35	-30	-31	18	35	54	55	100			
12	2	-28	-45	-40	-36	-36	21	41	64	66	50	100		
13	2	-32	-52	-47	-41	-42	26	48	75	77	59	70	100	
14	3	-30	-49	-44	-39	-39	28	47	73	75	57	68	79	100
15	3	-28	-45	-40	-35	-36	29	45	70	71	55	65	76	75
16	3	-18	-29	-26	-22	-23	36	37	59	58	46	55	64	64

100
64
100

EVALUATED DATA

I	E(I)	X(E(I))
1	2.9655E+06	0.0000E-01
2	3.0000E+06	7.0000E-09
3	4.0000E+06	6.0000E-06
4	4.2500E+06	3.2000E-05
5	4.5000E+06	1.4000E-04
6	5.0000E+06	1.1000E-03
7	5.5000E+06	4.7000E-03
8	6.0000E+06	1.2500E-02
9	6.4000E+06	1.9200E-02
10	7.0000E+06	2.8700E-02
11	8.0000E+06	4.2700E-02
12	9.0000E+06	5.5000E-02
13	1.0000E+07	6.9300E-02
14	1.1000E+07	8.7000E-02
15	1.2000E+07	1.0300E-01
16	1.2400E+07	1.0800E-01

17	1.2800E+07	1.1100E-01
18	1.3400E+07	1.1250E-01
19	1.4200E+07	1.0880E-01
20	1.5000E+07	9.9400E-02
21	1.6000E+07	8.1800E-02
22	1.7000E+07	6.9200E-02
23	1.8000E+07	5.8900E-02
24	1.9000E+07	5.1300E-02
25	2.0000E+07	4.8800E-02
26	1.0063E+07	0.0000E-01
27	1.0500E+07	3.4580E-02
28	1.1000E+07	1.5440E-01
29	1.1500E+07	3.1340E-01
30	1.2000E+07	4.7540E-01
31	1.2500E+07	6.0000E-01
32	1.3000E+07	6.9800E-01
33	1.3500E+07	7.8300E-01
34	1.4500E+07	9.0800E-01
35	1.5500E+07	9.9100E-01
36	1.6500E+07	1.0470E+00
37	1.7500E+07	1.0810E+00
38	1.8500E+07	1.1050E+00
39	1.9200E+07	1.1030E+00
40	1.9600E+07	1.0560E+00
41	2.0000E+07	9.7960E-01

THEORETICAL DATA

I	E(I)	REXP(E(I))	RTHEORY(E(I))
1	1.4670E+07	1.0810E-01	1.0328E-01
2	1.5300E+07	1.0000E-01	9.4120E-02
3	1.6550E+07	8.0100E-02	7.4870E-02
4	1.7000E+07	7.2100E-02	6.9200E-02
5	1.7860E+07	6.3100E-02	6.0342E-02
6	1.8950E+07	5.3200E-02	5.1680E-02
7	1.4670E+07	8.8300E+00	8.9285E+00
8	1.5300E+07	9.9300E+00	1.0353E+01
9	1.5520E+07	1.0520E+01	1.0993E+01
10	1.5900E+07	1.1460E+01	1.2128E+01
11	1.6350E+07	1.2480E+01	1.3420E+01
12	1.6460E+07	1.2970E+01	1.3746E+01
13	1.6550E+07	1.3350E+01	1.4007E+01
14	1.7000E+07	1.4730E+01	1.5376E+01
15	1.7860E+07	1.7160E+01	1.8058E+01
16	1.8950E+07	1.9240E+01	2.1357E+01

SENSITIVITY MATRIX G-ONLY NONZERO VALUES GIVEN:

ROW I	COLUMN J	G(I,J)
1	19	4.1250E-01
7	19	-3.5661E+01
1	20	5.8750E-01
2	20	7.0000E-01
7	20	-5.0790E+01
8	20	-7.6997E+01
9	20	-5.8470E+01
10	20	-1.4514E+01
2	21	3.0000E-01

3	21	4.5000E-01
8	21	-3.2999E+01
9	21	-6.3342E+01
10	21	-1.3063E+02
11	21	-1.1272E+02
12	21	-9.7665E+01
13	21	-8.4188E+01
3	22	5.5000E-01
4	22	1.0000E+00
5	22	1.4000E-01
11	22	-6.0694E+01
12	22	-8.3196E+01
13	22	-1.0290E+02
14	22	-2.2219E+02
15	22	-4.1896E+01
5	23	8.6000E-01
6	23	5.0000E-02
15	23	-2.5736E+02
16	23	-2.0662E+01
6	24	9.5000E-01
16	24	-3.9259E+02
7	34	8.0366E+00
8	34	2.1249E+00
7	35	1.6461E+00
8	35	8.4998E+00
9	35	1.0859E+01
10	35	7.1805E+00
11	35	1.9382E+00
12	35	5.2629E-01
9	36	2.2161E-01
10	36	4.7870E+00
11	36	1.0983E+01
12	36	1.2631E+01
13	36	1.2689E+01
14	36	7.2254E+00
13	37	6.6782E-01
14	37	7.2254E+00
15	37	1.0606E+01
15	38	5.9660E+00
16	38	6.9107E+00
16	39	1.2439E+01

THIS IS FOR17.DAT. THESE DATA WERE OUTPUT FROM OUTPUT.F4 DURING THE FIRST RUN.
FINAL THINNED DATA OF # 1 SET

I	E(I)	X(E(I))	STD(X(E(I)))
1	2.9655E+06	0.0000E-01	0.0000E-01
2	3.0000E+06	7.0107E-09	9.9845E+00
3	4.5000E+06	1.4039E-04	8.9742E+00
4	5.0000E+06	1.1054E-03	7.9578E+00
5	5.5000E+06	4.7336E-03	6.9426E+00
6	6.0000E+06	1.2609E-02	5.9350E+00
7	6.4000E+06	1.9410E-02	5.9145E+00
8	7.0000E+06	2.9027E-02	4.9168E+00
9	8.0000E+06	4.3107E-02	3.9214E+00
10	9.0000E+06	5.5784E-02	3.8913E+00
11	1.0000E+07	7.0535E-02	3.8480E+00
12	1.1000E+07	8.8454E-02	2.8539E+00
13	1.2000E+07	1.0515E-01	2.7866E+00
14	1.2400E+07	1.1082E-01	2.6836E+00
15	1.2800E+07	1.1402E-01	2.1132E+00
16	1.3400E+07	1.1556E-01	1.5275E+00
17	1.4200E+07	1.1250E-01	1.1222E+00
18	1.5000E+07	1.0319E-01	9.9722E-01
19	1.6000E+07	8.5812E-02	1.1271E+00
20	1.7000E+07	7.1227E-02	1.0361E+00
21	1.8000E+07	6.0431E-02	1.0767E+00
22	1.9000E+07	5.2429E-02	1.2350E+00
23	2.0000E+07	4.9874E-02	1.2349E+00

CORRELATION MATRIX

1	0																				
2	0	100																			
3	0	25	100																		
4	0	15	30	100																	
5	0	10	21	42	100																
6	0	8	17	34	56	100															
7	0	7	13	27	45	64	100														
8	0	5	11	21	36	51	64	100													
9	0	4	8	17	28	41	51	64	100												
10	0	3	7	13	23	32	40	50	63	100											
11	0	3	5	11	18	25	32	40	50	63	100										
12	0	2	4	8	14	20	25	31	39	49	62	100									
13	0	2	3	6	11	15	19	24	30	38	48	61	100								
14	0	1	2	5	8	12	15	18	23	29	37	46	59	100							
15	0	1	2	4	6	9	11	13	17	21	27	34	43	56	100						
16	0	1	1	2	4	6	7	9	11	14	18	22	29	37	49	100					
17	0	0	1	2	3	4	5	6	8	10	12	16	20	26	34	48	100				
18	0	0	1	1	2	3	3	4	5	6	8	10	13	17	23	31	45	100			
19	0	0	1	1	2	2	3	4	5	6	8	10	12	16	21	29	46	63	100		
20	0	0	1	1	2	3	3	4	5	6	8	10	13	16	22	30	47	70	78	100	
21	0	0	1	1	2	2	3	4	5	6	8	10	12	16	21	29	46	69	73	80	100
22	0	0	0	1	2	2	3	3	4	5	7	8	11	14	18	25	41	59	67	71	71	100
23	0	0	0	1	2	2	3	3	4	5	7	8	11	14	18	25	41	59	67	71	71	100	100

FINAL THINNED DATA OF # 2 SET

I	E(I)	X(E(I))	STD(X(E(I)))
1	1.0063E+07	0.0000E-01	0.0000E-01
2	1.0500E+07	3.6601E-02	3.9319E+01

3	1.1000E+07	1.5762E-01	9.5470E+00
4	1.1500E+07	3.1860E-01	5.6111E+00
5	1.2000E+07	4.8292E-01	4.5185E+00
6	1.2500E+07	6.0966E-01	3.7811E+00
7	1.3000E+07	7.1026E-01	3.2186E+00
8	1.3500E+07	7.9800E-01	2.4367E+00
9	1.4500E+07	9.3112E-01	1.3015E+00
10	1.5500E+07	1.0026E+00	1.2043E+00
11	1.6500E+07	1.0456E+00	1.1979E+00
12	1.7500E+07	1.0769E+00	1.3880E+00
13	1.8500E+07	1.0599E+00	1.9729E+00
14	1.9200E+07	1.0088E+00	1.9337E+00
15	1.9600E+07	1.0003E+00	3.9258E+00
16	2.0000E+07	9.2790E-01	3.9260E+00

CORRELATION MATRIX

1	0														
2	0	100													
3	0	49	100												
4	0	41	62	100											
5	0	34	52	70	100										
6	0	27	41	55	65	100									
7	0	19	29	40	47	57	100								
8	0	14	21	29	34	42	54	100							
9	0	8	12	17	20	24	31	44	100						
10	0	6	9	13	15	18	23	34	60	100					
11	0	6	10	13	15	19	24	34	66	79	100				
12	0	5	8	11	13	16	21	30	59	62	60	100			
13	0	4	6	8	9	11	14	21	41	44	50	20	100		
14	0	4	6	8	9	11	15	21	42	43	44	50	7	100	
15	0	1	2	3	3	4	5	8	16	16	17	17	10	28	100
16	0	1	2	3	3	4	5	8	16	16	17	17	10	28	100

FINAL THINNED CORRELATION MATRIX OF RECTANGULAR BLOCK 2 1

SET 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
SET 2																							
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	4	5	4	4	4	3	3
3	0	0	0	0	0	0	0	0	1	1	1	1	1	2	3	3	6	7	6	6	6	5	5
4	0	0	0	0	0	0	1	1	1	1	1	2	2	3	3	5	8	9	8	8	8	7	7
5	0	0	0	0	0	0	1	1	1	1	1	2	2	3	4	6	10	11	9	10	10	9	9
6	0	0	0	0	0	0	1	1	1	1	1	2	2	3	4	5	7	12	14	11	12	12	10	10
7	0	0	0	0	1	1	1	1	1	2	2	3	4	5	6	9	16	17	14	16	16	13	13
8	0	0	0	0	1	1	1	2	2	3	3	4	5	7	9	13	22	25	20	22	22	19	19
9	0	0	0	1	1	2	3	3	4	5	7	8	11	14	18	25	45	49	40	47	47	40	40
10	0	0	0	1	2	2	3	3	4	5	7	9	11	14	19	26	46	55	43	37	37	34	34
11	0	0	0	1	2	2	3	4	4	6	7	9	11	15	20	27	47	53	54	53	45	40	40
12	0	0	0	1	1	2	2	3	4	5	6	8	10	12	16	23	39	49	33	52	52	35	35
13	0	0	0	1	1	1	2	2	3	4	4	6	7	9	12	17	29	37	35	31	50	42	42
14	0	0	0	1	1	2	2	2	3	4	5	6	8	10	13	18	30	40	40	46	41	71	71
15	0	0	0	0	0	1	1	1	1	1	2	2	3	4	5	7	11	14	14	16	16	23	23
16	0	0	0	0	0	1	1	1	1	1	2	2	3	4	5	7	11	14	14	16	16	23	23

THIS IS FOR18.DAT. THESE DATA WERE OUTPUT FROM OUTPUT.F4 (DURING THE FIRST RUN). THIS FILE WAS RENAMED TO FOR18.ONE AND USED AS INPUT TO INPUT.F4 DURING THE SECOND RUN.

```
 2.6056E+04 5.5454E+01          0          0          0          0 561 3103
 0.0000E-01-2.9130E+06          0          0          3         25 561 3103
          2          2          8          4         25          2 561 3103
2.96550E+06 0.0000E-013.00000E+06 7.0107E-094.00000E+06 6.0092E-06 561 3103
4.25000E+06 3.2049E-054.50000E+06 1.4039E-045.00000E+06 1.1054E-03 561 3103
5.50000E+06 4.7336E-036.00000E+06 1.2609E-026.40000E+06 1.9410E-02 561 3103
7.00000E+06 2.9027E-028.00000E+06 4.3187E-029.00000E+06 5.5784E-02 561 3103
1.00000E+07 7.0535E-021.10000E+07 8.8454E-021.20000E+07 1.0515E-01 561 3103
1.24000E+07 1.1082E-011.28000E+07 1.1402E-011.34000E+07 1.1556E-01 561 3103
1.42000E+07 1.1250E-011.50000E+07 1.0319E-011.60000E+07 8.5812E-02 561 3103
1.70000E+07 7.1227E-021.80000E+07 6.0431E-021.90000E+07 5.2429E-02 561 3103
2.00000E+07 4.9874E-020.00000E-01 0.0000E-010.00000E-01 0.0000E-01 561 3103
 0.0000E-01 0.0000E-01          0          0          0          0 561 3  0
 0.0000E-01 0.0000E-01          0          0          0          0 561 0  0
 2.6056E+04 5.5454E+01          0          0          0          2 56133103
 0.0000E-01 0.0000E-01          0        103          0          1 56133103
 0.0000E-01 0.0000E-01          1          5        300         24 56133103
 1.0000E-05 2.9655E+06 3.0000E+06 4.5000E+06 5.0000E+06 5.5000E+06 56133103
 6.0000E+06 6.4000E+06 7.0000E+06 8.0000E+06 9.0000E+06 1.0000E+07 56133103
 1.1000E+07 1.2000E+07 1.2400E+07 1.2800E+07 1.3400E+07 1.4200E+07 56133103
 1.5000E+07 1.6000E+07 1.7000E+07 1.8000E+07 1.9000E+07 2.0000E+07 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 9.9691E-03 2.2394E-03 1.1908E-03 56133103
 7.2639E-04 4.9612E-04 3.9478E-04 2.6177E-04 1.6624E-04 1.3102E-04 56133103
 1.0243E-04 5.9644E-05 4.5177E-05 3.3042E-05 1.8969E-05 9.1075E-06 56133103
 4.7109E-06 2.7447E-06 2.8929E-06 2.7432E-06 2.7711E-06 2.7791E-06 56133103
 8.0535E-03 2.1407E-03 1.3059E-03 8.9189E-04 7.0970E-04 4.7061E-04 56133103
 2.9886E-04 2.3552E-04 1.8416E-04 1.0722E-04 8.1215E-05 5.9396E-05 56133103
 3.4103E-05 1.6372E-05 8.4678E-06 4.9354E-06 5.2008E-06 4.9319E-06 56133103
 4.9820E-06 4.9964E-06 6.3326E-03 2.3167E-03 1.5822E-03 1.2590E-03 56133103
 8.3487E-04 5.3018E-04 4.1782E-04 3.2668E-04 1.9022E-04 1.4407E-04 56133103
 1.0538E-04 6.0501E-05 2.9044E-05 1.5022E-05 8.7554E-06 9.2255E-06 56133103
 8.7490E-06 8.8380E-06 8.8636E-06 4.8199E-03 2.3022E-03 1.8321E-03 56133103
 1.2148E-03 7.7146E-04 6.0797E-04 4.7534E-04 2.7681E-04 2.0965E-04 56133103
 1.5333E-04 8.8028E-05 4.2262E-05 2.1858E-05 1.2740E-05 1.3425E-05 56133103
 1.2730E-05 1.2860E-05 1.2897E-05 3.5224E-03 2.2398E-03 1.4852E-03 56133103
 9.4319E-04 7.4328E-04 5.8116E-04 3.3840E-04 2.5631E-04 1.8746E-04 56133103
 1.0762E-04 5.1671E-05 2.6724E-05 1.5575E-05 1.6412E-05 1.5564E-05 56133103
 1.5722E-05 1.5768E-05 3.4981E-03 1.8525E-03 1.1764E-03 9.2707E-04 56133103
 7.2486E-04 4.2206E-04 3.1968E-04 2.3381E-04 1.3424E-04 6.4444E-05 56133103
 3.3329E-05 1.9429E-05 2.0469E-05 1.9412E-05 1.9610E-05 1.9667E-05 56133103
 2.4175E-03 1.2248E-03 9.6527E-04 7.5471E-04 4.3945E-04 3.3284E-04 56133103
 2.4344E-04 1.3977E-04 6.7098E-05 3.4708E-05 2.0225E-05 2.1315E-05 56133103
 2.0212E-05 2.0418E-05 2.0477E-05 1.5377E-03 9.6528E-04 7.5471E-04 56133103
 4.3944E-04 3.3285E-04 2.4344E-04 1.3977E-04 6.7097E-05 3.4706E-05 56133103
```

