ORNL/TM-11028

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

USER'S MANUAL FOR FERD-PC

B. D. Rooney

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this
report, send in name with report and
the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A06, Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-11028

Engineering Physics and Mathematics Division

USER'S MANUAL FOR FERD-PC

B. D. Rooney

Date Published - April 1989

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
operated by
Martin Marietta Energy Systems, Inc.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0290632 7

TABLE OF CONTENTS

	Page
LIST OF FIGURES	v
ABSTRACT	vii
1. INTRODUCTION	1
1.1 Purpose and Scope	1
1.2 Background	1
1.3 Method of Solution	2
2. PROGRAM DESCRIPTION	5
2.1 Fortran Language	5
2.2 System Requirements	5
2.3 Measured Performance	6
2.4 Program Structure	8
2.5 Description of Subroutines	8
3. RUNNING FERD-PC	15
3.1 System Configuration	15
3.2 Using Two Floppy Drives	15
3.3 Using a Hard or Fixed Disk	17
3.4 Modifying the Control Parameters From FERD-PC	19
3.5 Switching to Another Response File	19
3.6 Entering Calibration and Binning Parameters	20
4. THE RESPONSE DATA FILE	23
4.1 Description	23
4.2 Block (1) - Control Parameters	23
4.3 Block (2) - Integral Output	30
4.4 Blocks (3) & (4) - Pulse-Height Bins	30
4.5 Block (5) - Energy Bins	31
4.6 Block (6) - Window Row Energies	31
4.7 Block (7) - Gaussian FWHM	31
4.8 Block (8) - Window Matrix	32
4.9 Block (9) - Response Matrix	32
5. PULSE-HEIGHT DATA INPUT	33
5.1 Declaring the Type of Input File	33
5.2 Channel Pulse-Height Data Input	33
5.3 Prebinned Pulse-Height Data Input	36
6. FERD-PC OUTPUT	39
6.1 Main Output File	39
6.2 Binned Data Output	39
6.3 Log Output File	41
6.4 Punch Output File	46
6.5 Synthetic Window Output	46
6.6 Unformatted Response File	49
6.7 Temporary Scratch File	49

REFERENCES 51

APPENDIX A. PLOTFERD 53
A.1 Description 53
A.2 System Requirements 53
A.3 Choosing the Type of Plot 55
A.4 Function Keys 57

APPENDIX B. FERD-PC FORTRAN CODE 63

LIST OF FIGURES

		Page
Figure 1.1.	Pulse-Height distribution of neutrons with energies between 7.1 and 7.5 MeV using a 5-cm x 5-cm NE-213 scintillator.	4
Figure 1.2.	Unfolded neutron spectrum of the pulse-height distribution illustrated in Figure 1.1.	4
Figure 2.1.	Hierarchy of subroutine calls for FERD-PC.	9
Figure 3.1.	Input/Output files used by FERD-PC.	16
Figure 3.2.	Screen output from FERD-PC during sample run using two disk drives.	18
Figure 4.1.	The response data file.	24
Figure 5.1.	Sample input file for channel pulse-height data.	34
Figure 5.2.	Sample input file for pre-binned pulse-height data.	37
Figure 6.1.	Main output file from sample run.	40
Figure 6.2.	Binned data output from sample run. This file may also be used for prebinned input.	42
Figure 6.3.	Log output from sample run (last section only).	45
Figure 6.4.	Punch file output from sample run.	47
Figure 6.5.	Synthetic window output from sample run.	48
Figure A.1.	Plot output from sample run.	54
Figure A.2.	Plot output from PLOTFERD using response data file.	58

ABSTRACT

FERD-PC is a modified version of the unfolding code FERD, rewritten in Microsoft FORTRAN and optimized for use on an IBM-PC or compatible computer. It can be used to correct observed pulse-height distributions for the nonideal response of a pulse-height spectrometer and has been successfully applied to the analysis of data from the Tower Shielding Facility at Oak Ridge National Laboratory (ORNL). Modifications to the original code have been incorporated to optimize program execution and run time, while still maintaining most input and output options supported by the original code. In addition, several changes have been made to include an interactive binning code, a flux integration code, and a plotting utility, which supports enhanced color graphics (EGA) and a HP 7475A or equivalent plotter. This manual describes these changes and enhancements, and covers essential information to set up and run FERD-PC. Sample problems are included to illustrate input and output procedures using various options of the code.

1. INTRODUCTION

1.1 PURPOSE AND SCOPE

The purpose of this user's manual is to provide the reader with a detailed explanation of how to set up and apply FERD-PC to various applications requiring the unfolding of pulse-height data to a given response and window matrix. It has been modified specifically for the IBM-PC or compatible computer from an IBM mainframe version of FERD.[1] While it still maintains nearly all fundamental capabilities of the original code, it now provides the user with the interactive flexibility of the micro-computer. New features and improvements include: 1) an interactive binning code, 2) a restructured response file for easy editing and program control, 3) integral output of user-specified energy intervals, and 4) a plotting utility, named PLOTFERD, which supports enhanced color graphics and an HP 7475A or equivalent plotter. It is assumed throughout this manual that the user has a basic understanding of personal computers and their operation.

The primary scope used in the development of FERD-PC has been to apply the FERD process to analyze neutron spectra from pulse-height measurements obtained with a NE-213 spectrometer at the Tower Shielding Facility at ORNL. If the reader is interested in the detailed mathematical analysis of the FERD unfolding process, the previous user's manual on FERD and FERDO[1] should be used as a supplement to this document. However, this document is complete as a user's guide for FERD-PC and provides sample data, along with the response and window matrix, to illustrate the unfolding of neutron spectra obtained with a 5- x 5-cm NE-213 scintillator.

1.2 BACKGROUND

The development of the FERD unfolding process was originally conceived by W. R. Burrus in 1960. With others, he developed the code to solve radiation spectrum unfolding problems in which the true

neutron or gamma spectrum is related to a measured pulse-height spectrum. The main result of his work was the computer code FERDO and its many variants which include FERDOR,[2] SLOP,[3] and COOLC.[4] From FERDO, the FERD code evolved which included a more sophisticated process in determining confidence intervals. In 1983 Rust, Ingersoll, and Burrus[1] fully verified the numerical basis of the FERDO and FERD computer codes and modernized the source code. They also generated a user's manual which included the mathematical method in detail and gave several applied examples for both codes.

The Tower Shielding Facility at ORNL has been applying the FERD method to data analysis since the original development of the code. Several variations have been used and tested using a variety of equipment and procedures. With the increasing integration of personal computers into data acquisition applications, the requirement for local and immediate data analysis has again prompted the modification of FERD, this time for the IBM-PC. Although FERD-PC has been modified in many ways, the mathematical method is still the same one used in earlier versions of FERD. If the reader desires more information on the philosophy and mathematical theory of FERD, he is referred to Chapters 2 through 8 of Reference 1.

1.3 METHOD OF SOLUTION

The FERD method is typically applied to unfolding problems in which the real energy spectrum, x(E), is related to a measured pulse-height spectrum, b_i, by an integral equation of the form

$$\int_{E_{lo}}^{E_{up}} A_i(E)\, x(E)dE = \bar{b}_i + \varepsilon_i, \quad i = 1,2, \ldots M \tag{1-1}$$

where E represents energy, i represents the pulse-height bin, $A_i(E)$ represents the response matrix, and ε_i represents the stochastic counting error.[1] $A_i(E)dE$ can be thought of as the probability that a neutron or gamma ray with energy in the range $(E \pm \frac{1}{2}dE)$ will produce an

event or count in bin number i. Since such an equation does not have a unique solution for x(E), FERD-PC cannot solve directly for the true spectrum, x(E). Instead FERD-PC solves for a discrete approximation, Φ_k, to the pulse-height spectrum using the equation

$$\Phi_k = \int_{E_{lo}}^{E_{up}} W_k(E)\, x(E) dE, \qquad k = 1,2, \ldots, P \tag{1-2}$$

where $W_k(E)$ represents an ideal response and k is the energy bin number. In neutron and gamma spectral measurements, the window matrix can be thought of as a resolution function of an ideal spectrometer which has a single Gaussian response.[1] The effective resolution of the unfolded solution may then be varied by the choice of $W_k(E)$. In the above equations, M represents the total number of pulse-height bins while P represents the total number of energy bins. Confidence intervals, Φ_k^{lo} and Φ_k^{up}, are determined which include statistical uncertainties and the difficulty of fitting the unfolded solution.