2.0225E-05	2.1314E-05	2.0211E-05	2.0417E-05	2.0476E-05	1.5142E-03	56133103
9.4075E-04	5.4776E-04	4.1490E-04	3.0345E-04	1.7422E-04	8.3636E-05	56133103
4.3256E-05	2.5216E-05	2.6566E-05	2.5196E-05	2.5451E-05	2.5524E-05	56133103
1.4807E-03	6.8229E-04	5.1680E-04	3.7797E-04	2.1701E-04	1.0418E-04	56133103
5.3884E-05	3.1402E-05	3.3094E-05	3.1381E-05	3.1702E-05	3.1792E-05	56133103
8.1450E-04	4.8503E-04	3.5474E-04	2.0366E-04	9.7774E-05	5.0568E-05	56133103
2.9476E-05	3.1058E-05	2.9454E-05	2.9753E-05	2.9839E-05	7.7653E-04	56133103
4.4161E-04	2.5354E-04	1.2172E-04	6.2950E-05	3.6695E-05	3.8666E-05	56133103
3.6669E-05	3.7041E-05	3.7147E-05	7.2019E-04	3.1531E-04	1.5138E-04	56133103
7.8297E-05	4.5628E-05	4.8088E-05	4.5598E-05	4.6063E-05	4.6196E-05	56133103
4.4657E-04	1.5751E-04	8.1467E-05	4.7485E-05	5.0032E-05	4.7449E-05	56133103
4.7929E-05	4.8070E-05	2.3334E-04	8.1466E-05	4.7483E-05	5.0038E-05	56133103
4.7451E-05	4.7932E-05	4.8070E-05	1.2594E-04	5.0112E-05	5.7830E-05	56133103
5.5163E-05	5.5929E-05	5.6288E-05	9.9444E-05	7.0500E-05	7.2093E-05	56133103
7.4099E-05	7.2516E-05	1.2703E-04	9.1556E-05	8.8852E-05	9.3069E-05	56133103
1.0734E-04	8.9245E-05	9.0256E-05	1.1592E-04	9.4277E-05	1.5251E-04	56133103
0.0000E-01	0.0000E-01	651	16	0	1	56133103
0.0000E-01	0.0000E-01	0	6	24	17	56133103
1.0000E-05	2.9655E+06	3.0000E+^6	4.5000E+06	5.0000E+06	5.5000E+06	56133103
6.0000E+06	6.4000E+06	7.0000E+06	8.0000E+06	9.0000E+06	1.0000E+07	56133103
1.1000E+07	1.2000E+07	1.2400E+07	1.2800E+07	1.3400E+07	1.4200E+07	56133103
1.5000E+07	1.6000E+07	1.7000E+07	1.8000E+07	1.9000E+07	2.0000E+07	56133103
1.0000E-05	1.0063E+07	1.0500E+07	1.1000E+07	1.1500E+07	1.2000E+07	56133103
1.2500E+07	1.3000E+07	1.3500E+07	1.4500E+07	1.5500E+07	1.6500E+07	56133103
1.7500E+07	1.8500E+07	1.9200E+07	1.9600E+07	2.0000E+07	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	7.9455E-06	56133103
1.4284E-05	2.5339E-05	3.6872E-05	4.5079E-05	5.6225E-05	5.8542E-05	56133103
5.8542E-05	7.2972E-05	9.0891E-05	8.5306E-05	1.0620E-04	1.3207E-04	56133103
1.3743E-04	1.3743E-04	1.7972E-04	1.7768E-04	1.6325E-04	1.6556E-04	56133103
1.7203E-04	1.6849E-04	0.0000E-01	0.0000E-01	2.9421E-06	5.2892E-06	56133103
9.3834E-06	1.3653E-05	1.6692E-05	2.0820E-05	2.1678E-05	2.1677E-05	56133103
2.7021E-05	3.3656E-05	3.1588E-05	3.9324E-05	4.8904E-05	5.0887E-05	56133103
5.0888E-05	6.6545E-05	6.5792E-05	6.0451E-05	6.1310E-05	6.3702E-05	56133103
6.2389E-05	0.0000E-01	0.0000E-01	2.3501E-06	4.2248E-06	7.4950E-06	56133103
1.0906E-05	1.3334E-05	1.6630E-05	1.7316E-05	1.7315E-05	2.1584E-05	56133103
2.6884E-05	2.5232E-05	3.1411E-05	3.9063E-05	4.0648E-05	4.0649E-05	56133103
5.3158E-05	5.2555E-05	4.8285E-05	4.8971E-05	5.0883E-05	4.9833E-05	56133103
0.0000E-01	0.0000E-01	2.2400E-06	4.0269E-06	7.1438E-06	1.0395E-05	56133103
1.2709E-05	1.5851E-05	1.6504E-05	1.6504E-05	2.0572E-05	2.5624E-05	56133103
2.4049E-05	2.9940E-05	3.7233E-05	3.8743E-05	3.8743E-05	5.0666E-05	56133103
5.0092E-05	4.6023E-05	4.6676E-05	4.8500E-05	4.7502E-05	0.0000E-01	56133103
0.0000E-01	2.2808E-06	4.1002E-06	7.2739E-06	1.0584E-05	1.2940E-05	56133103
1.6140E-05	1.6805E-05	1.6804E-05	2.0947E-05	2.6092E-05	2.4487E-05	56133103
3.0484E-05	3.7910E-05	3.9448E-05	3.9449E-05	5.1587E-05	5.1004E-05	56133103
4.6867E-05	4.7524E-05	4.9375E-05	4.8364E-05	0.0000E-01	0.0000E-01	56133103
2.4844E-06	4.4665E-06	7.9237E-06	1.1530E-05	1.4096E-05	1.7581E-05	56133103
1.8306E-05	1.8306E-05	2.2818E-05	2.8422E-05	2.6676E-05	3.3207E-05	56133103

```
4.1297E-05 4.2973E-05 4.2973E-05 5.6196E-05 5.5563E-05 5.1046E-05 56133103
5.1773E-05 5.3789E-05 5.2685E-05 0.0000E-01 0.0000E-01 2.7053E-06 56133103
4.8635E-06 8.6280E-06 1.2554E-05 1.5349E-05 1.9144E-05 1.9933E-05 56133103
1.9933E-05 2.4845E-05 3.0947E-05 2.9045E-05 3.6158E-05 4.4968E-05 56133103
4.6792E-05 4.6792E-05 6.1191E-05 6.0499E-05 5.5587E-05 5.6377E-05 56133103
5.8573E-05 5.7376E-05 0.0000E-01 0.0000E-01 2.9122E-06 5.2355E-06 56133103
9.2878E-06 1.3515E-05 1.6523E-05 2.0608E-05 2.1457E-05 2.1457E-05 56133103
2.6745E-05 3.3315E-05 3.1267E-05 3.8924E-05 4.8406E-05 5.0370E-05 56133103
5.0371E-05 6.6155E-05 6.4147E-05 5.8756E-05 6.3167E-05 6.6270E-05 56133103
6.3917E-05 0.0000E-01 0.0000E-01 2.7994E-06 5.0329E-06 8.9284E-06 56133103
1.2992E-05 1.5883E-05 1.9811E-05 2.0627E-05 2.0627E-05 2.5711E-05 56133103
3.2026E-05 3.0057E-05 3.7418E-05 4.6534E-05 4.8422E-05 4.8422E-05 56133103
6.2750E-05 6.5785E-05 5.7694E-05 4.6338E-05 4.7864E-05 5.0739E-05 56133103
0.0000E-01 0.0000E-01 2.8677E-06 5.1554E-06 9.1460E-06 1.3308E-05 56133103
1.6270E-05 2.0292E-05 2.1129E-05 2.1129E-05 2.6337E-05 3.2805E-05 56133103
3.0788E-05 3.8328E-05 4.7667E-05 4.9599E-05 4.9600E-05 6.3814E-05 56133103
6.3591E-05 7.3294E-05 6.5164E-05 5.8460E-05 5.9655E-05 0.0000E-01 56133103
0.0000E-01 2.7843E-06 5.0056E-06 8.8802E-06 1.2921E-05 1.5798E-05 56133103
1.9704E-05 2.0515E-05 2.0515E-05 2.5574E-05 3.1851E-05 2.9895E-05 56133103
3.7216E-05 4.6280E-05 4.8160E-05 4.8160E-05 6.1412E-05 6.7897E-05 56133103
5.1821E-05 7.4781E-05 7.7407E-05 6.0164E-05 0.0000E-01 0.0000E-01 56133103
2.9843E-06 5.3651E-06 9.5176E-06 1.3849E-05 1.6932E-05 2.1117E-05 56133103
2.1988E-05 2.1989E-05 2.7408E-05 3.4140E-05 3.2041E-05 3.9887E-05 56133103
4.9606E-05 5.1615E-05 5.1617E-05 6.4450E-05 7.2249E-05 7.7619E-05 56133103
6.4255E-05 1.0524E-04 1.0144E-04 0.0000E-01 0.0000E-01 3.0572E-06 56133103
5.4964E-06 9.7505E-06 1.4188E-05 1.7346E-05 2.1635E-05 2.2526E-05 56133103
2.2526E-05 2.8078E-05 3.4975E-05 3.2824E-05 4.0865E-05 5.0820E-05 56133103
5.2882E-05 5.2881E-05 6.4140E-05 7.6625E-05 8.7711E-05 9.2485E-05 56133103
8.5324E-05 1.7028E-04 0.0000E-01 0.0000E-01 2.2676E-06 4.0765E-06 56133103
7.2318E-06 1.0523E-05 1.2865E-05 1.6046E-05 1.6707E-05 1.6706E-05 56133103
2.0826E-05 2.5940E-05 2.4344E-05 3.0309E-05 3.7691E-05 3.9220E-05 56133103
3.9220E-05 4.8086E-05 5.6202E-05 6.2011E-05 6.3851E-05 6.5824E-05 56133103
1.0948E-04 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
0.0000E-01 0.0000E-01          0          0          0          0 56133  0
0.0000E-01 0.0000E-01          0          0          0          0 561 0  0
```

THIS IS FOR11.DAT. THESE DATA WERE OUTPUT FROM INPUT.F4 DURING THE
SECOND RUN.
EVAL. DATA OF # 1 SET

I	E(I)	X(E(I))	STD(X(E(I)))
1	2.9655E+06	0.0000E-01	0.0000E-01
2	3.0000E+06	7.0107E-09	9.9845E+00
3	4.0000E+06	6.0092E-06	9.9845E+00
4	4.2500E+06	3.2049E-05	9.9845E+00
5	4.5000E+06	1.4039E-04	8.9741E+00
6	5.0000E+06	1.1054E-03	7.9578E+00
7	5.5000E+06	4.7336E-03	6.9425E+00
8	6.0000E+06	1.2609E-02	5.9350E+00
9	6.4000E+06	1.9410E-02	5.9145E+00
10	7.0000E+06	2.9027E-02	4.9168E+00
11	8.0000E+06	4.3187E-02	3.9214E+00
12	9.0000E+06	5.5784E-02	3.8913E+00
13	1.0000E+07	7.0535E-02	3.8480E+00
14	1.1000E+07	8.8454E-02	2.8539E+00
15	1.2000E+07	1.0515E-01	2.7866E+00
16	1.2400E+07	1.1082E-01	2.6836E+00
17	1.2800E+07	1.1402E-01	2.1132E+00
18	1.3400E+07	1.1556E-01	1.5275E+00
19	1.4200E+07	1.1250E-01	1.1222E+00
20	1.5000E+07	1.0319E-01	9.9722E-01
21	1.6000E+07	8.5812E-02	1.1271E+00
22	1.7000E+07	7.1227E-02	1.0361E+00
23	1.8000E+07	6.0431E-02	1.0767E+00
24	1.9000E+07	5.2429E-02	1.2349E+00
25	2.0000E+07	4.9874E-02	1.2349E+00

CORRELATION MATRIX

1	0																					
2	0	100																				
3	0	100	100																			
4	0	100	100	100																		
5	0	25	25	25	100																	
6	0	15	15	15	30	100																
7	0	10	10	10	21	42	100															
8	0	8	8	8	17	34	56	100														
9	0	7	7	7	13	27	45	64	100													
10	0	5	5	5	11	21	36	51	64	100												
11	0	4	4	4	8	17	28	41	51	64	100											
12	0	3	3	3	7	13	23	32	40	50	63	100										
13	0	3	3	3	5	11	18	25	32	40	50	63	100									
14	0	2	2	2	4	8	14	20	25	31	39	49	62	100								
15	0	2	2	2	3	6	11	15	19	24	30	38	48	61	100							
16	0	1	1	1	2	5	8	12	15	18	23	29	37	46	59	100						
17	0	1	1	1	2	4	6	9	11	13	17	21	27	34	43	56	100					
18	0	1	1	1	1	2	4	6	7	9	11	14	18	22	29	37	49	100				
19	0	0	0	0	1	2	3	4	5	6	8	10	12	16	20	26	34	48	100			
20	0	0	0	0	1	1	2	3	3	4	5	6	8	10	13	17	23	31	45	100		
21	0	0	0	0	1	1	2	2	3	4	5	6	8	10	12	16	21	29	46	63	100	
22	0	0	0	0	1	1	2	3	3	4	5	6	8	10	13	16	22	30	47	70	78	100
23	0	0	0	0	1	1	2	2	3	4	5	6	8	10	12	16	21	29	46	69	73	80	100
24	0	0	0	0	0	1	2	2	3	3	4	5	7	8	11	14	18	25	41	59	67	71	71	100

```
 25    0   0   0   0   0   1   2   2   3   3   4   5   7   8  11  14  18  25  41  59  67  71  71 100 100
EVAL. DATA OF  # 2 SET
 I       E(I)        X(E(I))    STD(X(E(I)))
  1   1.0063E+07  0.0000E-01  0.0000E-01
  2   1.0500E+07  3.6601E-02  3.9319E+01
  3   1.1000E+07  1.5762E-01  9.5470E+00
  4   1.1500E+07  3.1860E-01  5.6111E+00
  5   1.2000E+07  4.8292E-01  4.5185E+00
  6   1.2500E+07  6.0966E-01  3.7811E+00
  7   1.3000E+07  7.1026E-01  3.2187E+00
  8   1.3500E+07  7.9800E-01  2.4367E+00
  9   1.4500E+07  9.3112E-01  1.3015E+00
 10   1.5500E+07  1.0026E+00  1.2043E+00
 11   1.6500E+07  1.0456E+00  1.1979E+00
 12   1.7500E+07  1.0769E+00  1.3881E+00
 13   1.8500E+07  1.0599E+00  1.9729E+00
 14   1.9200E+07  1.0088E+00  1.9336E+00
 15   1.9600E+07  1.0003E+00  3.9258E+00
 16   2.0000E+07  9.2790E-01  3.9258E+00
    CORRELATION MATRIX
  1    0
  2    0 100
  3    0  49 100
  4    0  41  62 100
  5    0  34  52  70 100
  6    0  27  41  55  65 100
  7    0  19  29  40  47  57 100
  8    0  14  21  29  34  42  54 100
  9    0   8  12  17  20  24  31  44 100
 10    0   6   9  13  15  18  23  34  60 100
 11    0   6  10  13  15  19  24  34  66  79 100
 12    0   5   8  11  13  16  21  30  59  62  60 100
 13    0   4   6   8   9  11  14  21  41  44  50  20 100
 14    0   4   6   8   9  11  15  21  42  43  44  50   7 100
 15    0   1   2   3   3   4   5   8  16  16  17  17  10  28 100
 16    0   1   2   3   3   4   5   8  16  16  17  17  10  28 100 100
EVAL. DATA OF  # 3 SET
 I       E(I)        X(E(I))    STD(X(E(I)))
  1   9.5700E+05  0.0000E-01  0.0000E-01
  2   1.0000E+06  1.2800E-05  3.0000E+01
  3   1.1000E+06  2.5000E-05  3.0000E+01
  4   1.2000E+06  5.0000E-05  3.0000E+01
  5   1.3000E+06  1.1200E-04  3.0000E+01
  6   1.4000E+06  2.3000E-04  3.0000E+01
  7   1.5000E+06  4.7000E-04  3.0000E+01
  8   1.6000E+06  9.5000E-04  3.0000E+01
  9   1.7000E+06  1.8000E-03  3.0000E+01
 10   1.8000E+06  4.0000E-03  3.0000E+01
 11   1.9000E+06  7.8000E-03  3.0000E+01
 12   2.0000E+06  1.4500E-02  2.0000E+01
 13   2.1000E+06  2.9500E-02  2.0000E+01
 14   2.2000E+06  5.2000E-02  2.0000E+01
 15   2.3000E+06  8.4000E-02  2.0000E+01
 16   2.3500E+06  9.4000E-02  2.0000E+01
```

17	2.4000E+06	8.6500E-02	2.0000E+01
18	2.5000E+06	7.3000E-02	1.0000E+01
19	2.6000E+06	7.9000E-02	1.0000E+01
20	2.7000E+06	6.6000E-02	1.0000E+01
21	2.8000E+06	1.2900E-01	1.0000E+01
22	2.9000E+06	9.3000E-02	1.0000E+01
23	3.0000E+06	1.2900E-01	1.0000E+01
24	3.1000E+06	1.8200E-01	1.0000E+01
25	3.2000E+06	1.4100E-01	1.0000E+01
26	3.2500E+06	1.3500E-01	1.0000E+01
27	3.3000E+06	1.8800E-01	1.0000E+01
28	3.3500E+06	2.2300E-01	1.0000E+01
29	3.4000E+06	2.1000E-01	1.0000E+01
30	3.4500E+06	2.0100E-01	1.0000E+01
31	3.5000E+06	2.0200E-01	1.0000E+01
32	3.6000E+06	2.3800E-01	1.0000E+01
33	3.7000E+06	2.1000E-01	1.0000E+01
34	3.8000E+06	1.6600E-01	1.0000E+01
35	3.9000E+06	1.8500E-01	1.0000E+01
36	4.0000E+06	2.7000E-01	1.0000E+01
37	4.1000E+06	3.0500E-01	1.0000E+01
38	4.2000E+06	3.3400E-01	1.0000E+01
39	4.3000E+06	3.4900E-01	1.0000E+01
40	4.4000E+06	3.2400E-01	1.0000E+01
41	4.6000E+06	2.5400E-01	1.0000E+01
42	4.8000E+06	2.3600E-01	1.0000E+01
43	5.0000E+06	2.3500E-01	1.0000E+01
44	5.2000E+06	2.3500E-01	1.0000E+01
45	5.4000E+06	2.4000E-01	1.0000E+01
46	5.6000E+06	2.7700E-01	1.0000E+01
47	5.8000E+06	3.0300E-01	1.0000E+01
48	6.0000E+06	3.1100E-01	1.0000E+01
49	6.5000E+06	3.1400E-01	1.0000E+01
50	7.0000E+06	3.1600E-01	1.0000E+01
51	7.5000E+06	3.2000E-01	1.0000E+01
52	8.0000E+06	3.2400E-01	1.0000E+01
53	8.5000E+06	3.3000E-01	1.0000E+01
54	9.0000E+06	3.4000E-01	1.0000E+01
55	9.5000E+06	3.5400E-01	1.0000E+01
56	1.0000E+07	3.7000E-01	1.0000E+01
57	1.0500E+07	3.8200E-01	1.0000E+01
58	1.1000E+07	3.8500E-01	1.0000E+01
59	1.1500E+07	3.8400E-01	1.0000E+01
60	1.2000E+07	3.6600E-01	8.0000E+00
61	1.2500E+07	3.2800E-01	8.0000E+00
62	1.3000E+07	3.0600E-01	7.0000E+00
63	1.3500E+07	2.8000E-01	7.0000E+00
64	1.4000E+07	2.5300E-01	5.0000E+00
65	1.4500E+07	2.2500E-01	5.0000E+00
66	1.5000E+07	2.1000E-01	5.0000E+00
67	1.6000E+07	1.6400E-01	5.0000E+00
68	1.7000E+07	1.2700E-01	5.0000E+00
69	1.8000E+07	1.0100E-01	6.0000E+00
70	1.9000E+07	8.7000E-02	6.0000E+00

```
71   2.0000E+07  7.7000E-02  6.0000E+00
  CORRELATION MATRIX
 1    0
 2    0 100
 3    0 100 100
 4    0 100 100 100
 5    0 100 100 100 100
 6    0 100 100 100 100 100
 7    0 100 100 100 100 100 100
 8    0   0   0   0   0   0   0 100
 9    0   0   0   0   0   0   0 100 100
10    0   0   0   0   0   0   0 100 100 100
11    0   0   0   0   0   0   0 100 100 100 100
12    0   0   0   0   0   0   0  64  64  64  64 100
13    0   0   0   0   0   0   0  64  64  64  64 100 100
14    0   0   0   0   0   0   0  64  64  64  64 100 100 100
15    0   0   0   0   0   0   0  64  64  64  64 100 100 100 100
16    0   0   0   0   0   0   0  64  64  64  64 100 100 100 100 100
17    0   0   0   0   0   0   0  64  64  64  64 100 100 100 100 100 100
18    0   0   0   0   0   0   0  51  51  51  51  64  64  64  64  64  64 100
19    0   0   0   0   0   0   0  51  51  51  51  64  64  64  64  64  64 100 100
20    0   0   0   0   0   0   0  51  51  51  51  64  64  64  64  64  64 100 100 100
21    0   0   0   0   0   0   0  51  51  51  51  64  64  64  64  64  64 100 100 100 100
22    0   0   0   0   0   0   0  51  51  51  51  64  64  64  64  64  64 100 100 100 100 100
23    0   0   0   0   0   0   0  41  41  41  41  51  51  51  51  51  51  64  64  64  64  64 100
24    0   0   0   0   0   0   0  41  41  41  41  51  51  51  51  51  51  64  64  64  64  64 100 100
25    0   0   0   0   0   0   0  41  41  41  41  51  51  51  51  51  51  64  64  64  64  64 100 100 100
26    0   0   0   0   0   0   0  41  41  41  41  51  51  51  51  51  51  64  64  64  64  64 100 100 100 100
27    0   0   0   0   0   0   0  41  41  41  41  51  51  51  51  51  51  64  64  64  64  64 100 100 100 100 100
28    0   0   0   0   0   0   0  41  41  41  41  51  51  51  51  51  51  64  64  64  64  64 100 100 100 100 100 100

29    0   0   0   0   0   0   0  41  41  41  41  51  51  51  51  51  51  64  64  64  64  64 100 100 100 100 100 100
        100
30    0   0   0   0   0   0   0  41  41  41  41  51  51  51  51  51  51  64  64  64  64  64 100 100 100 100 100 100
        100 100
31    0   0   0   0   0   0   0  33  33  33  33  41  41  41  41  41  41  51  51  51  51  51  64  64  64  64  64  64
        64  64 100
32    0   0   0   0   0   0   0  33  33  33  33  41  41  41  41  41  41  51  51  51  51  51  64  64  64  64  64  64
        64  64 100 100
33    0   0   0   0   0   0   0  33  33  33  33  41  41  41  41  41  41  51  51  51  51  51  64  64  64  64  64  64
        64  64 100 100 100
34    0   0   0   0   0   0   0  33  33  33  33  41  41  41  41  41  41  51  51  51  51  51  64  64  64  64  64  64
        64  64 100 100 100 100
35    0   0   0   0   0   0   0  33  33  33  33  41  41  41  41  41  41  51  51  51  51  51  64  64  64  64  64  64
        64  64 100 100 100 100 100
36    0   0   0   0   0   0   0  26  26  26  26  33  33  33  33  33  33  41  41  41  41  41  51  51  51  51  51  51
        51  51  64  64  64  64  64 100
37    0   0   0   0   0   0   0  26  26  26  26  33  33  33  33  33  33  41  41  41  41  41  51  51  51  51  51  51
        51  51  64  64  64  64  64 100 100
38    0   0   0   0   0   0   0  26  26  26  26  33  33  33  33  33  33  41  41  41  41  41  51  51  51  51  51  51
        51  51  64  64  64  64  64 100 100 100
39    0   0   0   0   0   0   0  26  26  26  26  33  33  33  33  33  33  41  41  41  41  41  51  51  51  51  51  51
        51  51  64  64  64  64  64 100 100 100 100
40    0   0   0   0   0   0   0  26  26  26  26  33  33  33  33  33  33  41  41  41  41  41  51  51  51  51  51  51
```

```
                    51  51  64  64  64  64  64 100 100 100 100 100
41    0   0   0   0  0   0   0  26  26  26  26  33  33  33  33  33  33  41  41  41  41  41  51  51  51  51  51  51
                    51  51  64  64  64  64  64 100 100 100 100 100 100
42    0   0   0   0  0   0   0  26  26  26  26  33  33  33  33  33  33  41  41  41  41  41  51  51  51  51  51  51
                    51  51  64  64  64  64  64 100 100 100 100 100 100 100
43    0   0   0   0  0   0   0  21  21  21  21  26  26  26  26  26  26  33  33  33  33  33  41  41  41  41  41  41
                    41  41  51  51  51  51  51  64  64  64  64  64  64  64 100
44    0   0   0   0  0   0   0  21  21  21  21  26  26  26  26  26  26  33  33  33  33  33  41  41  41  41  41  41
                    41  41  51  51  51  51  51  64  64  64  64  64  64  64 100 100
45    0   0   0   0  0   0   0  21  21  21  21  26  26  26  26  26  26  33  33  33  33  33  41  41  41  41  41  41
                    41  41  51  51  51  51  51  64  64  64  64  64  64  64 100 100 100
46    0   0   0   0  0   0   0  21  21  21  21  26  26  26  26  26  26  33  33  33  33  33  41  41  41  41  41  41
                    41  41  51  51  51  51  51  64  64  64  64  64  64  64 100 100 100 100
47    0   0   0   0  0   0   0  21  21  21  21  26  26  26  26  26  26  33  33  33  33  33  41  41  41  41  41  41
                    41  41  51  51  51  51  51  64  64  64  64  64  64  64 100 100 100 100 100
48    0   0   0   0  0   0   0  19  19  19  19  24  24  24  24  24  24  29  29  29  29  29  37  37  37  37  37  37
                    37  37  46  46  46  46  46  58  58  58  58  58  58  58  72  72  72  72  72 100
49    0   0   0   0  0   0   0  19  19  19  19  24  24  24  24  24  24  29  29  29  29  29  37  37  37  37  37  37
                    37  37  46  46  46  46  46  58  58  58  58  58  58  58  72  72  72  72  72 100 100
50    0   0   0   0  0   0   0  17  17  17  17  21  21  21  21  21  21  27  27  27  27  27  33  33  33  33  33  33
                    33  33  41  41  41  41  41  52  52  52  52  52  52  52  65  65  65  65  65  81  81 100
51    0   0   0   0  0   0   0  17  17  17  17  21  21  21  21  21  21  27  27  27  27  27  33  33  33  33  33  33
                    33  33  41  41  41  41  41  52  52  52  52  52  52  52  65  65  65  65  65  81  81 100 100
52    0   0   0   0  0   0   0  15  15  15  15  19  19  19  19  19  19  24  24  24  24  24  30  30  30  30  30  30
                    30  30  37  37  37  37  37  47  47  47  47  47  47  47  58  58  58  58  58  73  73  81  81 100
53    0   0   0   0  0   0   0  15  15  15  15  19  19  19  19  19  19  24  24  24  24  24  30  30  30  30  30  30
                    30  30  37  37  37  37  37  47  47  47  47  47  47  47  58  58  58  58  58  73  73  81  81 100 100
54    0   0   0   0  0   0   0  14  14  14  14  17  17  17  17  17  17  21  21  21  21  21  27  27  27  27  27  27
                    27  27  34  34  34  34  34  42  42  42  42  42  42  42  52  52  52  52  52  66  66  73  73  81  81
                   100
55    0   0   0   0  0   0   0  14  14  14  14  17  17  17  17  17  17  21  21  21  21  21  27  27  27  27  27  27
                    27  27  34  34  34  34  34  42  42  42  42  42  42  42  52  52  52  52  52  66  66  73  73  81  81
                   100 100
56    0   0   0   0  0   0   0  11  11  11  11  14  14  14  14  14  14  17  17  17  17  17  21  21  21  21  21  21
                    21  21  27  27  27  27  27  34  34  34  34  34  34  34  42  42  42  42  42  52  52  58  58  65  65
                    72  72 100
57    0   0   0   0  0   0   0  11  11  11  11  14  14  14  14  14  14  17  17  17  17  17  21  21  21  21  21  21
                    21  21  27  27  27  27  27  34  34  34  34  34  34  34  42  42  42  42  42  52  52  58  58  65  65
                    72  72 100 100
58    0   0   0   0  0   0   0   9   9   9   9  11  11  11  11  11  11  14  14  14  14  14  17  17  17  17  17  17
                    17  17  21  21  21  21  21  27  27  27  27  27  27  27  34  34  34  34  34  42  42  47  47  52  52
                    58  58  64  64 100
59    0   0   0   0  0   0   0   9   9   9   9  11  11  11  11  11  11  14  14  14  14  14  17  17  17  17  17  17
                    17  17  21  21  21  21  21  27  27  27  27  27  27  27  34  34  34  34  34  42  42  47  47  52  52
                    58  58  64  64 100 100
60    0   0   0   0  0   0   0   7   7   7   7   9   9   9   9   9   9  11  11  11  11  11  14  14  14  14  14  14
                    14  14  17  17  17  17  17  21  21  21  21  21  21  21  27  27  27  27  27  34  34  37  37  41  41
                    46  46  51  51  64  64 100
61    0   0   0   0  0   0   0   7   7   7   7   9   9   9   9   9   9  11  11  11  11  11  14  14  14  14  14  14
                    14  14  17  17  17  17  17  21  21  21  21  21  21  21  27  27  27  27  27  34  34  37  37  41  41
                    46  46  51  51  64  64 100 100
62    0   0   0   0  0   0   0   7   7   7   7   8   8   8   8   8   8  10  10  10  10  10  13  13  13  13  13  13
                    13  13  16  16  16  16  16  20  20  20  20  20  20  20  26  26  26  26  26  32  32  35  35  39  39
                    44  44  49  49  61  61  76  76 100
```

```
63   0   0   0   0   0   0   0   7   7   7   7   8   8   8   8   8   8  10  10  10  10  10  13  13  13  13  13  13
                13  13  16  16  16  16  16  20  20  20  20  20  20  20  26  26  26  26  26  32  32  35  35  39  39
                44  44  49  49  61  61  76  76 100 100
64   0   0   0   0   0   0   0   6   6   6   6   8   8   8   8   8   8  10  10  10  10  10  12  12  12  12  12  12
                12  12  16  16  16  16  16  19  19  19  19  19  19  19  24  24  24  24  24  30  30  34  34  37  37
                42  42  46  46  58  58  72  72  90  90 100
65   0   0   0   0   0   0   0   6   6   6   6   8   8   8   8   8   8  10  10  10  10  10  12  12  12  12  12  12
                12  12  16  16  16  16  16  19  19  19  19  19  19  19  24  24  24  24  24  30  30  34  34  37  37
                42  42  46  46  58  58  72  72  90  90 100 100
66   0   0   0   0   0   0   0   6   6   6   6   8   8   8   8   8   8  10  10  10  10  10  12  12  12  12  12  12
                12  12  16  16  16  16  16  19  19  19  19  19  19  19  24  24  24  24  24  30  30  34  34  37  37
                42  42  46  46  58  58  72  72  90  90 100 100 100
67   0   0   0   0   0   0   0   6   6   6   6   8   8   8   8   8   8   9   9   9   9   9  12  12  12  12  12  12
                12  12  15  15  15  15  15  18  18  18  18  18  18  18  23  23  23  23  23  29  29  32  32  36  36
                40  40  44  44  55  55  69  69  86  86  90  90  90 100
68   0   0   0   0   0   0   0   6   6   6   6   8   8   8   8   8   8   9   9   9   9   9  12  12  12  12  12  12
                12  12  15  15  15  15  15  18  18  18  18  18  18  18  23  23  23  23  23  29  29  32  32  36  36
                40  40  44  44  55  55  69  69  86  86  90  90  90 100 100
69   0   0   0   0   0   0   0   6   6   6   6   7   7   7   7   7   7   9   9   9   9   9  11  11  11  11  11  11
                11  11  14  14  14  14  14  18  18  18  18  18  18  18  22  22  22  22  22  27  27  30  30  34  34
                38  38  42  42  52  52  65  65  81  81  86  86  86  90  90 100
70   0   0   0   0   0   0   0   6   6   6   6   7   7   7   7   7   7   9   9   9   9   9  11  11  11  11  11  11
                11  11  14  14  14  14  14  18  18  18  18  18  18  18  22  22  22  22  22  27  27  30  30  34  34
                38  38  42  42  52  52  65  65  81  81  86  86  86  90  90 100 100
71   0   0   0   0   0   0   0   6   6   6   6   7   7   7   7   7   7   9   9   9   9   9  11  11  11  11  11  11
                11  11  14  14  14  14  14  18  18  18  18  18  18  18  22  22  22  22  22  27  27  30  30  34  34
                38  38  42  42  52  52  65  65  81  81  86  86  86  90  90 100 100 100
```