Figure 1.1 shows an observed pulse-height distribution of neutrons with energies between 7.1 and 7.5 MeV. This spectrum was obtained using the Oak Ridge Electron Linear Accelerator (ORELA) and incorporating time of flight discrimination techniques. Figure 1.2 illustrates the corresponding unfolded spectrum using FERD-PC. While better resolution may be obtained by choosing window resolutions that are more narrow, the confidence interval degrades and low counting statistics may produce severe oscillations.

The mathematical method of FERD-PC is essentially the same as the original FERD code. The only exception is FERD-PC assumes no uncertainty in the response matrix, while FERD allows additional uncertainty by means of an upper and lower response matrix. Except for this modification, the unfolding processes for FERD-PC and FERD are identical.

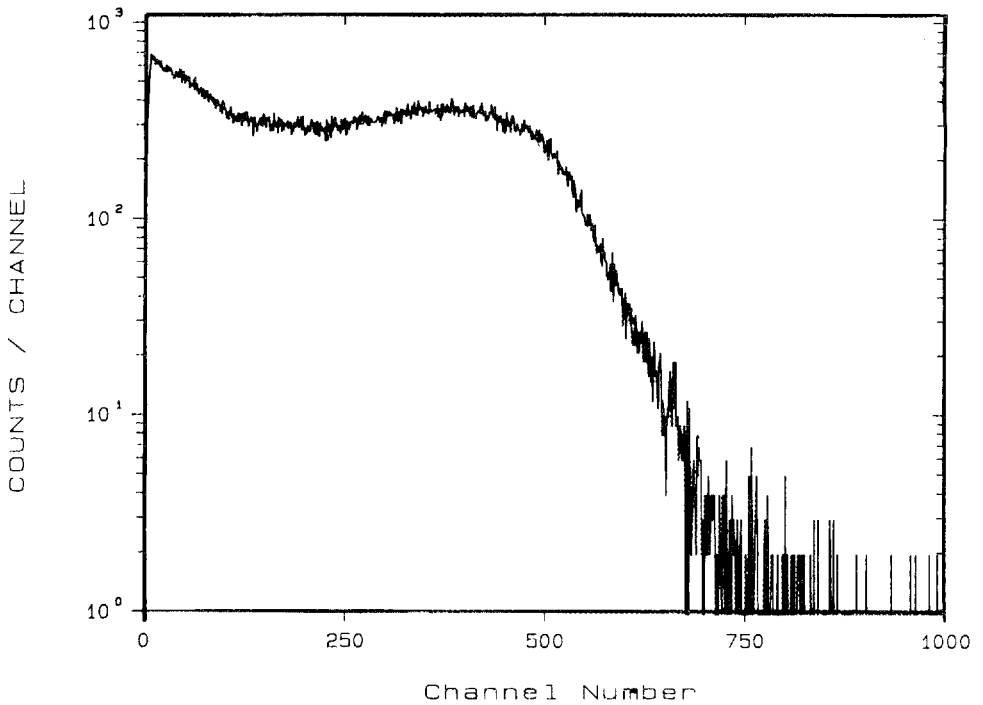

Figure 1.1. Pulse-height distribution of neutrons with energies between 7.1 and 7.5 MeV using a 5-cm by 5-cm NE-213 scintillator.

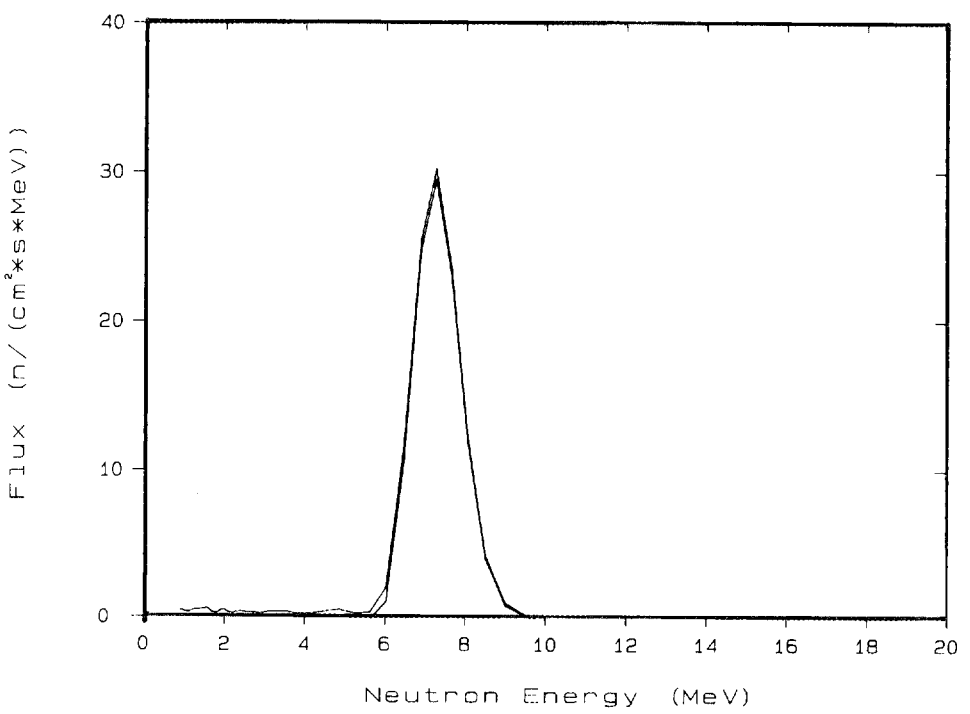

Figure 1.2. Unfolded neutron spectrum of the pulse-height distribution illustrated in Figure 1.1.

2. PROGRAM DESCRIPTION

2.1 FORTRAN LANGUAGE

The FERD-PC computer code used in this report was written and complied using Microsoft's FORTRAN Optimizing Compiler version 4.01 for a DOS operating system. The ANSI 77 FORTRAN standard was adhered to in most instances with the exception of a few non-standard extensions which better utilize the full potential of the personal computer. Changes to the source code will require the user to have access to a suitable compiler; however, every effort was made in restructuring the code to allow nearly all options and parameters to be changed interactively or by editing the response file. In addition, input format can be specified at the beginning of each data file, thus eliminating the need to change the source code for different input formats. This feature is illustrated in the sample input in Chapter 5.

2.2 SYSTEM REQUIREMENTS

The following system requirements pertain specifically to the compiled version of FERD-PC used in this report, which utilizes a 68 by 55 response matrix to perform the unfolding process. Response matrices larger than this may require additional memory and disk space. To accommodate FERD-PC, the computer system must have at a minimum the following components:

* IBM-PC or compatible that runs DOS version 2.1 or later.
* Two 360k disk drives or one high capacity or fixed drive.
* A minimum of 360k (kilobytes) of available user memory.
* An 8087/80287/80387 math coprocessor installed.

A hard or fixed disk is highly recommended. Additional requirements for graphics output using PLOTFERD include a color

graphics adapter or an enhanced graphics adapter with a compatible display. Alternate math packages and emulators are available with most compilers which would eliminate the need for the math coprocessor; however, the improvement in speed the math coprocessor provides is well worth the investment as is seen in the next section.