THINNED DATA

I	E(I)	X(E(I))	STD(X(E(I)))
1	9.5700E+05	0.0000E-01	0.0000E-01
2	1.0000E+06	1.2800E-05	3.0000E+01
3	1.3000E+06	1.1200E-04	3.0000E+01
4	1.6000E+06	9.5000E-04	3.0000E+01
5	1.9000E+06	7.8000E-03	3.0000E+01
6	2.0000E+06	1.4500E-02	2.0000E+01
7	2.5000E+06	7.3000E-02	1.0000E+01
8	3.0000E+06	1.2900E-01	1.0000E+01
9	3.5000E+06	2.0200E-01	1.0000E+01
10	4.0000E+06	2.7000E-01	1.0000E+01
11	5.0000E+06	2.3500E-01	1.0000E+01
12	6.0000E+06	3.1100E-01	1.0000E+01
13	7.0000E+06	3.1600E-01	1.0000E+01
14	8.0000E+06	3.2400E-01	1.0000E+01
15	9.0000E+06	3.4000E-01	1.0000E+01
16	1.0000E+07	3.7000E-01	1.0000E+01
17	1.1000E+07	3.8500E-01	1.0000E+01
18	1.2000E+07	3.6600E-01	8.0000E+00
19	1.3000E+07	3.0600E-01	7.0000E+00
20	1.4000E+07	2.5300E-01	5.0000E+00
21	1.6000E+07	1.6400E-01	5.0000E+00
22	1.8000E+07	1.0100E-01	6.0000E+00
23	2.0000E+07	7.7000E-02	6.0000E+00

CORRELATION MATRIX

1	0

2	0	100																						
3	0	100	100																					
4	0	0	0	100																				
5	0	0	0	100	100																			
6	0	0	0	64	64	100																		
7	0	0	0	51	51	64	100																	
8	0	0	0	41	41	51	64	100																
9	0	0	0	33	33	41	51	64	100															
10	0	0	0	26	26	33	41	51	64	100														
11	0	0	0	21	21	26	33	41	51	64	100													
12	0	0	0	19	19	24	29	37	46	58	72	100												
13	0	0	0	17	17	21	27	33	41	52	65	81	100											
14	0	0	0	15	15	19	24	30	37	47	58	73	81	100										
15	0	0	0	14	14	17	21	27	34	42	52	66	73	81	100									
16	0	0	0	11	11	14	17	21	27	34	42	52	58	65	72	100								
17	0	0	0	9	9	11	14	17	21	27	34	42	47	52	58	64	100							
18	0	0	0	7	7	9	11	14	17	21	27	34	37	41	46	51	64	100						
19	0	0	0	7	7	8	10	13	16	20	26	32	35	39	44	49	61	76	100					
20	0	0	0	6	6	8	10	12	16	19	24	30	34	37	42	46	58	72	90	100				
21	0	0	0	6	6	8	9	12	15	18	23	29	32	36	40	44	55	69	86	90	100			
22	0	0	0	6	6	7	9	11	14	18	22	27	30	34	38	42	52	65	81	86	90	100		
23	0	0	0	6	6	7	9	11	14	18	22	27	30	34	38	42	52	65	81	86	90	100	100	

CORRELATION MATRIX OF RECTANGULAR BLOCK 2 1

SET 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
SET 2																									
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	4	5	4	4	4	3	3
3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	2	3	3	6	7	6	6	6	5	5
4	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2	3	3	5	8	9	8	8	8	7	7
5	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2	3	4	6	10	11	9	10	10	9	9
6	0	0	0	0	0	0	0	1	1	1	1	1	2	2	3	4	5	7	12	14	11	12	12	10	10
7	0	0	0	0	0	0	1	1	1	1	1	2	2	3	4	5	6	9	16	17	14	16	16	13	13
8	0	0	0	0	0	0	1	1	1	2	2	3	3	4	5	7	9	13	22	25	20	22	22	19	19
9	0	0	0	0	0	1	1	2	3	3	4	5	7	8	11	14	18	25	45	49	40	47	47	40	40
10	0	0	0	0	0	1	2	2	3	3	4	5	7	9	11	14	19	26	46	55	43	37	37	34	34
11	0	0	0	0	0	1	2	2	3	4	4	6	7	9	11	15	20	27	47	53	54	53	45	40	40
12	0	0	0	0	0	1	1	2	2	3	4	5	6	8	10	12	16	23	39	49	33	52	52	35	35
13	0	0	0	0	0	1	1	1	2	2	3	4	4	6	7	9	12	17	29	37	35	31	50	42	42
14	0	0	0	0	0	0	1	1	2	2	2	3	4	5	6	8	10	13	18	30	40	40	46	41	71
15	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2	3	4	5	7	11	14	14	16	16	23
16	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2	3	4	5	7	11	14	14	16	16	23

THIS IS FOR14.DAT. THESE DATA WERE OUTPUT FROM GLUCS.F4 DURING THE SECOND RUN.

EXPERIMENTAL DATA

I	E(I)	X(E(I))	STD(X(E(I)))
1	6.7000E+06	8.8400E-02	3.0000E+00
2	7.0100E+06	9.2300E-02	2.9000E+00
3	7.5000E+06	1.1050E-01	2.9000E+00
4	7.8000E+06	1.4010E-01	2.9000E+00
5	8.1000E+06	1.4210E-01	3.0000E+00
6	8.2700E+06	1.4010E-01	3.0000E+00
7	8.4000E+06	1.5070E-01	3.0000E+00
8	8.6100E+06	1.6150E-01	3.0000E+00
9	8.8100E+06	1.6660E-01	3.0000E+00
10	9.0200E+06	1.7920E-01	2.9000E+00
11	9.6300E+06	1.7970E-01	2.8000E+00
12	1.0030E+07	1.9560E-01	2.8000E+00
13	1.0130E+07	1.9770E-01	2.7000E+00
14	1.0400E+07	2.0680E-01	2.7000E+00
15	1.0620E+07	2.1350E-01	2.7000E+00
16	1.0810E+07	2.1150E-01	2.7000E+00
17	1.1000E+07	2.2730E-01	2.7000E+00
18	1.1220E+07	2.3180E-01	2.6000E+00
19	1.1500E+07	2.4680E-01	3.1000E+00
20	1.1610E+07	2.6170E-01	3.0000E+00
21	1.2100E+07	2.8330E-01	2.9000E+00
22	1.2590E+07	3.3330E-01	2.7000E+00
23	1.3080E+07	3.7380E-01	3.5000E+00
24	1.3580E+07	4.1520E-01	3.5000E+00
25	1.4500E+07	4.8670E-01	3.6000E+00
26	1.0130E+07	1.7800E-02	5.7000E+01
27	1.0400E+07	4.5500E-02	1.1000E+01
28	1.0500E+07	9.0700E-02	1.4000E+01
29	1.0620E+07	1.9300E-01	6.5000E+00
30	1.0810E+07	2.1300E-01	5.9000E+00
31	1.1000E+07	4.2100E-01	4.2000E+00
32	1.1220E+07	5.1800E-01	6.8000E+00
33	1.1500E+07	7.8400E-01	5.9000E+00
34	1.1610E+07	9.7700E-01	5.3000E+00
35	1.2000E+07	1.2460E+00	5.3000E+00
36	1.2100E+07	1.3610E+00	5.1000E+00
37	1.2600E+07	1.9550E+00	4.6000E+00
38	1.3080E+07	2.5410E+00	6.5000E+00
39	1.3580E+07	3.1160E+00	6.9000E+00
40	1.4500E+07	4.3050E+00	3.1000E+00
41	1.2530E+07	1.0090E+00	5.4000E+00
42	1.3580E+07	1.0820E+00	3.4000E+00
43	1.3890E+07	1.0730E+00	3.4000E+00
44	1.4240E+07	1.0000E+00	3.6000E+00
45	1.4680E+07	9.8200E-01	3.7000E+00
46	1.4760E+07	9.7300E-01	3.7000E+00
47	1.6100E+07	7.4900E-01	5.5000E+00
48	1.6600E+07	6.9300E-01	5.6000E+00
49	1.7500E+07	6.1300E-01	5.9000E+00
50	1.8500E+07	5.1300E-01	5.5000E+00

51	1.9100E+07	4.8100E-01	5.7000E+00
52	1.9800E+07	4.3900E-01	5.4000E+00
53	1.3580E+07	8.9800E-01	3.0000E+00
54	1.3890E+07	9.3400E-01	3.0000E+00
55	1.4250E+07	9.6900E-01	3.0000E+00
56	1.4680E+07	1.0080E+00	3.1000E+00
57	1.4740E+07	1.0300E+00	3.0000E+00
58	1.6300E+07	1.0430E+00	5.9000E+00
59	1.6500E+07	1.1190E+00	5.5000E+00
60	1.6600E+07	1.0500E+00	5.9000E+00
61	1.7900E+07	1.1040E+00	6.1000E+00
62	1.8300E+07	1.1170E+00	6.0000E+00
63	1.8500E+07	1.1600E+00	5.8000E+00
64	1.9800E+07	1.0590E+00	5.8000E+00

I CORRELATION MATRIX

1	100																												
2	46	100																											
3	46	48	100																										
4	46	48	48	100																									
5	44	46	46	46	100																								
6	44	46	46	46	44	100																							
7	44	46	46	46	44	44	100																						
8	44	46	46	46	44	44	44	100																					
9	44	46	46	46	44	44	44	44	100																				
10	46	48	48	48	46	46	46	46	46	100																			
11	48	49	49	49	48	48	48	48	48	49	100																		
12	48	49	49	49	48	48	48	48	48	49	51	100																	
13	49	51	51	51	49	49	49	49	49	51	53	53	100																
14	49	51	51	51	49	49	49	49	49	51	53	53	55	100															
15	49	51	51	51	49	49	49	49	49	51	53	53	55	55	100														
16	49	51	51	51	49	49	49	49	49	51	53	53	55	55	55	100													
17	49	51	51	51	49	49	49	49	49	51	53	53	55	55	55	55	100												
18	51	53	53	53	51	51	51	51	51	53	55	55	57	57	57	57	57	100											
19	43	44	44	44	43	43	43	43	43	44	46	46	48	48	48	48	48	50	100										
20	44	46	46	46	44	44	44	44	44	46	48	48	49	49	49	49	49	51	43	100									
21	46	48	48	48	46	46	46	46	46	48	49	49	51	51	51	51	51	53	44	46	100								
22	49	51	51	51	49	49	49	49	49	51	53	53	55	55	55	55	55	57	48	49	51	100							
23	38	39	39	39	38	38	38	38	38	39	41	41	42	42	42	42	42	44	37	38	39	42	100						
24	38	39	39	39	38	38	38	38	38	39	41	41	42	42	42	42	42	44	37	38	39	42	33	100					
25	37	38	38	38	37	37	37	37	37	38	40	40	41	41	41	41	41	43	36	37	38	41	32	32	100				
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	100			
27	6	6	6	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	6	6	6	7	5	5	5	1	100		
28	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	1	3	100	
29	10	11	11	11	10	10	10	10	10	11	11	11	11	11	11	11	11	12	10	10	11	11	9	9	9	1	6	4	
					100																								
30	11	12	12	12	11	11	11	11	11	12	12	12	13	13	13	13	13	13	11	11	12	13	10	10	9	1	6	5	
					10	100																							
31	16	16	16	16	16	16	16	16	16	16	17	17	18	18	18	18	18	18	15	16	16	18	14	14	13	2	9	7	
					15	16	100																						
32	10	10	10	10	10	10	10	10	10	10	11	11	11	11	11	11	11	11	9	10	10	11	8	8	8	1	5	4	
					9	10	14	100																					
33	11	12	12	12	11	11	11	11	11	12	12	12	13	13	13	13	13	13	11	11	12	13	10	10	9	1	6	5	
					10	11	16	10	100																				

```
34  13  13  13  13  13  13  13  13  13  13  13  13  14  14  14  14  14  15  12  13  13  14  11  11  10   1   7   5
                    12  13  18  11  13 100
35  13  13  13  13  13  13  13  13  13  13  13  13  14  14  14  14  14  15  12  13  13  14  11  11  10   1   7   5
                    12  13  18  11  13  14 100
36  13  14  14  14  13  13  13  13  13  14  14  14  15  15  15  15  15  15  13  13  14  15  11  11  11   1   7   6
                    12  13  19  12  13  15  15 100
37  14  15  15  15  14  14  14  14  14  15  16  16  16  16  16  16  16  17  14  14  15  16  12  12  12   2   8   6
                    13  15  21  13  15  16  16  17 100
38  10  11  11  11  10  10  10  10  10  10  11  11  11  11  11  11  11  11  12  10  10  11  11   9   9   9   1   6   4
                     9  10  15   9  10  12  12  12  13 100
39  10  10  10  10  10  10  10  10  10  10  10  10  11  11  11  11  11  11   9  10  10  11   8   8   8   1   5   4
                     9  10  14   9  10  11  11  11  13   9 100
40  22  22  22  22  22  22  22  22  22  22  23  23  24  24  24  24  24  25  21  22  22  24  18  18  18   2  12   9
                    20  22  31  19  22  24  24  25  28  20  19 100
41   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
42   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
43   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
44   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
45   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
46   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
47   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
48   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
49   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
50   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
51   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
52   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
53   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0 100
54   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                   100
55   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0 100
56   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0 100
57   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0 100
58   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
                0  0  0  0 100
59   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0 100
60   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0 100
61   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0 100
62   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0 100
63   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0  0 100
64   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                0  0  0  0  0  0  0  0  0  0 100
```