2.3 MEASURED PERFORMANCE

The precision of FERD-PC running on a personal computer has been shown to be accurate to 4 significant figures when compared to the original version of FERD running on an IBM 3033 main frame. Differences in the two results are due to round off errors during the binning process and at the beginning of the unfolding process which are then propagated and exaggerated as the unfolding process continues. This difference is predominant in regions where statistics are generally poor and small fluctuations have a greater contribution to certain regions of the unfolded spectrum.

Running times of FERD-PC are very dependent upon the size of the response matrix. In this report, the code has been applied only to unfolding neutron spectra with a response consisting of a 68 by 55 matrix. Configuring the system on a hard (fixed) disk, using an unformatted response matrix, and minimizing the generated output files will maximize performance. One can save about 30 seconds by using an unformatted response file compared to reading the response data from a ASCII file on floppy disk. To illustrate run times associated with the unfolding of neutron spectra, several runs were conducted on various computers using the same data and compared against the original version of FERD, which was run on an IBM 3033 mainframe computer. These results are given in Table 2.1. Included are comparable run times when FERD-PC is complied with a alternate math and emulator package which was provided with the FORTRAN compiler. The dramatic improvement in speed when using the math co-processor illustrates its value. Also included in Table 2.1 is a sample run where FERD-PC was complied on a DEC VAX II computer. This sample run actually used a slight variation of FERD-PC designed to unfold multiple spectra in one run.

Table 2.1. CPU run times comparing FERD and FERD-PC when unfolding the same pulse-height spectrum with a 68 x 55 response matrix

CODE	COMPUTER SYSTEM	CO-PROCESSOR	CPU TIME
FERD	IBM 3033	--	8 sec.
FERD-PC	DEC VAX II	--	8 sec.
FERD-PC	IBM PS/2 (20 MHz)	80387	20 sec.
FERD-PC	IBM PC AT (8 MHz)	80287	119 sec.
FERD-PC	IBM PC XT	8087	204 sec.
FERD-PC	IBM PC XT	None[a]	1081 sec.
FERD-PC	IBM PC XT	None[b]	3094 sec.

[a]This measurement was performed using an alternate math package provided by the compiler in place of the math co-processor.

[b]This measurement was performed using an emulator package provided by the compiler in place of the math co-processor.

Several changes were made to the original version of FERD in order to initially install it on a personal computer. Further changes to the code have caused about a factor of two increase in the speed of FERD-PC beyond the original PC version. Much of this increase in speed has been attributed to the removal of the option which allows an additional uncertainty calculation through the use of an upper and lower response matrix.

2.4 PROGRAM STRUCTURE

FERD-PC can be divided into three separate sections apart from the main subroutine. These are: 1) a pulse-height binning code, 2) the FERD unfolding code, and 3) a flux integration code. The hierarchy of subroutine calls for these sections are illustrated in Figure 2.1. The binning code is used to bin the raw pulse-height data into the same intervals as the response matrix. The unfolding code will calculate the differential flux for a given pulse-height spectrum. The integration routine will integrate the unfolded results over the intervals specified in the response data file.

In addition to these sections, FERD-PC requires an external file referred to as the response file. While this response file contains the response and window matrix, it also contains many other parameters which pertain to and control the binning, unfolding, and integration routines. All of the output options referred to in this report can be controlled by editing the response file. The default file name used for the response file is RESPONSE.DAT and is described in detail in Chapter 4.

2.5 DESCRIPTION OF SUBROUTINES

MAIN PROGRAM: The main program for FERD-PC is used to read in the control parameters from the response data file and provide the user with the option to display or edit these control parameters. It then prompts the user for applicable files and path names and opens these

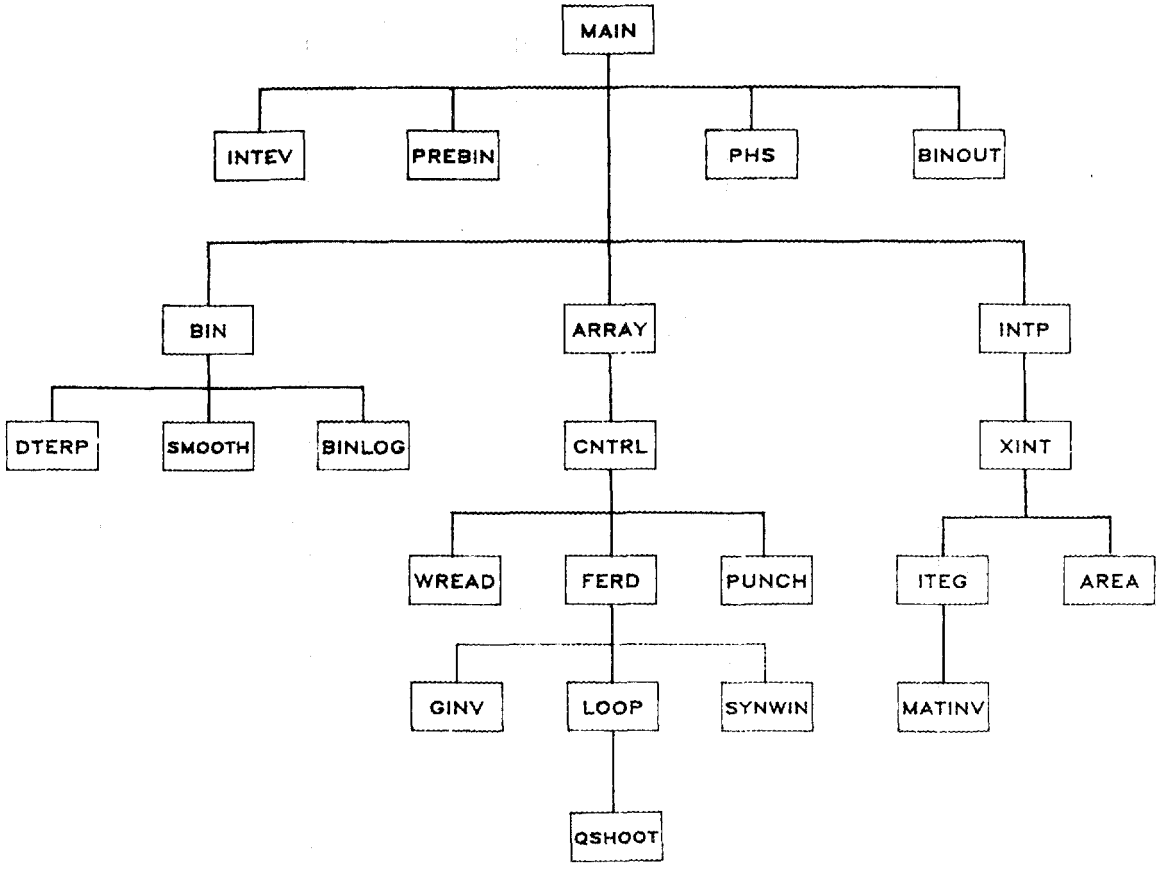

Figure 2.1. Hierarchy of subroutine calls for FERD-PC.

files in preparation for data input or output. If the prebinned option is not used, it will pass control to subroutine BIN where the raw pulse-height data is binned according to the parameters in blocks 3 and 4 of the response file. Once all input data have been read and binned, it passes control over to ARRAY which is the initial subroutine for the unfolding process. After the unfolding process, the main program calls subroutine INTP, where the unfolded result is integrated over each energy interval listed in block 2 of the response file. Upon return from the integration routine, FERD-PC displays the message 'UNFOLDING COMPLETE' and displays the run time for the unfolding and integration processes.

SUBROUTINE INTEV: This subroutine is used to read the energy intervals from the integral output table (block 2) in the response file. FERD-PC uses these intervals to integrate the unfolded results and will output these values as integral flux along with the unfolded results in the main output file. As many as fifty intervals may be read in by this subroutine in any order the user desires. The only requirement is that all intervals must be within the boundaries of the response matrix.

SUBROUTINE PREBIN: This subroutine reads in a previously binned data file. It is used only if the control parameter for reading in a prebinned data file is set to 1 in the response file, otherwise the program will assume a raw pulse-height spectrum as input and ignore this subroutine.

SUBROUTINE PHS: This subroutine reads in the pulse-height binning intervals (block 3) and binning width (block 4) used to bin the raw pulse-height data. If prebinned input is selected as one of the control parameters, this subroutine is not used. The pulse-height binning intervals are given in light units and correspond to the center of each bin.

SUBROUTINE BIN: This is the main controlling subroutine for binning the pulse-height data. The binning code is used to format the raw pulse-height data into bins as a function of light units. This is an interactive subroutine where the user is prompted for the file name (path included), calibration data, live time, and power normalization of the measured pulse-height data. The user has the option of subtracting out a background file, even if the background and foreground measurements are taken at different gain settings. It uses blocks 3 and 4 from the response file as the intervals to bin the pulse-height data. The result of this binning routine is the integral count rate for each light unit bin and corresponds to each row of the response matrix.

SUBROUTINE DTERP: This subroutine is a linear interpolation routine which determines the summed counts per channel at each interval boundary. Since the binning is performed from low to high energies, consecutively, it remembers the last upper boundary and uses this as the next lower boundary. This eliminates the need to loop through every lower channel each time it is called, however, care must be taken to reset this subroutine for each new spectrum. This is done by calling DTERP with the argument NTERMS set to zero.

SUBROUTINE SMOOTH: This subroutine is used to smooth the pulse-height data during the binning process. The user has the option to select the number of smoothing iterations performed on each spectrum during input of the calibration parameters. This subroutine is called once for every smoothing iteration selected.

SUBROUTINE BINOUT: This subroutine creates an output file listing the binned data. The user can use this output as a prebinned data file for additional FERD-PC runs without having to enter the calibration parameters each time.

SUBROUTINE BINLOG: This subroutine is nearly identical to subroutine BINOUT with the exception that it writes the binned data to

the log output file. This is used only if the option to create a log file of the unfolding process is selected.

SUBROUTINE ARRAY: This is the initial subroutine for the unfolding code. It allocates and assigns space for variables within array D based on the size of the response and window matrices. This eliminates the need to readjust array sizes for different size matrices. Array D is initially dimensioned in the main program to a size of 15000 words. For much larger arrays, the user may have to change this allocation, provided the compiler supports the additional size. If the size of this array is changed, the variable LTH in the main program must also be changed accordingly. The variable LTH is used by this subroutine to check the required memory against available memory.

SUBROUTINE CNTRL: This is the main controlling subroutine for the FERD unfolding process. It handles any additional data input such as the window and response matrices and sets up any arrays needed for subroutine FERD.

SUBROUTINE WREAD: This subroutine will calculate a Gaussian response given a FWHM resolution in percent. This is used to calculate the window matrix if using block 7 in the response file to provide all window data to FERD-PC. This subroutine is not used if the window matrix is read from the unformatted file or block 8 in the formatted response file.

SUBROUTINE FERD: This is the work horse of the unfolding code. The mathematical arguments of this subroutine remain identical to the original version and will not be discussed here in detail. It begins by calculating the initial upper bounds for the true spectrum and the error vector associated with the standard deviations of the observed counts. It then calls GINV to compute the "working inverse" used to calculate the coefficients of each window vector which are used in the biased estimation. After unscaling the working inverse, FERD enters

the window loop which calculates the upper and lower bounds for the current spectral estimate. This unfolded result, along with the error and synthetic window, is returned from the loop and passed on to subroutines which output the data to appropriate files.

SUBROUTINE LOOP: This subroutine is used to calculate the final unfolded estimate for each energy loop in subroutine FERD. It iteratively readjusts the initial vector of estimation until the proper bias inequality is satisfied. The result is a final biased estimate corresponding to flux per MeV when unfolding pulse-height spectra. This subroutine also calculates and stores the synthetic window for each final estimate if that particular option is chosen.

FUNCTION QSHOOT: This function gives an estimate of the error due to quadratic undershoot of the synthetic window between comparison points. If QSHOOT exceeds 20% of the confidence interval width then the energy mesh points may need to be taken closer together.[1] This value is calculated for each energy and is printed out in the final table as QSHOOT in the LOG output file.

SUBROUTINE GINV: This subroutine uses a Gramm-Schmidt-Hilbert orthogonalization procedure to replace the biased inverse matrix with a transpose of a scaled working inverse. During this process it uses the space of the original response matrix A as working storage, thus upon return this matrix must be reloaded from the temporary scratch unit. It might be significant to note that this subroutine utilizes almost half of the total CPU time required by FERD-PC.

SUBROUTINE PUNCH: This subroutine outputs the unfolded results into a punch style file consisting of a title card, and three columns of data. This may be useful for other programs that require the final output from FERD-PC.

SUBROUTINE SYNWIN: This subroutine outputs the synthetic window data generated by FERD-PC. It is output in the form of a condensed

response file so PLOTFERD can easily use this file to plot this data as it would plot the original input window data. It also provides the means to overlap and compare the synthetic windows to the original windows.

SUBROUTINE INTP: This is the initial subroutine for integrating the unfolding results. This subroutine starts out by writing the unfolded results to the main output file and then calls subroutine XINT for each interval. After the integral flux is determined for each interval the data is written to the main output file.

SUBROUTINE XINT: This subroutine sets up the required parameters for each interval boundary on the unfolded spectrum then sums up the flux between each point. It uses subroutine ITEG to perform the actual integration at each boundary.

FUNCTION AREA: This function calculates the area between one or more multiple boundaries. This function is called by subroutine XINT and can also call subroutine ITEG.

SUBROUTINE ITEG: This subroutine performs the actual integration of each interval. It interpolates the boundaries using a multiple point fitting process and then determines the area under the calculated curve.

SUBROUTINE MATINV: This subroutine inverts the matrix ARRAY and places the inverse back into ARRAY. This is used as part of the integration process for subroutine ITEG.

3. RUNNING FERD-PC

3.1 SYSTEM CONFIGURATION

Under most DOS operating systems the number of files that may be accessed at the same time is limited unless specified in the system configuration file (CONFIG.SYS). Since FERD-PC may access as many as nine files concurrently, the user must specify at least this number during boot up. This is accomplished through the FILES command in the CONFIG.SYS file. If not already in the root directory, the CONFIG.SYS file must be placed there to include the following command line:

FILES = 10

This will specify to the operating system the maximum number the files that can be opened concurrently. Remember, once the CONFIG.SYS file is edited the computer must be re-booted to activate any changes. If the number of files opened by FERD-PC exceeds the allocated number of the operating system, FERD-PC may respond with a "cannot find file" message, even if the file exists. Figure 3.1 illustrates all possible input and output files used by FERD-PC. For more information on file allocation see the reference for your specific operating system.