EVALUATED DATA

I	E(I)	X(E(I))
1	2.9655E+06	0.0000E-01
2	3.0000E+06	7.0107E-09
3	4.0000E+06	6.0092E-06
4	4.2500E+06	3.2049E-05
5	4.5000E+06	1.4039E-04
6	5.0000E+06	1.1054E-03
7	5.5000E+06	4.7336E-03
8	6.0000E+06	1.2609E-02
9	6.4000E+06	1.9410E-02
10	7.0000E+06	2.9027E-02
11	8.0000E+06	4.3187E-02
12	9.0000E+06	5.5784E-02
13	1.0000E+07	7.0535E-02
14	1.1000E+07	8.8454E-02
15	1.2000E+07	1.0515E-01
16	1.2400E+07	1.1082E-01
17	1.2800E+07	1.1402E-01
18	1.3400E+07	1.1556E-01
19	1.4200E+07	1.1250E-01
20	1.5000E+07	1.0319E-01
21	1.6000E+07	8.5812E-02
22	1.7000E+07	7.1227E-02
23	1.8000E+07	6.0431E-02
24	1.9000E+07	5.2429E-02
25	2.0000E+07	4.9874E-02
26	1.0063E+07	0.0000E-01
27	1.0500E+07	3.6601E-02
28	1.1000E+07	1.5762E-01
29	1.1500E+07	3.1860E-01
30	1.2000E+07	4.8292E-01
31	1.2500E+07	6.0966E-01
32	1.3000E+07	7.1026E-01
33	1.3500E+07	7.9800E-01

34	1.4500E+07	9.3112E-01
35	1.5500E+07	1.0026E+00
36	1.6500E+07	1.0456E+00
37	1.7500E+07	1.0769E+00
38	1.8500E+07	1.0599E+00
39	1.9200E+07	1.0088E+00
40	1.9600E+07	1.0003E+00
41	2.0000E+07	9.2790E-01
42	9.5700E+05	0.0000E-01
43	1.0000E+06	1.2800E-05
44	1.3000E+06	1.1200E-04
45	1.6000E+06	9.5000E-04
46	1.9000E+06	7.8000E-03
47	2.0000E+06	1.4500E-02
48	2.5000E+06	7.3000E-02
49	3.0000E+06	1.2900E-01
50	3.5000E+06	2.0200E-01
51	4.0000E+06	2.7000E-01
52	5.0000E+06	2.3500E-01
53	6.0000E+06	3.1100E-01
54	7.0000E+06	3.1600E-01
55	8.0000E+06	3.2400E-01
56	9.0000E+06	3.4000E-01
57	1.0000E+07	3.7000E-01
58	1.1000E+07	3.8500E-01
59	1.2000E+07	3.6600E-01
60	1.3000E+07	3.0600E-01
61	1.4000E+07	2.5300E-01
62	1.6000E+07	1.6400E-01
63	1.8000E+07	1.0100E-01
64	2.0000E+07	7.7000E-02

THEORETICAL DATA

I	E(I)	REXP(E(I))	RTHEORY(E(I))
1	6.7000E+06	8.8400E-02	7.7006E-02
2	7.0100E+06	9.2300E-02	9.2282E-02
3	7.5000E+06	1.1050E-01	1.1283E-01
4	7.8000E+06	1.4010E-01	1.2517E-01
5	8.1000E+06	1.4210E-01	1.3651E-01
6	8.2700E+06	1.4010E-01	1.4190E-01
7	8.4000E+06	1.5070E-01	1.4596E-01
8	8.6100E+06	1.6150E-01	1.5242E-01
9	8.8100E+06	1.6660E-01	1.5845E-01
10	9.0200E+06	1.7920E-01	1.6465E-01
11	9.6300E+06	1.7970E-01	1.8132E-01
12	1.0030E+07	1.9560E-01	1.9185E-01
13	1.0130E+07	1.9770E-01	1.9590E-01
14	1.0400E+07	2.0680E-01	2.0666E-01
15	1.0620E+07	2.1350E-01	2.1525E-01
16	1.0810E+07	2.1150E-01	2.2256E-01
17	1.1000E+07	2.2730E-01	2.2975E-01
18	1.1220E+07	2.3180E-01	2.4192E-01
19	1.1500E+07	2.4680E-01	2.5779E-01
20	1.1610E+07	2.6170E-01	2.6416E-01
21	1.2100E+07	2.8330E-01	2.9602E-01

22	1.2590E+07	3.3330E-01	3.3981E-01
23	1.3080E+07	3.7380E-01	3.8023E-01
24	1.3580E+07	4.1520E-01	4.1732E-01
25	1.4500E+07	4.8670E-01	4.7241E-01
26	1.0130E+07	1.7800E-02	1.5087E-02
27	1.0400E+07	4.5500E-02	7.5068E-02
28	1.0500E+07	9.0700E-02	9.6956E-02
29	1.0620E+07	1.9300E-01	1.7307E-01
30	1.0810E+07	2.1300E-01	2.9212E-01
31	1.1000E+07	4.2100E-01	4.0940E-01
32	1.1220E+07	5.1800E-01	5.9989E-01
33	1.1500E+07	7.8400E-01	8.4847E-01
34	1.1610E+07	9.7700E-01	9.5003E-01
35	1.2000E+07	1.2460E+00	1.3195E+00
36	1.2100E+07	1.3610E+00	1.4119E+00
37	1.2600E+07	1.9550E+00	1.9084E+00
38	1.3080E+07	2.5410E+00	2.4002E+00
39	1.3580E+07	3.1160E+00	2.9378E+00
40	1.4500E+07	4.3050E+00	4.0352E+00
41	1.2530E+07	1.0090E+00	1.0262E+00
42	1.3580E+07	1.0820E+00	1.0538E+00
43	1.3890E+07	1.0730E+00	1.0429E+00
44	1.4240E+07	1.0000E+00	1.0278E+00
45	1.4680E+07	9.8200E-01	9.8078E-01
46	1.4760E+07	9.7300E-01	9.7224E-01
47	1.6100E+07	7.4900E-01	7.7382E-01
48	1.6600E+07	6.9300E-01	7.0692E-01
49	1.7500E+07	6.1300E-01	6.0389E-01
50	1.8500E+07	5.1300E-01	5.1766E-01
51	1.9100E+07	4.8100E-01	4.7862E-01
52	1.9800E+07	4.3900E-01	4.6221E-01
53	1.3580E+07	8.9800E-01	8.6847E-01
54	1.3890E+07	9.3400E-01	9.1279E-01
55	1.4250E+07	9.6900E-01	9.6426E-01
56	1.4680E+07	1.0080E+00	1.0138E+00
57	1.4740E+07	1.0300E+00	1.0184E+00
58	1.6300E+07	1.0430E+00	1.1137E+00
59	1.6500E+07	1.1190E+00	1.1229E+00
60	1.6600E+07	1.0500E+00	1.1263E+00
61	1.7900E+07	1.1040E+00	1.1493E+00
62	1.8300E+07	1.1170E+00	1.1420E+00
63	1.8500E+07	1.1600E+00	1.1383E+00
64	1.9800E+07	1.0590E+00	1.0354E+00

SENSITIVITY MATRIX G-ONLY NONZERO VALUES GIVEN:

ROW I	COLUMN J	G(I,J)
1	9	1.5898E+00
1	10	1.5898E+00
2	10	3.1321E+00
3	10	1.5625E+00
4	10	6.2035E-01
2	11	3.1638E-02
3	11	1.5625E+00
4	11	2.4814E+00
5	11	2.7641E+00

6	11	2.2234E+00
7	11	1.8160E+00
8	11	1.1685E+00
9	11	5.6387E-01
5	12	3.0713E-01
6	12	8.2237E-01
7	12	1.2107E+00
8	12	1.8277E+00
9	12	2.4038E+00
10	12	2.8773E+00
11	12	1.0309E+00
10	13	5.8720E-02
11	13	1.7554E+00
12	13	2.6184E+00
13	13	2.3390E+00
14	13	1.5957E+00
15	13	1.0018E+00
16	13	4.9719E-01
12	14	8.0983E-02
13	14	3.4951E-01
14	14	1.0638E+00
15	14	1.6346E+00
16	14	2.1196E+00
17	14	2.5974E+00
18	14	2.0482E+00
19	14	1.3316E+00
20	14	1.0444E+00
18	15	5.7770E-01
19	15	1.3316E+00
20	15	1.6336E+00
21	15	2.0833E+00
21	16	6.9444E-01
22	16	1.5880E+00
41	16	6.1922E+00
22	17	1.4368E+00
23	17	1.7674E+00
41	17	2.9814E+00
23	18	1.5465E+00
24	18	2.8155E+00
42	18	7.1095E+00
43	18	3.5548E+00
24	19	8.1741E-01
25	19	2.7086E+00
41	19	-5.8835E+00
42	19	-3.9778E+00
43	19	-3.6066E-01
44	19	2.8223E+00
45	19	-1.9539E+00
46	19	-2.8223E+00
47	19	-4.4367E+00
48	19	-4.0531E+00
49	19	-3.4624E+00
50	19	-2.9680E+00
51	19	-2.7441E+00

52	19	-2.6501E+00
25	20	1.6251E+00
41	20	-3.5301E+00
42	20	-3.6251E+00
43	20	-3.5877E+00
44	20	-3.0769E+00
45	20	2.1302E+00
46	20	3.0769E+00
47	20	-2.6620E+00
48	20	-2.4319E+00
49	20	-2.0774E+00
50	20	-1.7808E+00
51	20	-1.6465E+00
52	20	-1.5900E+00
47	21	8.2562E+00
48	21	3.6694E+00
47	22	9.1736E-01
48	22	5.5041E+00
49	22	4.5868E+00
49	23	4.5868E+00
50	23	4.5868E+00
50	24	4.5868E+00
51	24	8.2562E+00
52	24	1.8347E+00
51	25	9.1736E-01
52	25	7.3389E+00
26	26	2.2763E+00
27	26	6.0860E-01
26	27	4.1220E-01
27	27	2.0510E+00
28	27	2.6490E+00
29	27	2.0037E+00
30	27	9.9437E-01
29	28	6.3274E-01
30	28	1.6224E+00
31	28	2.5974E+00
32	28	1.4705E+00
32	29	1.1554E+00
33	29	2.6631E+00
34	29	2.0889E+00
34	30	5.8916E-01
35	30	2.7322E+00
36	30	2.2222E+00
36	31	5.5556E-01
37	31	. 4242E+00
37	32	6.0606E-01
38	32	2.7837E+00
38	33	5.3022E-01
39	33	3.3423E+00
53	33	9.8806E-01
54	33	6.5513E-01
55	33	2.6849E-01
39	34	2.9063E-01
40	34	4.3337E+00

53	34	-8.4680E-01
54	34	-5.6146E-01
55	34	-2.3011E-01
56	34	-2.0816E-01
57	34	-2.7754E-01
58	34	-1.1961E+00
59	34	-1.2060E+00
60	34	-1.2096E+00
61	34	-1.2343E+00
62	34	-1.2264E+00
63	34	-1.2225E+00
64	34	-1.1120E+00
56	35	1.9332E-01
57	35	2.5775E-01
58	35	2.1480E-01
58	36	8.5918E-01
59	36	1.0740E+00
60	36	9.6658E-01
60	37	1.0740E-01
61	37	6.4439E-01
62	37	2.1480E-01
61	38	4.2959E-01
62	38	8.5918E-01
63	38	1.0740E+00
64	40	5.3699E-01
64	41	5.3699E-01
1	53	-7.3456E-02
1	54	-1.7140E-01
2	54	-2.8904E-01
3	54	-1.7630E-01
4	54	-7.7649E-02
2	55	-2.9196E-03
3	55	-1.7630E-01
4	55	-3.1060E-01
5	55	-3.7732E-01
6	55	-3.1550E-01
7	55	-2.6506E-01
8	55	-1.7810E-01
9	55	-8.9343E-02
5	56	-4.1925E-02
6	56	-1.1669E-01
7	56	-1.7671E-01
8	56	-2.7857E-01
9	56	-3.8088E-01
10	56	-4.7374E-01
11	56	-1.8693E-01
10	57	-9.6681E-03
11	57	-3.1829E-01
12	57	-5.0236E-01
13	57	-4.5821E-01
14	57	-3.2977E-01
15	57	-2.1565E-01
16	57	-1.1065E-01
26	57	-3.5289E-02

27	57	-1.1979E-01
28	57	-1.2842E-01
29	57	-1.7339E-01
30	57	-1.4524E-01
12	58	-1.5537E-02
13	58	-6.8468E-02
14	58	-2.1985E-01
15	58	-3.5185E-01
16	58	-4.7172E-01
17	58	-5.9675E-01
18	58	-4.9550E-01
19	58	-3.4327E-01
20	58	-2.7589E-01
26	58	-5.2730E-03
27	58	-7.9859E-02
28	58	-1.2842E-01
29	58	-2.8290E-01
30	58	-6.1917E-01
31	58	-1.0634E+00
32	58	-1.2287E+00
33	58	-1.1298E+00
34	58	-9.9224E-01
18	59	-1.3976E-01
19	59	-3.4327E-01
20	59	-4.3152E-01
21	59	-7.4005E-01
22	59	-4.2142E-01
32	59	-3.4656E-01
33	59	-1.1298E+00
34	59	-1.5520E+00
35	59	-3.6051E+00
36	59	-3.5296E+00
37	59	-2.3132E+00
21	60	-8.2[illegible]
22	60	-6.06[illegible]
23	60	-1.1592[illegible]
24	60	-6.3676E-01
36	60	-3.9218E-01
37	60	-3.4699E+00
38	60	-7.3178E+00
39	60	-4.4825E+00
23	61	-1.0080E-01
24	61	-8.7933E-01
25	61	-1.5355E+00
38	61	-6.3633E-01
39	61	-6.1902E+00
40	61	-1.3115E+01
25	62	-5.1182E-01
40	62	-4.3718E+00

THIS IS FOR17.DAT. THESE DATA WERE OUTPUT FROM OUTPUT.F4 DURING THE
SECOND RUN.
FINAL THINNED DATA OF # 1 SET

I	E(I)	X(E(I))	STD(X(E(I)))
1	2.9655E+06	0.0000E-01	0.0000E-01
2	3.0000E+06	7.0635E-09	9.9046E+00
3	4.5000E+06	1.4229E-04	8.8353E+00
4	5.0000E+06	1.1319E-03	7.7048E+00
5	5.5000E+06	4.8990E-03	6.5466E+00
6	6.0000E+06	1.3148E-02	5.4078E+00
7	6.4000E+06	2.0951E-02	4.8160E+00
8	7.0000E+06	2.9138E-02	4.2784E+00
9	8.0000E+06	4.3523E-02	3.5474E+00
10	9.0000E+06	5.6405E-02	3.4831E+00
11	1.0000E+07	7.0824E-02	3.4319E+00
12	1.1000E+07	8.8335E-02	2.5377E+00
13	1.2000E+07	1.0466E-01	2.3561E+00
14	1.2400E+07	1.1012E-01	2.2816E+00
15	1.2800E+07	1.1345E-01	1.8991E+00
16	1.3400E+07	1.1545E-01	1.4157E+00
17	1.4200E+07	1.1253E-01	1.0877E+00
18	1.5000E+07	1.0319E-01	9.8237E-01
19	1.6000E+07	8.5661E-02	1.1077E+00
20	1.7000E+07	7.1159E-02	1.0187E+00
21	1.8000E+07	6.0397E-02	1.0545E+00
22	1.9000E+07	5.2354E-02	1.1972E+00
23	2.0000E+07	4.9802E-02	1.1972E+00

CORRELATION MATRIX

1	0																				
2	0	100																			
3	0	25	100																		
4	0	15	29	100																	
5	0	10	20	40	100																
6	0	8	16	31	53	100															
7	0	6	12	24	40	59	100														
8	0	5	9	19	32	47	62	100													
9	0	4	8	16	26	39	52	66	100												
10	0	3	6	12	21	31	42	53	65	100											
11	0	2	5	10	16	24	32	41	50	62	100										
12	0	2	4	7	12	18	24	31	38	47	59	100									
13	0	1	3	5	9	13	18	23	28	34	43	56	100								
14	0	1	2	4	7	10	14	17	21	26	33	42	57	100							
15	0	1	2	3	5	8	10	13	16	19	24	32	43	45	100						
16	0	1	1	2	4	6	8	9	11	14	17	23	29	37	47	100					
17	0	0	1	2	3	5	6	8	10	12	15	19	25	31	38	50	100				
18	0	0	1	1	2	3	5	6	7	9	11	14	18	23	27	36	46	100			
19	0	0	1	1	2	3	4	5	6	8	10	13	17	21	25	33	49	63	100		
20	0	0	1	1	2	3	4	5	7	8	10	13	17	21	26	35	50	70	78	100	
21	0	0	1	1	2	3	4	5	7	8	10	13	17	21	26	34	49	69	73	80	100
22	0	0	1	1	2	3	4	5	6	7	9	12	15	19	23	30	45	59	66	70	70	100
23	0	0	1	1	2	3	4	5	6	7	9	12	15	19	23	30	45	59	66	70	70	100	100

FINAL THINNED DATA OF # 2 SET

I	E(I)	X(E(I))	STD(X(E(I)))
1	1.0063E+07	0.0000E-01	0.0000E-01

2	1.0500E+07	2.6597E-02	8.5978E+00
3	1.1000E+07	1.4915E-01	4.0446E+00
4	1.1500E+07	3.0948E-01	3.5021E+00
5	1.2000E+07	4.7605E-01	3.0999E+00
6	1.2500E+07	6.1174E-01	2.8935E+00
7	1.3000E+07	7.1900E-01	2.7002E+00
8	1.3500E+07	8.1276E-01	1.9195E+00
9	1.4500E+07	9.3628E-01	1.2105E+00
10	1.5500E+07	1.0034E+00	1.1822E+00
11	1.6500E+07	1.0456E+00	1.1722E+00
12	1.7500E+07	1.0781E+00	1.3647E+00
13	1.8500E+07	1.0603E+00	1.8705E+00
14	1.9200E+07	1.0087E+00	1.8896E+00
15	1.9600E+07	1.0085E+00	3.3097E+00
16	2.0000E+07	9.3547E-01	3.3099E+00

CORRELATION MATRIX

1	0														
2	0	100													
3	0	22	100												
4	0	24	43	100											
5	0	20	35	51	100										
6	0	16	27	38	48	100									
7	0	11	20	26	32	42	100								
8	0	9	16	20	25	31	43	100							
9	0	9	16	19	22	26	33	54	100						
10	0	7	12	14	17	19	24	38	62	100					
11	0	7	12	15	18	20	25	41	68	78	100				
12	0	6	10	13	15	17	21	34	59	62	60	100			
13	0	5	8	10	12	14	17	27	47	45	51	21	100		
14	0	5	8	9	11	13	16	25	43	44	45	50	8	100	
15	0	2	4	5	6	7	8	13	24	20	21	20	13	28	100
16	0	2	4	5	6	7	8	13	24	20	21	20	13	28	100 100

FINAL THINNED DATA OF # 3 SET

I	E(I)	X(E(I))	STD(X(E(I)))
1	9.5700E+05	0.0000E-01	0.0000E-01
2	1.0000E+06	1.2800E-05	3.0000E+01
3	1.6000E+06	9.1523E-04	3.0739E+01
4	2.0000E+06	1.4058E-02	2.0212E+01
5	2.5000E+06	7.1608E-02	9.8709E+00
6	3.0000E+06	1.2593E-01	9.7311E+00
7	3.5000E+06	1.9598E-01	9.4890E+00
8	4.0000E+06	2.5995E-01	9.0653E+00
9	5.0000E+06	2.2406E-01	8.3099E+00
10	6.0000E+06	2.8437E-01	6.7765E+00
11	7.0000E+06	3.1804E-01	4.6065E+00
12	8.0000E+06	3.2503E-01	4.0676E+00
13	9.0000E+06	3.3497E-01	4.1552E+00
14	1.0000E+07	3.7918E-01	3.7998E+00
15	1.1000E+07	3.0646E-01	2.9631E+00
16	1.2000E+07	3.8721E-01	2.7331E+00
17	1.3000E+07	3.1456E-01	2.6434E+00
18	1.4000E+07	2.5443E-01	2.1646E+00
19	1.6000E+07	1.6493E-01	2.6621E+00
20	1.8000E+07	1.0177E-01	3.5831E+00

21 2.0000E+07 7.7585E-02 3.5831E+00

CORRELATION MATRIX

1	0																					
2	0	100																				
3	0	0	100																			
4	0	0	63	100																		
5	0	0	49	62	100																	
6	0	0	38	48	61	100																
7	0	0	29	37	46	59	100															
8	0	0	21	27	34	43	56	100														
9	0	0	14	18	23	29	38	49	100													
10	0	0	10	13	17	21	27	36	50	100												
11	0	0	7	9	11	14	19	25	34	53	100											
12	0	0	6	8	10	12	16	21	29	47	65	100										
13	0	0	5	6	8	10	13	18	24	39	57	64	100									
14	0	0	4	5	7	9	11	15	20	33	47	56	63	100								
15	0	0	4	5	6	7	10	13	18	29	40	47	53	60	100							
16	0	0	3	4	5	6	8	11	15	24	33	39	43	50	57	100						
17	0	0	3	3	4	5	7	9	13	21	28	33	36	38	48	47	100					
18	0	0	2	3	4	5	6	8	12	19	26	30	32	34	42	46	63	100				
19	0	0	2	2	3	4	5	6	9	15	20	23	25	26	33	35	51	61	100			
20	0	0	2	2	2	3	4	5	7	12	17	20	21	22	28	30	43	53	71	100		
21	0	0	2	2	2	3	4	5	7	12	17	20	21	22	28	30	43	53	71	100	100	

FINAL THINNED CORRELATION MATRIX OF RECTANGULAR BLOCK 2 1

SET 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
SET 2																							
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	2	4	7	10	13	17	21	26	40	35	21	15	10	7	7	6	5	6	6	5	5
3	0	1	2	4	7	10	14	18	22	28	36	57	34	22	16	11	11	10	9	10	10	8	8
4	0	1	2	3	5	8	10	13	16	21	26	40	37	27	19	13	14	12	11	12	12	10	10
5	0	1	1	2	4	6	8	11	14	17	22	30	40	31	22	15	16	15	13	14	14	12	12
6	0	0	1	2	3	4	6	8	10	12	16	22	28	25	21	17	18	17	14	16	16	14	14
7	0	0	1	1	2	3	4	5	7	8	11	15	19	19	19	17	20	20	17	18	18	16	16
8	0	0	1	1	2	3	4	5	6	7	9	13	16	18	20	21	29	30	25	28	28	25	25
9	0	0	1	1	2	3	4	6	7	8	11	14	18	22	26	32	48	51	42	48	49	43	43
10	0	0	1	1	2	3	4	5	6	8	10	13	16	20	24	30	47	54	43	37	37	35	35
11	0	0	1	1	2	3	4	5	6	8	10	13	17	21	25	32	49	53	54	53	45	41	41
12	0	0	1	1	2	3	4	4	5	7	8	11	14	17	21	27	40	49	33	52	52	36	36
13	0	0	0	1	1	2	3	3	4	5	7	9	11	14	16	21	31	37	35	32	49	41	41
14	0	0	0	1	1	2	3	4	4	5	7	9	11	14	17	22	32	40	40	45	41	71	71
15	0	0	0	0	1	1	1	2	2	2	3	4	5	6	8	10	15	17	16	16	18	23	23
16	0	0	0	0	1	1	1	2	2	2	3	4	5	6	8	10	15	17	16	18	18	23	23

FINAL THINNED CORRELATION MATRIX OF RECTANGULAR BLOCK 3 1

SET 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
SET 3																							
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	2	3	4	6	6	5	4	3	2	2	1	1	1	0	0	0	0	0	0	0
4	0	0	1	2	3	5	7	8	7	5	4	3	2	1	1	1	1	0	0	0	0	0	0
5	0	1	1	3	4	6	9	10	8	7	5	4	2	2	1	1	1	1	1	1	1	0	0
6	0	1	2	3	5	8	12	13	10	8	6	5	3	2	2	1	1	1	1	1	1	1	1
7	0	1	2	4	7	10	15	17	14	11	8	6	4	3	2	1	1	1	1	1	1	1	1
8	0	1	3	5	9	14	20	22	18	14	11	8	5	4	3	2	1	1	1	1	1	1	1
9	0	2	4	8	13	19	28	31	25	20	15	11	7	5	4	2	2	2	2	2	2	1	1

10	0	3	6	13	22	33	50	49	39	32	24	17	12	9	6	4	3	3	2	3	3	2	2
11	0	4	8	17	29	42	58	84	60	46	35	26	18	13	9	6	5	4	4	4	4	3	3
12	0	3	6	12	21	31	41	55	82	57	42	31	21	16	11	7	6	5	4	5	5	4	4
13	0	2	5	10	17	25	33	43	54	81	51	37	26	19	14	9	7	6	5	6	6	5	5
14	0	2	4	7	13	18	24	33	41	51	83	50	34	26	18	12	10	8	7	8	8	7	7
15	0	1	2	5	8	12	15	21	27	34	44	76	44	30	22	15	12	10	9	9	9	8	8
16	0	1	1	3	5	7	8	13	17	21	28	37	61	48	33	20	17	13	12	13	13	11	11
17	0	0	0	1	1	1	1	4	5	8	11	15	17	30	38	32	24	19	17	18	18	16	16
18	0	0	0	0	0	0	0	2	3	5	7	10	13	17	21	25	25	21	18	20	20	17	17
19	0	0	0	0	0	0	0	2	3	4	6	9	10	15	18	18	19	16	14	15	15	13	13
20	0	0	0	0	0	0	0	2	2	4	5	7	9	12	15	16	16	13	11	12	12	11	11
21	0	0	0	0	0	0	0	2	2	4	5	7	9	12	15	16	16	13	11	12	12	11	11

FINAL THINNED CORRELATION MATRIX OF RECTANGULAR BLOCK 3 2

SET 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SET 3																
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	2	2	2	1	1	1	1	1	0	0	0	0	0	0	0
4	0	2	3	2	2	1	1	1	1	0	1	0	0	0	0	0
5	0	3	3	3	2	2	1	1	1	1	1	1	0	0	0	0
6	0	4	4	3	3	2	2	1	1	1	1	1	1	1	0	0
7	0	5	5	4	4	3	2	1	1	1	1	1	1	1	0	0
8	0	7	7	6	5	4	3	2	2	1	1	1	1	1	0	0
9	0	9	10	8	7	5	4	3	2	2	2	2	1	1	1	1
10	0	15	16	12	11	8	6	4	4	3	3	3	2	2	1	1
11	0	21	23	18	16	12	8	6	6	4	5	4	3	3	1	1
12	0	25	27	21	18	14	10	7	7	5	6	5	4	4	2	2
13	0	29	31	24	21	16	11	9	8	6	7	6	4	4	2	2
14	0	44	38	29	26	19	13	11	10	8	9	7	6	6	3	3
15	0	37	61	43	32	24	17	13	12	10	11	9	7	7	3	3
16	0	22	30	42	49	35	22	17	16	13	14	12	9	9	4	4
17	0	14	18	18	19	23	23	20	22	18	19	16	13	13	6	6
18	0	10	11	12	13	14	15	19	27	21	22	19	15	14	7	7
19	0	8	9	9	10	11	12	14	21	16	17	14	11	11	5	5
20	0	7	8	8	9	10	10	12	17	13	14	12	9	9	5	5
21	0	7	8	8	9	10	10	12	17	13	14	12	9	9	5	5

THIS IS FOR18.DAT. THESE DATA WERE OUTPUT FROM OUTPUT.F4 DURING THE SECOND RUN.

```
 2.6056E+04 5.5454E+01          0          0          0          0 561 3103
 0.0000E-01-2.9130E+06          0          0          3         25 561 3103
          2          2          8          4         25          2 561 3103
2.96550E+06 0.0000E-013.00000E+06 7.0635E-094.00000E+06 6.0545E-06 561 3103
4.25000E+06 3.2290E-054.50000E+06 1.4229E-045.00000E+06 1.1319E-03 561 3103
5.50000E+06 4.8990E-036.00000E+06 1.3148E-026.40000E+06 2.0951E-02 561 3103
7.00000E+06 2.913[illegible]-028.00000E+06 4.3523E-029.00000E+06 5.6405E-02 561 3103
1.00000E+07 7.0824E-021.10000E+07 8.8335E-021.20000E+07 1.0466E-01 561 3103
1.24000E+07 1.1012E-011.28000E+07 1.1345E-011.34000E+07 1.1545E-01 561 3103
1.42000E+07 1.1253E-011.50000E+07 1.0319E-011.60000E+07 8.5661E-02 561 3103
1.70000E+07 7.1159E-021.80000E+07 6.0397E-021.90000E+07 5.2354E-02 561 3103
2.00000E+07 4.9802E-020.00000E-01 0.0000E-010.00000E-01 0.0000E-01 561 3103
 0.0000E-01 0.0000E-01          0          0          0          0 561 3  0
 0.0000E-01 0.0000E-01          0          0          0          0 561 0  0
 2.6056E+04 5.5454E+01          0          0          0          3 56133103
 0.0000E-01 0.0000E-01          0        103          0          1 56133103
 0.0000E-01 0.0000E-01          1          5        300         24 56133103
 1.0000E-05 2.9655E+06 3.0000E+06 4.5000E+06 5.0000E+06 5.5000E+06 56133103
 6.0000E+06 6.4000E+06 7.0000E+06 8.0000E+06 9.0000E+06 1.0000E+07 56133103
 1.1000E+07 1.2000E+07 1.2400E+07 1.2800E+07 1.3400E+07 1.4200E+07 56133103
 1.5000E+07 1.6000E+07 1.7000E+07 1.8000E+07 1.9000E+07 2.0000E+07 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
 0.0000E-01 0.0000E-01 0.0000E-01 9.8102E-03 2.1743E-03 1.1213E-03 56133103
 6.4926E-04 4.1475E-04 2.8004E-04 1.9625E-04 1.3556E-04 1.0627E-04 56133103
 8.0717E-05 4.5202E-05 3.1177E-05 2.2840E-05 1.4161E-05 7.6989E-06 56133103
 4.9700E-06 3.3078E-06 3.3677E-06 3.2374E-06 3.2901E-06 3.3410E-06 56133103
 7.8063E-03 2.0039E-03 1.1603E-03 7.4117E-04 5.0045E-04 3.5072E-04 56133103
 2.4225E-04 1.8990E-04 1.4426E-04 8.0777E-05 5.5716E-05 4.0814E-05 56133103
 2.5310E-05 1.3758E-05 8.8809E-06 5.9123E-06 6.0184E-06 5.7860E-06 56133103
 5.8800E-06 5.9706E-06 5.9363E-03 2.0374E-03 1.3015E-03 8.7875E-04 56133103
 6.1586E-04 4.2536E-04 3.3345E-04 2.5330E-04 1.4184E-04 9.7827E-05 56133103
 7.1675E-05 4.4442E-05 2.4158E-05 1.5594E-05 1.0382E-05 1.0567E-05 56133103
 1.0160E-05 1.0325E-05 1.0484E-05 4.2858E-03 1.8736E-03 1.2652E-03 56133103
 8.8660E-04 6.1237E-04 4.8005E-04 3.6467E-04 2.0422E-04 1.4085E-04 56133103
 1.0318E-04 6.3977E-05 3.4780E-05 2.2449E-05 1.4946E-05 1.5214E-05 56133103
 1.4626E-05 1.4864E-05 1.5093E-05 2.9245E-03 1.5352E-03 1.0759E-03 56133103
 7.4311E-04 5.8250E-04 4.4251E-04 2.4780E-04 1.7091E-04 1.2521E-04 56133103
 7.7633E-05 4.2205E-05 2.7241E-05 1.8135E-05 1.8460E-05 1.7748E-05 56133103
 1.8037E-05 1.8315E-05 2.3194E-03 1.2818E-03 8.9622E-04 6.9954E-04 56133103
 5.3193E-04 2.9805E-04 2.0605E-04 1.5110E-04 9.3828E-05 5.1199E-05 56133103
 3.3055E-05 2.1936E-05 2.2331E-05 2.1458E-05 2.1801E-05 2.2145E-05 56133103
 1.8305E-03 9.9687E-04 7.8879E-04 5.9656E-04 3.3368E-04 2.2931E-04 56133103
 1.6778E-04 1.0382E-04 5.6112E-05 3.6210E-05 2.4229E-05 2.4654E-05 56133103
 2.3720E-05 2.4120E-05 2.4479E-05 1.2584E-03 8.0235E-04 6.0860E-04 56133103
 3.4014E-04 2.3341E-04 1.7064E-04 1.0549E-04 5.6853E-05 3.6679E-05 56133103
 2.4604E-05 2.5029E-05 2.4091E-05 2.4502E-05 2.4862E-05 1.2132E-03 56133103
```