3.2 USING TWO FLOPPY DRIVES

A hard or fixed disk is highly recommended for FERD-PC due to its size and the number of data files it may generate; however, it can be set upfor use with two floppy drives. To do this, the user must insert the disk containing FERD-PC.EXE and RESPONSE.DAT into drive A. If the option to use the unformatted response file is selected, it must also be included on the same drive. Next, the disk containing any input data should be inserted into drive B. There should be ample space on this disk for FERD-PC output. If there is not sufficient space, the program will terminate early after displaying a "no space left on

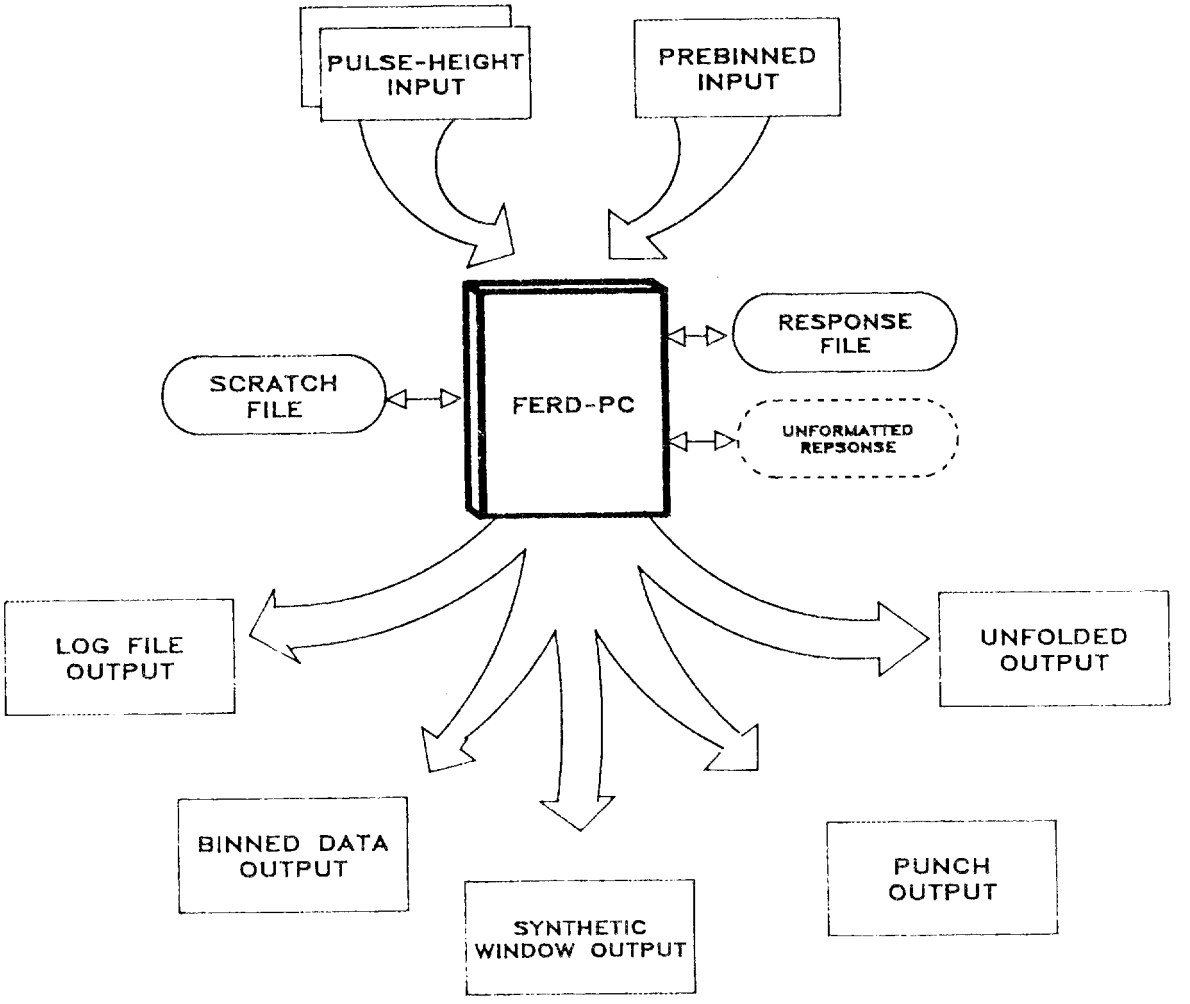

Figure 3.1. Input/Output files used by FERD-PC.

device" message. Set the default path to drive A. FERD-PC is now ready to run.

All input and output filenames entered while running FERD-PC using a two drive setup must include the drive B: prefix. If subdirectories are used, they must also be included when entering the file path name. All output files will then be written to the disk in drive B. If using a high capacity floppy drive, there may be sufficient space available for FERD-PC and all input and output files, but keep in mind that some files such as the LOG output may take as many as 90 kilobytes. Also one must consider that a temporary scratch file is written to the default drive, approximately 30 kilobytes, and then deleted at the end of the program. An example of running FERD-PC using two floppy disks is illustrated in Figure 3.2.

3.3 USING A HARD OR FIXED DISK

As stated above, a hard or fixed disk is highly recommended. FERD-PC may run as much as 30 percent faster using a hard disk as opposed to floppy drives. To install FERD-PC the user must copy the following programs onto the hard disk.

* FERD-PC.EXE
* RESPONSE.DAT
* RESPONSE.UNF (optional unformatted response file)
* PLOTFERD.EXE (utility program used for plotting)

The directory containing these programs must be the default directory prior to running FERD-PC. Input data may be in any drive or directory; however, the file name must include the path if different than the default path. If the input data file is in the same directory as FERD-PC.EXE a path does not need to be included when entering the file name. All output data will be written to the same directory as FERD-PC.EXE unless a path is also included when entering the output file name.

```
A>FERD-PC

PROGRAM FERD-PC

Display or Modify Control Parameters (Y/N) ? Y

    Parameter      Description

A)  0              : 0/1 = No effect / Prompt user for different response file
B)  0              : 0/1 = No effect / Create unformatted response file
C)  0              : 0/1 = No effect / Use unformatted response file
D)  1              : 0/1 = No effect / Create log file of unfolding process
E)  0              : 0/1 = No effect / Write response matrix to log file
F)  0              : 0/1 = No effect / Normalize data to a power level (kW)
G)  0              : 0/1 = No effect / Use prebinned data file for input
H)  1              : 0/1 = No effect / Output binned data to file
I)  1              : 0/1 = No effect / Output results to punch style file
J)  1              : 0/1 = No effect / Output synthetic window resolutions
K)  1              : 0/1 = Use windows from blk (8) / FWHM from blk (7)
O)  1.000          : Parameter TAU used for weighting the biased estimation
P)  1.000          : Detector normalization factor (multiplier)
Q)  RESPONSE.UNF   : Default file name for unformatted response file
R)  1.000          : Normalization parameter for block 7 (FWHM)

Enter corresponding letter to change parameter
Then hit Enter (blank = no more changes) ... ?

Enter a title for this run ................. ? SAMPLE RUN - PUBE SOURCE
Enter file name for log output ............. ? B:SAMPLE.LOG
Enter file name for binned data output ..... ? B:SAMPLE.BIN
Enter file name for punch output ........... ? B:SAMPLE.PUN
Enter file name for synthetic window output. ? B:SAMPLE.WIN
Enter file name for unfolded output ........ ? B:SAMPLE.OUT

Enter file name of foreground data ............... ? B:SAMPLE.FGD
Light units at calibration point ................. ? .895
Calibration channel .............................. ? 131.6
Zero Intercept (Channel) ......................... ? 13.0
Live Time (sec) .................................. ? 8000
Number of smoothing iterations (0 to 8) .......... ? 0

Enter file name of background data (blank = none). ? B:SAMPLE.BGD
Calibration channel .............................. ? 132.3
Zero Intercept (Channel) ......................... ? 13.0
Live Time (sec) .................................. ? 8000
Number of smoothing iterations (0 to 8) .......... ? 1

UNFOLDING COMPLETE!!!

RUN TIME = 194 seconds

A>
```

Figure 3.2. Screen output from FERD-PC during sample run using two disk drives.

3.4 MODIFYING THE CONTROL PARAMETERS FROM FERD-PC

Every time FERD-PC is run, the user will be asked if he wishes to display or modify the control parameters in the response file. Entering (N)o to this question will accept all control parameters as is and the program will continue without displaying these parameters. One may also enter a negative reply here by hitting only the carriage return. Entering a (Y)es here will display block 1 from the response file along with a letter corresponding to each control parameter as was shown in Figure 3.2.

The user will then be asked to enter the letter of the corresponding control parameter he wishes to change. The user can change any parameter here by entering the appropriate letter. The parameter will be automatically changed and displayed or the user will be prompted to enter the new parameter. A blank carriage return without any letter will accept all control parameters as displayed and the program will continue. If any of the control parameters are changed in this manner the program will eventually ask if the user wants to make these changes permanent before continuing. A reply of (Y)es will edit the response file and permanently copy over all changes to it. A reply of (N)o will not effect the response file but will use the displayed control parameters, including any changes, for the current run.

It should also be noted that not all control parameters in block 1 of the response file are displayed using the above procedure. The number of response bins and window vectors corresponding to the response function and window matrix can not be changed without changing the entire response file, thus it is of no practical use to edit these parameters while running FERD-PC. Further information on each of these control parameters are described in Chapter 4.

3.5 SWITCHING TO ANOTHER RESPONSE FILE

If FERD-PC is unable to find a response data file called RESPONSE.DAT when it is started, it will prompt the user for a

different file and path name. However, if the user wants to use several different response files in the same directory, he can switch from RESPONSE.DAT to another response file by modifying the first control parameter in block 1. If this control parameter is set to 1, FERD-PC will prompt the user for the name of the response file at the start of each run. This new response file will then be used just as RESPONSE.DAT would have been used. Each response file must use the same format as described in Chapter 4.

The unformatted response file is used to read in only the response and window matrices and does not include any binning or control parameters. An ASCII response file which includes these parameters in block 1 must be accessed by FERD-PC before the program will be allowed to continue. If the unformatted response file is used to replace the ASCII formatted response file a format error will occur.

3.6. ENTERING CALIBRATION AND BINNING PARAMETERS

If the control parameter for the prebinned input data is not selected, the program will assume that the input file consists of raw pulse-height data and will prompt the user for a foreground and background data file. If the files exist, FERD-PC will prompt the user for calibration parameters to be used in binning the data, as was illustrated in Figure 3.2. The following is a description of each question used in the binning process.

A) Light Units at Calibration Point: This is the calibration energy in light units corresponding to the calibration channel. For example, the half-height point at the upper shoulder (compton edge) of the 1.28-MeV ^{22}Na gamma calibration spectrum is approximately 0.895 light units for the response matrix listed in this report.

B) Calibration Channel: This is the pulse-height channel corresponding to the calibration point of the system. This

corresponds to the number light units entered in the previous question.

C) Zero Intercept: This is the zero pulse-height channel corresponding to zero light units of the system. This channel number is obtained by extrapolating several energy points on the pulse-height spectrum using different gamma-ray sources or a pulse-height generator.

D) Live Time: This is the live time of the of the current input spectrum entered in seconds.

E) Reactor Power: This is a normalization factor used only if the appropriate control parameter in block 1 of the response file is set. This factor is used to calculate the count rate on a per-kilowatt basis, and is entered in kilowatts.

F) Number of Smoothing Iterations: This is the number of smoothing iterations performed on the raw pulse-height counts prior to any other calculations.

After the binning parameters for the foreground data are entered, the user has the option of subtracting a background run from the data. If no background run is included, the user should enter only a carriage return when prompted for the background file name. If a background file is included, the user enters the file name and calibration data as he did for the foreground data. It should be noted that the calibration channels of the foreground and background data do not have to be the same. The binning process bins each file separately before subtracting the background.

4. THE RESPONSE DATA FILE

4.1 DESCRIPTION

The response file is an external file comprised of nine blocks of data containing numerous control parameters, binning data, and output information for FERD-PC. This is an ASCII file which also contains the response and window matrices required for the unfolding process. Upon running FERD-PC the program looks for an ASCII data file called RESPONSE.DAT. If this file is not found in the same directory the user will be asked to enter a new file and path name. FERD-PC will be unable to continue without this file since it contains necessary user input to execute the unfolding process. An example of the RESPONSE.DAT file used in this report is shown in Figure 4.1.

The user also has the option to create and use an unformatted response file which contains the response and window matrices. This enables FERD-PC to read in the unfolding parameters much faster than using the formatted ASCII response file. However, RESPONSE.DAT is still required to establish the control parameters needed by FERD-PC. The unformatted file may be created from the ASCII response file by selecting the appropriate control parameter in block 1. The following is a detailed description of each data block listed in the response file.

4.2 BLOCK (1) - CONTROL PARAMETERS

This section contains numerous parameters and flags used by FERD-PC to control several input and output options. Upon running FERD-PC, the program looks for the block 1 label within the first ten lines of the response file. This permits up to nine lines of comments to be inserted prior to this label. Each block label is represented by the block number enclosed in parenthesis. Block 1 is identified by locating "(1)" using format (A3). If the first three characters of the block 1 label are not located, the program will terminate execution with a message that block 1 was not found. FERD-PC uses this approach

NE-213 NEUTRON RESPONSE - FILE #110

(1) CONTROL PARAMETERS FOR FERD-PC:

A)	0	: 0/1 = No effect / Prompt user for different response file
B)	0	: 0/1 = No effect / Create unformated response file
C)	0	: 0/1 = No effect / Use unformated response file
D)	0	: 0/1 = No effect / Create log file of unfolding process
E)	0	: 0/1 = No effect / Write response matrix to log file
F)	0	: 0/1 = No effect / Normalize data to a power level (kW)
G)	0	: 0/1 = No effect / Use prebinned data file for input
H)	0	: 0/1 = No effect / Output binned data to file
I)	0	: 0/1 = No effect / Output results to punch style file
J)	0	: 0/1 = No effect / Output synthetic window resolutions
K)	1	: 0/1 = Use windows from blk (8) / FWHM from blk (7)
L)	68	: Number of pulse-height bins (rows in response matrix)
M)	55	: Number of energy bins (col's in response & window matrix)
N)	55	: Number of window vectors (rows in window matrix)
O)	1.000	: Parameter TAU used for weighting the biased estimation
P)	1.000	: Detector normalization factor (multiplier)
Q)	RESPONSE.UNF	: Default file name for unformatted response file
R)	1.000	: Normalization parameter for block 7 (FWHM)

(2) ENERGY INTERVALS USED IN INTEGRATED OUTPUT IN MeV (6X,2F8.0)

LOW	HIGH
1.000	1.200
1.200	1.600
1.600	2.000
2.000	3.000
3.000	4.000
4.000	6.000
6.000	8.000
8.000	10.00
10.00	12.00
1.000	10.00

(3) PULSE HEIGHT BINS FOR MATRIX IN LIGHT UNITS (5X,8F9.0)

1.075E-1	1.225E-1	1.375E-1	1.525E-1	1.675E-1	1.825E-1	2.050E-1	2.350E-1
2.650E-1	2.950E-1	3.250E-1	3.550E-1	3.850E-1	4.150E-1	4.550E-1	5.050E-1
5.550E-1	6.050E-1	6.550E-1	7.050E-1	7.550E-1	8.050E-1	8.550E-1	9.050E-1
9.750E-1	1.065E+0	1.155E+0	1.245E+0	1.335E+0	1.425E+0	1.515E+0	1.605E+0
1.695E+0	1.785E+0	1.915E+0	2.085E+0	2.255E+0	2.425E+0	2.595E+0	2.765E+0
2.935E+0	3.105E+0	3.275E+0	3.445E+0	3.685E+0	3.995E+0	4.305E+0	4.615E+0
4.925E+0	5.235E+0	5.545E+0	5.855E+0	6.165E+0	6.475E+0	6.910E+0	7.470E+0
8.030E+0	8.590E+0	9.150E+0	9.710E+0	1.027E+1	1.083E+1	1.139E+1	1.195E+1
1.251E+1	1.307E+1	1.363E+1	1.419E+1				

(4) PULSE HEIGHT BINNING WIDTH IN LIGHT UNITS (5X,8F9.0)

0.0150	0.0150	0.0150	0.0150	0.0150	0.0150	0.0300	0.0300
0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0500	0.0500
0.0500	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500
0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900
0.0900	0.0900	0.1700	0.1700	0.1700	0.1700	0.1700	0.1700
0.1700	0.1700	0.1700	0.1700	0.3100	0.3100	0.3100	0.3100
0.3100	0.3100	0.3100	0.3100	0.3100	0.3100	0.5600	0.5600
0.5600	0.5600	0.5600	0.5600	0.5600	0.5600	0.5600	0.5600
0.5600	0.5600	0.5600	0.5600				

Figure 4.1. The response data file.