7.3665E-04	4.1210E-04	2.8228E-04	2.0632E-04	1.2742E-04	6.8515E-05	56133103
4.4195E-05	2.9709E-05	3.0213E-05	2.9093E-05	2.9595E-05	3.0022E-05	56133103
1.1778E-03	5.1042E-04	3.4960E-04	2.5525E-04	1.5734E-04	8.4370E-05	56133103
5.4463E-05	3.6703E-05	3.7322E-05	3.5950E-05	3.6582E-05	3.7099E-05	56133103
6.4399E-04	3.3772E-04	2.4566E-04	1.5187E-04	8.1568E-05	5.2653E-05	56133103
3.5553E-05	3.6138E-05	3.4813E-05	3.5427E-05	3.5927E-05	5.5514E-04	56133103
3.0607E-04	1.9082E-04	9.7294E-05	6.3064E-05	4.2403E-05	4.3161E-05	56133103
4.1529E-05	4.2239E-05	4.2855E-05	5.2055E-04	1.9629E-04	1.1893E-04	56133103
7.6334E-05	5.1005E-05	5.1858E-05	4.9925E-05	5.0764E-05	5.1518E-05	56133103
3.6067E-04	1.2743E-04	7.8377E-05	5.1302E-05	5.2831E-05	5.0684E-05	56133103
5.1458E-05	5.2288E-05	2.0041E-04	7.6818E-05	5.0685E-05	5.2118E-05	56133103
4.9873E-05	5.0546E-05	5.1383E-05	1.1831E-04	4.8976E-05	5.8831E-05	56133103
5.5796E-05	5.6556E-05	5.8679E-05	9.6505E-05	6.8270E-05	6.9727E-05	56133103
7.1432E-05	6.9898E-05	1.2271E-04	8.7846E-05	8.4973E-05	8.8075E-05	56133103
1.0377E-04	8.5493E-05	8.5557E-05	1.1119E-04	8.8579E-05	1.4332E-04	56133103
0.0000E-01	0.0000E-01	651	16	0	1	56133103
0.0000E-01	0.0000E-01	0	6	24	17	56133103
1.0000E-05	2.9655E+06	3.0000E+06	4.5000E+06	5.0000E+06	5.5000E+06	56133103
6.0000E+06	6.4000E+06	7.0000E+06	8.0000E+06	9.0000E+06	1.0000E+07	56133103
1.1000E+07	1.2000E+07	1.2400E+07	1.2800E+07	1.3400E+07	1.4200E+07	56133103
1.5000E+07	1.6000E+07	1.7000E+07	1.8000E+07	1.9000E+07	2.0000E+07	56133103
1.0000E-05	1.0063E+07	1.0500E+07	1.1000E+07	1.1500E+07	1.2000E+07	56133103
1.2500E+07	1.3000E+07	1.3500E+07	1.4500E+07	1.5500E+07	1.6500E+07	56133103
1.7500E+07	1.8500E+07	1.9200E+07	1.9600E+07	2.0000E+07	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	8.3058E-05	56133103
1.4844E-04	2.6064E-04	3.7524E-04	4.5534E-04	5.3857E-04	6.2883E-04	56133103
6.4806E-04	7.9263E-04	1.1891E-03	7.5482E-04	4.2969E-04	2.9180E-04	56133103
1.7085E-04	8.6572E-05	6.4545E-05	5.1961E-05	5.1274E-05	5.1039E-05	56133103
5.2859E-05	5.2712E-05	0.0000E-01	0.0000E-01	4.0818E-05	7.2946E-05	56133103
1.2809E-04	1.8442E-04	2.2377E-04	2.6441E-04	3.0930E-04	3.1916E-04	56133103
3.8994E-04	5.0186E-04	5.8301E-04	3.2199E-04	2.0680E-04	1.2214E-04	56133103
6.2970E-05	4.9059E-05	4.0496E-05	3.9722E-05	3.9723E-05	4.1230E-05	56133103
4.1030E-05	0.0000E-01	0.0000E-01	2.6017E-05	4.6494E-05	8.1641E-05	56133103
1.1754E-04	1.4262E-04	1.6797E-04	1.9809E-04	2.0483E-04	2.5075E-04	56133103
3.1643E-04	3.5774E-04	3.0733E-04	2.1894E-04	1.2953E-04	6.5086E-05	56133103
5.1786E-05	4.2596E-05	4.1723E-05	4.1619E-05	4.3160E-05	4.2995E-05	56133103
0.0000E-01	0.0000E-01	1.9004E-05	3.3962E-05	5.9635E-05	8.5860E-05	56133103
1.0418E-04	1.2232E-04	1.4534E-04	1.5056E-04	1.8465E-04	2.3323E-04	56133103
2.3973E-04	2.8874E-04	2.1935E-04	1.3239E-04	6.7160E-05	5.4068E-05	56133103
4.4575E-05	4.3591E-05	4.3448E-05	4.5051E-05	4.4894E-05	0.0000E-01	56133103
0.0000E-01	1.2684E-05	2.2667E-05	3.9802E-05	5.7302E-05	6.9531E-05	56133103
8.1309E-05	9.7555E-05	1.0131E-04	1.2448E-04	1.5744E-04	1.5973E-04	56133103
1.9249E-04	1.6561E-04	1.1695E-04	6.8463E-05	5.5962E-05	4.7260E-05	56133103
4.5920E-05	4.5857E-05	4.7625E-05	4.7407E-05	0.0000E-01	0.0000E-01	56133103
8.1071E-06	1.4488E-05	2.5440E-05	3.6626E-05	4.4443E-05	5.1914E-05	56133103
6.2458E-05	6.4906E-05	7.9783E-05	1.0087E-04	1.0147E-04	1.1881E-04	56133103
1.1772E-04	9.6556E-05	6.6029E-05	5.9009E-05	5.2243E-05	5.0177E-05	56133103

5.0294E-05	5.2379E-05	5.2024E-05	0.0000E-01	0.0000E-01	5.1655E-06	56133103
9.2315E-06	1.6209E-05	2.3337E-05	2.8318E-05	3.3406E-05	3.9248E-05	56133103
4.0533E-05	4.9588E-05	6.2325E-05	6.1794E-05	7.1907E-05	7.8160E-05	56133103
7.1155E-05	5.8265E-05	6.1021E-05	5.7148E-05	5.4191E-05	5.4847E-05	56133103
5.7384E-05	5.6755E-05	0.0000E-01	0.0000E-01	3.8236E-06	6.8332E-06	56133103
1.1999E-05	1.7274E-05	2.0961E-05	2.5030E-05	2.8544E-05	2.9244E-05	56133103
3.5555E-05	4.4339E-05	4.3155E-05	5.0665E-05	5.9635E-05	5.8893E-05	56133103
5.4737E-05	6.3239E-05	6.0352E-05	5.6925E-05	5.9622E-05	6.2942E-05	56133103
6.1592E-05	0.0000E-01	0.0000E-01	3.4827E-06	6.2243E-06	1.0929E-05	56133103
1.5735E-05	1.9092E-05	2.2949E-05	2.5748E-05	2.6262E-05	3.1816E-05	56133103
3.9494E-05	3.8398E-05	4.5459E-05	5.3744E-05	5.3447E-05	5.0812E-05	56133103
5.9990E-05	6.3176E-05	5.6100E-05	4.4810E-05	4.6175E-05	4.9798E-05	56133103
0.0000E-01	0.0000E-01	3.5722E-06	6.3839E-06	1.1210E-05	1.6138E-05	56133103
1.9582E-05	2.3522E-05	2.6433E-05	2.6971E-05	3.2689E-05	4.0592E-05	56133103
3.9453E-05	4.6678E-05	5.5234E-05	5.5084E-05	5.2359E-05	6.1931E-05	56133103
6.0867E-05	7.0559E-05	6.2721E-05	5.5971E-05	5.7615E-05	0.0000E-01	56133103
0.0000E-01	3.5029E-06	6.2601E-06	1.0992E-05	1.5825E-05	1.9204E-05	56133103
2.3067E-05	2.5922E-05	2.6448E-05	3.2057E-05	3.9807E-05	3.8687E-05	56133103
4.5755E-05	5.4210E-05	5.3968E-05	5.1325E-05	5.9520E-05	6.5095E-05	56133103
5.0005E-05	7.2168E-05	7.5061E-05	5.8291E-05	0.0000E-01	0.0000E-01	56133103
3.7625E-06	6.7237E-06	1.1807E-05	1.6998E-05	2.0626E-05	2.4773E-05	56133103
2.7845E-05	2.8413E-05	3.4438E-05	4.2770E-05	4.1555E-05	4.9146E-05	56133103
5.8267E-05	5.8132E-05	5.5354E-05	6.4054E-05	6.7981E-05	7.2160E-05	56133103
6.0170E-05	9.6254E-05	9.2310E-05	0.0000E-01	0.0000E-01	3.8953E-06	56133103
6.9615E-06	1.2223E-05	1.7598E-05	2.1354E-05	2.5694E-05	2.8753E-05	56133103
2.9304E-05	3.5482E-05	4.4009E-05	4.2727E-05	5.0654E-05	6.0231E-05	56133103
6.0354E-05	5.7961E-05	6.6403E-05	7.3704E-05	8.2936E-05	8.7185E-05	56133103
8.0735E-05	1.6098E-04	0.0000E-01	0.0000E-01	3.1310E-06	5.5955E-06	56133103
9.8255E-06	1.4146E-05	1.7164E-05	2.0591E-05	2.3214E-05	2.3708E-05	56133103
2.8753E-05	3.5740E-05	3.4736E-05	4.1019E-05	4.8566E-05	4.8371E-05	56133103
4.5853E-05	5.2800E-05	5.5340E-05	5.7705E-05	5.9686E-05	6.1825E-05	56133103
9.0587E-05	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	32	103	0	1	56133103
0.0000E-01	0.0000E-01	0	6	24	22	56133103
1.0000E-05	2.9655E+06	3.0000E+06	4.5000E+06	5.0000E+06	5.5000E+06	56133103
6.0000E+06	6.4000E+06	7.0000E+06	8.0000E+06	9.0000E+06	1.0000E+07	56133103
1.1000E+07	1.2000E+07	1.2400E+07	1.2800E+07	1.3400E+07	1.4200E+07	56133103
1.5000E+07	1.6000E+07	1.7000E+07	1.8000E+07	1.9000E+07	2.0000E+07	56133103
1.0000E-05	9.5700E+05	1.0000E+06	1.6000E+06	2.0000E+06	2.5000E+06	56133103
3.0000E+06	3.5000E+06	4.0000E+06	5.0000E+06	6.0000E+06	7.0000E+06	56133103
8.0000E+06	9.0000E+06	1.0000E+07	1.1000E+07	1.2000E+07	1.3000E+07	56133103
1.4000E+07	1.6000E+07	1.8000E+07	2.0000E+07	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	1.1970E-04	2.1392E-04	3.7564E-04	56133103

7.3665E-04	4.1210E-04	2.8228E-04	2.0632E-04	1.2742E-04	6.8515E-05	56133103
4.4195E-05	2.9709E-05	3.0213E-05	2.9093E-05	2.9595E-05	3.0022E-05	56133103
1.1778E-03	5.1042E-04	3.4960E-04	2.5525E-04	1.5734E-04	8.4370E-05	56133103
5.4463E-05	3.6703E-05	3.7322E-05	3.5950E-05	3.6582E-05	3.7099E-05	56133103
6.4399E-04	3.3772E-04	2.4566E-04	1.5187E-04	8.1568E-05	5.2653E-05	56133103
3.5553E-05	3.6138E-05	3.4813E-05	3.5427E-05	3.5927E-05	5.5514E-04	56133103
3.0607E-04	1.9082E-04	9.7294E-05	6.3064E-05	4.2403E-05	4.3161E-05	56133103
4.1529E-05	4.2239E-05	4.2855E-05	5.2055E-04	1.9629E-04	1.1893E-04	56133103
7.6334E-05	5.1005E-05	5.1858E-05	4.9925E-05	5.0764E-05	5.1518E-05	56133103
3.6067E-04	1.2743E-04	7.8377E-05	5.1302E-05	5.2831E-05	5.0684E-05	56133103
5.1458E-05	5.2288E-05	2.0041E-04	7.6818E-05	5.0685E-05	5.2118E-05	56133103
4.9873E-05	5.0546E-05	5.1383E-05	1.1831E-04	4.8976E-05	5.8831E-05	56133103
5.5796E-05	5.6556E-05	5.8679E-05	9.6505E-05	6.8270E-05	6.9727E-05	56133103
7.1432E-05	6.9898E-05	1.2271E-04	8.7846E-05	8.4973E-05	8.8075E-05	56133103
1.0377E-04	8.5493E-05	8.5557E-05	1.1119E-04	8.8579E-05	1.4332E-04	56133103
0.0000E-01	0.0000E-01	651	16	0	1	56133103
0.0000E-01	0.0000E-01	0	6	24	17	56133103
1.0000E-05	2.9655E+06	3.0000E+06	4.5000E+06	5.0000E+06	5.5000E+06	56133103
6.0000E+06	6.4000E+06	7.0000E+06	8.0000E+06	9.0000E+06	1.0000E+07	56133103
1.1000E+07	1.2000E+07	1.2400E+07	1.2800E+07	1.3400E+07	1.4200E+07	56133103
1.5000E+07	1.6000E+07	1.7000E+07	1.8000E+07	1.9000E+07	2.0000E+07	56133103
1.0000E-05	1.0063E+07	1.0500E+07	1.1000E+07	1.1500E+07	1.2000E+07	56133103
1.2500E+07	1.3000E+07	1.3500E+07	1.4500E+07	1.5500E+07	1.6500E+07	56133103
1.7500E+07	1.8500E+07	1.9200E+07	1.9600E+07	2.0000E+07	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	8.3058E-05	56133103
1.4844E-04	2.6064E-04	3.7524E-04	4.5534E-04	5.3857E-04	6.2883E-04	56133103
6.4806E-04	7.9263E-04	1.1891E-03	7.5482E-04	4.2969E-04	2.9180E-04	56133103
1.7085E-04	8.6572E-05	6.4545E-05	5.1961E-05	5.1274E-05	5.1039E-05	56133103
5.2859E-05	5.2712E-05	0.0000E-01	0.0000E-01	4.0818E-05	7.2946E-05	56133103
1.2809E-04	1.8442E-04	2.2377E-04	2.6441E-04	3.0930E-04	3.1916E-04	56133103
3.8994E-04	5.0186E-04	5.8301E-04	3.2199E-04	2.0680E-04	1.2214E-04	56133103
6.2970E-05	4.9059E-05	4.0496E-05	3.9722E-05	3.9723E-05	4.1230E-05	56133103
4.1030E-05	0.0000E-01	0.0000E-01	2.6017E-05	4.6494E-05	8.1641E-05	56133103
1.1754E-04	1.4262E-04	1.6797E-04	1.9809E-04	2.0483E-04	2.5075E-04	56133103
3.1643E-04	3.5774E-04	3.0733E-04	2.1894E-04	1.2953E-04	6.5086E-05	56133103
5.1786E-05	4.2596E-05	4.1723E-05	4.1619E-05	4.3160E-05	4.2995E-05	56133103
0.0000E-01	0.0000E-01	1.9004E-05	3.3962E-05	5.9635E-05	8.5860E-05	56133103
1.0418E-04	1.2232E-04	1.4534E-04	1.5056E-04	1.8465E-04	2.3323E-04	56133103
2.3973E-04	2.8874E-04	2.1935E-04	1.3239E-04	6.7160E-05	5.4068E-05	56133103
4.4575E-05	4.3591E-05	4.3448E-05	4.5051E-05	4.4894E-05	0.0000E-01	56133103
0.0000E-01	1.2684E-05	2.2667E-05	3.9802E-05	5.7302E-05	6.9531E-05	56133103
8.1309E-05	9.7555E-05	1.0131E-04	1.2448E-04	1.5744E-04	1.5973E-04	56133103
1.9249E-04	1.6561E-04	1.1695E-04	5.8463E-05	5.5962E-05	4.7260E-05	56133103
4.5920E-05	4.5857E-05	4.7625E-05	4.7407E-05	0.0000E-01	0.0000E-01	56133103
8.1071E-06	1.4488E-05	2.5440E-05	3.6626E-05	4.4443E-05	5.1914E-05	56133103
6.2458E-05	6.4906E-05	7.9783E-05	1.0087E-04	1.0147E-04	1.1881E-04	56133103
1.1772E-04	9.6556E-05	6.6029E-05	5.9009E-05	5.2243E-05	5.0177E-05	56133103

5.0294E-05	5.2379E-05	5.2024E-05	0.0000E-01	0.0000E-01	5.1655E-06	56133103
9.2315E-06	1.6209E-05	2.3337E-05	2.8318E-05	3.3406E-05	3.9248E-05	56133103
4.0533E-05	4.9588E-05	6.2325E-05	6.1794E-05	7.1907E-05	7.8160E-05	56133103
7.1155E-05	5.8265E-05	6.1021E-05	5.7148E-05	5.4191E-05	5.4847E-05	56133103
5.7384E-05	5.6755E-05	0.0000E-01	0.0000E-01	3.8236E-06	6.8332E-06	56133103
1.1999E-05	1.7274E-05	2.0961E-05	2.5030E-05	2.8544E-05	2.9244E-05	56133103
3.5555E-05	4.4339E-05	4.3155E-05	5.0665E-05	5.9635E-05	5.8893E-05	56133103
5.4737E-05	6.3239E-05	6.0352E-05	5.6925E-05	5.9622E-05	6.2942E-05	56133103
6.1592E-05	0.0000E-01	0.0000E-01	3.4827E-06	6.2243E-06	1.0929E-05	56133103
1.5735E-05	1.9092E-05	2.2949E-05	2.5748E-05	2.6262E-05	3.1816E-05	56133103
3.9494E-05	3.8398E-05	4.5459E-05	5.3744E-05	5.3447E-05	5.0812E-05	56133103
5.9990E-05	6.3176E-05	5.6100E-05	4.4810E-05	4.6175E-05	4.9798E-05	56133103
0.0000E-01	0.0000E-01	3.5722E-06	6.3839E-06	1.1210E-05	1.6138E-05	56133103
1.9582E-05	2.3522E-05	2.6433E-05	2.6971E-05	3.2689E-05	4.0592E-05	56133103
3.9453E-05	4.6678E-05	5.5234E-05	5.5084E-05	5.2359E-05	6.1931E-05	56133103
6.0867E-05	7.0559E-05	6.2721E-05	5.5971E-05	5.7615E-05	0.0000E-01	56133103
0.0000E-01	3.5029E-06	6.2601E-06	1.0992E-05	1.5825E-05	1.9204E-05	56133103
2.3067E-05	2.5922E-05	2.6448E-05	3.2057E-05	3.9807E-05	3.8687E-05	56133103
4.5755E-05	5.4210E-05	5.3968E-05	5.1325E-05	5.9520E-05	6.5095E-05	56133103
5.0005E-05	7.2168E-05	7.5061E-05	5.8291E-05	0.0000E-01	0.0000E-01	56133103
3.7625E-06	6.7237E-06	1.1807E-05	1.6998E-05	2.0626E-05	2.4773E-05	56133103
2.7845E-05	2.8413E-05	3.4438E-05	4.2770E-05	4.1555E-05	4.9146E-05	56133103
5.8267E-05	5.8132E-05	5.5354E-05	6.4054E-05	6.7981E-05	7.2160E-05	56133103
6.0170E-05	9.6254E-05	9.2310E-05	0.0000E-01	0.0000E-01	3.8953E-06	56133103
6.9615E-06	1.2223E-05	1.7598E-05	2.1354E-05	2.5694E-05	2.8753E-05	56133103
2.9304E-05	3.5482E-05	4.4009E-05	4.2727E-05	5.0654E-05	6.0231E-05	56133103
6.0354E-05	5.7961E-05	6.6403E-05	7.3704E-05	8.2936E-05	8.7185E-05	56133103
8.0735E-05	1.6098E-04	0.0000E-01	0.0000E-01	3.1310E-06	5.5955E-06	56133103
9.8255E-06	1.4146E-05	1.7164E-05	2.0591E-05	2.3214E-05	2.3708E-05	56133103
2.8753E-05	3.5740E-05	3.4736E-05	4.1019E-05	4.8566E-05	4.8371E-05	56133103
4.5853E-05	5.2800E-05	5.5340E-05	5.7705E-05	5.9686E-05	6.1825E-05	56133103
9.0587E-05	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	32	103	0	1	56133103
0.0000E-01	0.0000E-01	0	6	24	22	56133103
1.0000E-05	2.9655E+06	3.0000E+06	4.5000E+06	5.0000E+06	5.5000E+06	56133103
6.0000E+06	6.4000E+06	7.0000E+06	8.0000E+06	9.0000E+06	1.0000E+07	56133103
1.1000E+07	1.2000E+07	1.2400E+07	1.2800E+07	1.3400E+07	1.4200E+07	56133103
1.5000E+07	1.6000E+07	1.7000E+07	1.8000E+07	1.9000E+07	2.0000E+07	56133103
1.0000E-05	9.5700E+05	1.0000E+06	1.6000E+06	2.0000E+06	2.5000E+06	56133103
3.0000E+06	3.5000E+06	4.0000E+06	5.0000E+06	6.0000E+06	7.0000E+06	56133103
8.0000E+06	9.0000E+06	1.0000E+07	1.1000E+07	1.2000E+07	1.3000E+07	56133103
1.4000E+07	1.6000E+07	1.8000E+07	2.0000E+07	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.00[illegible]E-01	0.0000E-01	0.0000E-01	0.0000E-01	56133103
0.0000E-01	0.0000E-01	0.0000E-01	1.1970E-04	2.1392E-04	3.7564E-04	56133103

5.4080E-04 6.5622E-04 8.7338E-04 8.4468E-04 5.6322E-04 4.4332E-04 56133103
3.2792E-04 1.7830E-04 1.1359E-04 8.0398E-05 4.7291E-05 2.1866E-05 56133103
1.3959E-05 1.0764E-05 1.0833E-05 1.0640E-05 1.0966E-05 1.0987E-05 56133103
0.0000E-01 0.0000E-01 9.9124E-05 1.7715E-04 3.1105E-04 4.4783E-04 56133103
5.4340E-04 7.2319E-04 6.9945E-04 4.6639E-04 3.6711E-04 2.7153E-04 56133103
1.4765E-04 9.4066E-05 6.6573E-05 3.9160E-05 1.8106E-05 1.1559E-05 56133103
8.9133E-06 8.9709E-06 8.8106E-06 9.0805E-06 9.0979E-06 0.0000E-01 56133103
0.0000E-01 6.1233E-05 1.0943E-04 1.9216E-04 2.7664E-04 3.3569E-04 56133103
4.4676E-04 4.3209E-04 2.8811E-04 2.2677E-04 1.6773E-04 9.1202E-05 56133103
5.8105E-05 4.1122E-05 2.4189E-05 1.1184E-05 7.1406E-06 5.5063E-06 56133103
5.5416E-06 5.4428E-06 5.6094E-06 5.6202E-06 0.0000E-01 0.0000E-01 56133103
7.6915E-05 1.3746E-04 2.4136E-04 3.4749E-04 4.2165E-04 5.6114E-04 56133103
5.4277E-04 3.6186E-04 2.8483E-04 2.1067E-04 1.1455E-04 7.2987E-05 56133103
5.1656E-05 3.0385E-05 1.4049E-05 8.9684E-06 6.9158E-06 6.9601E-06 56133103
6.8360E-06 7.0454E-06 7.0589E-06 0.0000E-01 0.0000E-01 9.6734E-05 56133103
1.7288E-04 3.0356E-04 4.3704E-04 5.3032E-04 7.0578E-04 6.8261E-04 56133103
4.5513E-04 3.5823E-04 2.6498E-04 1.4409E-04 9.1784E-05 6.4964E-05 56133103
3.8214E-05 1.7670E-05 1.1281E-05 8.6989E-06 8.7545E-06 8.5983E-06 56133103
8.8615E-06 8.8789E-06 0.0000E-01 0.0000E-01 1.2185E-04 2.1777E-04 56133103
3.8238E-04 5.5052E-04 6.6800E-04 8.8904E-04 8.5985E-04 5.7332E-04 56133103
4.5124E-04 3.3377E-04 1.8149E-04 1.1562E-04 8.1829E-05 4.8136E-05 56133103
2.2257E-05 1.4209E-05 1.0957E-05 1.1027E-05 1.0830E-05 1.1162E-05 56133103
1.1184E-05 0.0000E-01 0.0000E-01 1.5381E-04 2.7487E-04 4.8262E-04 56133103
6.9487E-04 8.4316E-04 1.1222E-03 1.0853E-03 7.2365E-04 5.6959E-04 56133103
4.2130E-04 2.2908E-04 1.4594E-04 1.0329E-04 6.0760E-05 2.8094E-05 56133103
1.7936E-05 1.3830E-05 1.3919E-05 1.3671E-05 1.4090E-05 1.4116E-05 56133103
0.0000E-01 0.0000E-01 2.1749E-04 3.8868E-04 6.8249E-04 9.8261E-04 56133103
1.1923E-03 1.6300E-03 1.4327E-03 9.4218E-04 7.4527E-04 5.5069E-04 56133103
2.9923E-04 1.9007E-04 1.3434E-04 7.8836E-05 3.6183E-05 2.3091E-05 56133103
1.7929E-05 1.8034E-05 1.7729E-05 1.8283E-05 1.8307E-05 0.0000E-01 56133103
0.0000E-01 1.9147E-04 3.4218E-04 6.0085E-04 8.6504E-04 1.0497E-03 56133103
1.2901E-03 1.6627E-03 9.7421E-04 7.4179E-04 5.5627E-04 3.0280E-04 56133103
1.9405E-04 1.3757E-04 8.1216E-05 3.8033E-05 2.4305E-05 1.8520E-05 56133103
1.8656E-05 1.8294E-05 1.8834E-05 1.8889E-05 0.0000E-01 0.0000E-01 56133103
1.2411E-04 2.2180E-04 3.8945E-04 5.6069E-04 6.8038E-04 8.0360E-04 56133103
9.4920E-04 1.1818E-03 8.0160E-04 5.8019E-04 3.1764E-04 2.0319E-04 56133103
1.4437E-04 8.5221E-05 3.9963E-05 2.5544E-05 1.9432E-05 1.9576E-05 56133103
1.9192E-05 1.9756E-05 1.9816E-05 0.0000E-01 0.0000E-01 1.0069E-04 56133103
1.7993E-04 3.1594E-04 4.5486E-04 5.5195E-04 6.5392E-04 7.6322E-04 56133103
7.8970E-04 1.1657E-03 7.2555E-04 3.9213E-04 2.5526E-04 1.8124E-04 56133103
1.0795E-04 5.1890E-05 3.3209E-05 2.4732E-05 2.4951E-05 2.4391E-05 56133103
2.5058E-05 2.5181E-05 0.0000E-01 0.0000E-01 6.9263E-05 1.2379E-04 56133103
2.1735E-04 3.1291E-04 3.7970E-04 4.4548E-04 5.3065E-04 5.4954E-04 56133103
6.7530E-04 1.0815E-03 4.7757E-04 3.0178E-04 2.2207E-04 1.3234E-04 56133103
6.4941E-05 4.1782E-05 3.0398E-05 3.0721E-05 2.9928E-05 3.0684E-05 56133103
3.0896E-05 0.0000E-01 0.0000E-01 3.4255E-05 6.1215E-05 1.0749E-04 56133103
1.5477E-04 1.8778E-04 2.1579E-04 2.6980E-04 2.8319E-04 3.5036E-04 56133103
4.4305E-04 5.7115E-04 3.0780E-04 2.0170E-04 1.2382E-04 6.2166E-05 56133103
3.9732E-05 2.8942E-05 2.9241E-05 2.8493E-05 2.9213E-05 2.9415E-05 56133103
0.0000E-01 0.0000E-01 1.8334E-05 3.2765E-05 5.7531E-05 8.2828E-05 56133103
1.0051E-04 1.1161E-04 1.5096E-04 1.6129E-04 2.0252E-04 2.5985E-04 56133103
2.5515E-04 3.9294E-04 2.9724E-04 1.6981E-04 7.5507E-05 5.0470E-05 56133103
3.6227E-05 3.6610E-05 3.5536E-05 3.6360E-05 3.6682E-05 0.0000E-01 56133103
0.0000E-01 3.4078E-06 6.0902E-06 1.0696E-05 1.5394E-05 1.8680E-05 56133103

```
1.2279E-05 4.2342E-05 5.1329E-05 6.9882E-05 9.5568E-05 1.0051E-04 56133103
1.0526E-04 1.7972E-04 1.9071E-04 1.2068E-04 6.8943E-05 4.8273E-05 56133103
4.8984E-05 4.7691E-05 4.8819E-05 4.9252E-05 0.0000E-01 0.0000E-01 56133103
6.8608E-07 1.2261E-06 2.1535E-06 3.0988E-06 3.7609E-06-3.4760E-06 56133103
1.8547E-05 2.5293E-05 3.6727E-05 5.2523E-05 5.7286E-05 6.5982E-05 56133103
8.5817E-05 8.7632E-05 7.5288E-05 5.8870E-05 4.4274E-05 4.3338E-05 56133103
4.3166E-05 4.4554E-05 4.4589E-05 0.0000E-01 0.0000E-01 1.0795E-06 56133103
1.9293E-06 3.3882E-06 4.8763E-06 5.9175E-06-3.3234E-07 2.0523E-05 56133103
2.6901E-05 3.8230E-05 5.3858E-05 5.8091E-05 6.5788E-05 8.8134E-05 56133103
8.8839E-05 6.8819E-05 5.4376E-05 4.1143E-05 4.0209E-05 4.0049E-05 56133103
4.1336E-05 4.1361E-05 0.0000E-01 0.0000E-01 1.3428E-06 2.3999E-06 56133103
4.2147E-06 6.0659E-06 7.3611E-06 4.3311E-07 2.4111E-05 3.1318E-05 56133103
4.4321E-05 6.2247E-05 6.6934E-05 7.5657E-05 1.0158E-04 1.0250E-04 56133103
7.9388E-05 6.1182E-05 4.6022E-05 4.5092E-05 4.4852E-05 4.6271E-05 56133103
4.6323E-05 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 0.0000E-01 56133103
0.0000E-01 0.0000E-01          0          0          0          0 56133  0
0.0000E-01 0.0000E-01          0          0          0          0 561 0  0
```