(5) ENERGY BINS FOR RESPONSE AND WINDOW MATRIX IN MeV (5X,7F10.0)

```
2.347E+01 2.256E+01 2.157E+01 2.055E+01 1.949E+01 1.852E+01 1.753E+01
1.653E+01 1.562E+01 1.479E+01 1.402E+01 1.320E+01 1.239E+01 1.175E+01
1.124E+01 1.077E+01 1.025E+01 9.741E+00 9.261E+00 8.757E+00 8.236E+00
7.736E+00 7.235E+00 6.839E+00 6.555E+00 6.254E+00 5.938E+00 5.645E+00
5.345E+00 5.042E+00 4.752E+00 4.453E+00 4.152E+00 3.906E+00 3.707E+00
3.500E+00 3.296E+00 3.097E+00 2.898E+00 2.702E+00 2.498E+00 2.296E+00
2.099E+00 1.932E+00 1.812E+00 1.709E+00 1.609E+00 1.510E+00 1.412E+00
1.309E+00 1.204E+00 1.106E+00 1.006E+00 9.069E-01 8.113E-01
```

(6) ROW ENERGIES FOR WINDOW MATRIX IN MeV (5X,7F10.0)

```
2.347E+01 2.256E+01 2.157E+01 2.055E+01 1.949E+01 1.852E+01 1.753E+01
1.653E+01 1.562E+01 1.479E+01 1.402E+01 1.320E+01 1.239E+01 1.175E+01
1.124E+01 1.077E+01 1.025E+01 9.741E+00 9.261E+00 8.757E+00 8.236E+00
7.736E+00 7.235E+00 6.839E+00 6.555E+00 6.254E+00 5.938E+00 5.645E+00
5.345E+00 5.042E+00 4.752E+00 4.453E+00 4.152E+00 3.906E+00 3.707E+00
3.500E+00 3.296E+00 3.097E+00 2.898E+00 2.702E+00 2.498E+00 2.296E+00
2.099E+00 1.932E+00 1.812E+00 1.709E+00 1.609E+00 1.510E+00 1.412E+00
1.309E+00 1.204E+00 1.106E+00 1.006E+00 9.069E-01 8.113E-01
```

(7) GAUSSIAN FWHM (%) USED TO CALCULATE WINDOWS (5X,7F10.0)

```
16.6894   16.7241   16.7649   16.8137   16.8646   16.9161   16.9745
17.0448   17.1145   17.1821   17.2552   17.3410   17.4337   17.5198
17.5988   17.6720   17.7578   17.8544   17.9535   18.0697   18.2019
18.3459   18.5091   18.6537   18.7678   18.8972   19.0480   19.2003
19.3735   19.5666   19.7729   20.0116   20.2812   20.5316   20.7515
21.0104   21.2923   21.5911   21.9330   22.3169   22.7635   23.2764
23.8552   24.4264   24.8877   25.3320   25.8097   26.3376   26.9147
27.6032   28.3973   29.2468   30.2660   31.4432   32.7997
```

(8) WINDOW MATRIX (1X,2I3,1X,7F10.0)

```
 1  1  2.3984E-1 2.0650E-1 1.2491E-1 5.1376E-2 1.3701E-2 2.8638E-3 4.0817E-4
 1  8  3.9812E-5 3.4974E-6 2.9309E-7 0.0000E+0 0.0000E+0 0.0000E+0 0.0000E+0
 2  1  2.1190E-1 2.4899E-1 2.0572E-1 1.1336E-1 3.9715E-2 1.0365E-2 1.8032E-3
 2  8  2.0918E-4 2.0997E-5 1.9471E-6 1.6869E-7 0.0000E+0 0.0000E+0 0.0000E+0
                                   .
                                   .
                                   .
54 43  0.0000E+0 0.0000E+0 0.0000E+0 0.0000E+0 0.0000E+0 1.3541E-5 5.4937E-4
54 50  1.3291E-2 1.6244E-1 8.5267E-1 2.3570E+0 3.2944E+0 2.4124E+0 0.0000E+0
55 43  0.0000E+0 0.0000E+0 0.0000E+0 0.0000E+0 0.0000E+0 0.0000E+0 2.5814E-6
55 50  2.1661E-4 8.4232E-3 1.1776E-1 8.0021E-1 2.4683E+0 3.5303E+0 0.0000E+0
```

(9) RESPONSE MATRIX 68 x 55 (2I3,8F9.0)

```
 1  1 1.81E-02 2.04E-02 2.21E-02 2.26E-02 2.66E-02 3.47E-02 3.78E-02 4.31E-02
 1  9 4.09E-02 4.37E-02 4.73E-02 5.11E-02 6.00E-02 7.26E-02 8.61E-02 7.18E-02
 1 17 5.10E-02 5.59E-02 5.89E-02 4.89E-02 4.25E-02 3.84E-02 4.17E-02 4.46E-02
 1 25 5.00E-02 5.89E-02 6.09E-02 6.09E-02 6.01E-02 6.32E-02 6.92E-02 7.62E-02
                                   .
                                   .
                                   .
67  1 6.81E-03 1.51E-03 4.75E-04 2.38E-04 6.13E-05 0.00E+00 0.00E+00 0.00E+00
67  9 3.55E-05 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00
68  1 1.73E-03 9.59E-04 3.88E-04 1.78E-04 6.14E-05 0.00E+00 0.00E+00 0.00E+00
68  9 3.55E-05 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00
00  0

END
```

Figure 4.1. The response data file (continued).

in locating each block of data in the response file. A blank line must follow each block label.

This block can be easily edited directly or through FERD-PC to change any of the input and output options. Selecting (Y)es to the question in FERD-PC:

Display or Modify Control Parameters (Y/N) ?

will display block 1 and enable the user to make changes to these parameters for the current run or permanently write these changes to the response file. The following describes each line of this block in detail with each letter referencing the control parameter as listed when running FERD-PC. Each control parameter must start in column 6 and must not exceed 12 characters in length.

LINE A: 0 = No effect, continue using the current file.
1 = Prompt the user for a different response file at the beginning of each run. This option permits the user to employ different response files for the unfolding process. The user will be automatically prompted by FERD-PC for a new file and path name. The RESPONSE.DAT file will then be closed and the new response file used in place of it. The user may select the current response file by re-entering the file name. This control parameter will be ignored in the new response file.

LINE B: 0 = No effect.
1 = Write the response and window matrices, blocks 5 through 9, to an unformatted file using the file name listed in line Q. This option will create an unformatted file containing the response and window matrix listed in the current response file. This permits the following option in line C to be used in subsequent runs. If the filename already exists in the same directory it will be over written.

LINE C: 0 = Read in response and windows using the current file.
1 = Read the response and window matrices from an unformatted file with the file name given the line Q. Using an unformatted file to read in the response and window matrices may expedite the run time of FERD-PC as much as 30 percent. The unformatted file must exist in the same directory as FERD-PC and can be created using the option in line B. This does not eliminate the need for RESPONSE.DAT which still contains the control and output parameters for FERD-PC.

LINE D: 0 = No effect.
1 = This option will cause FERD-PC to create a log file during the unfolding process. This log file will contain selected control parameters, input data, binning results, and FERD-PC output. This option can be very useful to insure correct format of input files and to review FERD-PC output. The user will be prompted for an output file and path name if this option is selected.

LINE E: 0 = No effect.
1 = This option will output the window and response matrices to the log file in ASCII format. This enables the user to check the matrices FERD-PC is using. Selecting this option automatically selects the previous option in line D which opens the log file.

LINE F: 0 = Do not normalize input data to a power level.
1 = This option is needed when the radiation source is provided by a reactor. It normalizes the input data to a power level in kilowatts. This is used only for the binning process and will force FERD-PC to prompt the user for an average power in kilowatts. The binned count rate is divided by this power during the binning process.

LINE G: 0 = Input raw pulse-height data then bin it.
1 = Input previously binned data file. This enables FERD-PC to read in previously binned data without going through the binning process. The user will always be prompted for an input file and path name no matter which option is selected here.

LINE H: 0 = No effect.
1 = Output binned data to file. This option will force FERD-PC to output the binning results in a format that can be used by FERD-PC for prebinned input or by the plotting utility PLOTFERD. The user will be asked to enter the output file and path name if this option is selected.

LINE I: 0 = No Effect.
1 = Output the unfolded results into a punch type file. This is a condensed ASCII data file which enables other programs to read FERD-PC output more easily. The user will be asked to enter the output file and path name if this option is selected.

LINE J: 0 = Use the window matrix listed in block 8. Block 7 is not needed if this option is selected.
1 = Calculate the window matrix using the FWHM data listed in block 7. This will calculate a Gaussian response at each energy bin using the FWHM value. The FWHM is input in percent with the sequence proceeding from the highest to lowest energy interval. Block 8 is not needed if this option is selected.

LINE K: 0 = No Effect.
1 = Output the synthetic windows that are calculated during the unfolding process. This option will output the synthetic windows as a matrix and Gaussian FWHM

percent into a condensed response type data file so it can be displayed by PLOTFERD. The user will be automatically prompted for the output file and path name if this option is selected.

LINE L: NRR = Number of pulse-height bins used in the unfolding process. This number corresponds to the number of rows in the response matrix and the number of bins listed in block 3 and block 4. This parameter can not be edited from FERD-PC since the response matrix must also be adjusted accordingly.

LINE M: NCL = Number of energy bins in block 5. This number corresponds to the number of columns in the window and response matrices. This parameter can not be edited from FERD-PC since the response matrix must also be adjusted accordingly.

LINE N: NRW = Number of window vectors for each energy bin. This number corresponds to the number of rows in the window matrix and size of the energy array listed in block 6. This parameter can not be edited from FERD-PC since the window matrix must also be adjusted accordingly.

LINE O: TAU = Adjustable parameter which determines relative weighing of bias and variance in the estimation process. If TAU is less than the condition indicator, which is calculated by FERD-PC, TAU will be reset to the value of the condition indicator.[1]

LINE P: DNM = Detector normalization factor. Used in the binning process only, it provides a method for normalizing the absolute detector efficiency. The count rate is multiplied by this factor during the binning process.

LINE Q: Default file name for the unformatted response file. When options in lines B or C are selected, this will be the name of the file used by FERD-PC. If B or C options are not used this line will be ignored.

LINE R: RES = Normalization parameter used in calculating the window response matrix from block 7. This parameter is used to multiply each resolution listed in block 7. This line is ignored if block 8 is used to supply the window response. This line must be followed by a blank line.

4.3 BLOCK (2) - INTEGRAL OUTPUT

Block 2 contains the integration intervals used by FERD-PC. The program integrates the unfolded results over these energy intervals and includes it in the final output file along with the unfolded data. The user can specify any number of intervals providing these intervals are within the boundaries of the output energy bins listed in block 4. The energy data given here must be in units of MeV and cannot exceed 50 intervals. This block must be preceded by a blank line and must have a blank line after the last energy interval. The format used to read in these intervals is (6X,2F8.0).

4.4. BLOCKS (3) & (4) - PULSE-HEIGHT BINS

Block 3 contains the midpoints, in light units, of each pulse-height bin that is used in FERD-PC. These energies correspond to each row of the response matrix and are used to format the input data during the binning process. The format used to read in these midpoints is (5X,8F9.0).

Block 4 contains the width, in light units, of each pulse-height bin listed in block 3. It is used only by the binning process. The format used to read in these intervals is (5X,8F9.0). Blocks 3 and 4 are not needed if prebinned data are used for input.

4.5 BLOCK (5) - ENERGY BINS

Block 5 contains the array of energy bins which the pulse-height data are converted to during the unfolding process. These points correspond to the column energies in the response and window matrices. While not directly used by the unfolding process, these points are needed for the integration routine and for clarifying output. The format used to read these mid-points is (5X,7F10.0) and must be followed by a blank line after the last data line. See Figure 4.1.

4.6 BLOCK (6) - WINDOW ROW ENERGIES

Block 6 contains the array of energies used for each row in the window matrix. It is along these energies that the Gaussian window functions are calculated and correspond to each row in the window matrix. The column and row energies of the window matrix do not have to be identical as given in the sample file. In fact, it is recommended that the minimum and maximum values of block 6 should be sufficiently inside the total range of the column energies so that none of the tabulated windows are incomplete (cut off before they go to zero).[1] The format used to read in these energies is (5X,7F10.0) and must be followed by a blank line after the last data line. See Figure 4.1.

4.7 BLOCK (7) - GAUSSIAN FWHM

Block 7 contains the FWHM ratio, in percent, which FERD-PC uses to calculate the window matrix. This option is selected in line J of block 1 and is used in lieu of reading the windows directly from block 8. If this option to calculate the window matrix is chosen, block 8 will not be used. Likewise, if FERD-PC uses block 8 for the window input, block 7 will be ignored. The format used to read in these parameters is (5X,7F10.0) and must be followed by a blank line after the last data line. See Figure 4.1.

4.8 BLOCK (8) - WINDOW MATRIX

Block 8 contains the window matrix for FERD-PC. This block is used to read the window matrix directly into FERD-PC. If the option to read the matrices from the unformatted response file is selected this block will not be used. Block 8 is also ignored if the option to calculate the windows from block 7 is chosen. It uses column format and is read in by FERD-PC using format (1X,2I3,1X,7F10.0). The last data line in this block must be followed by a blank line. See Figure 4.1.

4.9 BLOCK (9) - RESPONSE MATRIX

Block 9 contains the response matrix for FERD-PC. The parameters used here are listed as a percent value and represents the probability of an event within a specific pulse-height bin given an incident energy interval. The matrix uses column format and is read in by FERD-PC using format (2I3,8F9.0). Each column represents the incident or true energy interval while each row represents the pulse-height bin. This block is ignored if the option to read the matrices from the unformatted response file is selected. The last line of this matrix must be followed by a blank line or zeros. See Figure 4.1.

5. PULSE-HEIGHT DATA INPUT

5.1 DECLARING THE TYPE OF INPUT FILE

There are two types of pulse-height data input that may be used with FERD-PC. The first type consists of channel pulse-height data in which FERD-PC prompts the user for calibration parameters, then bins the data according to the binning parameters in the response file. The second type allows prebinned pulse-height data to be input into FERD-PC, thus eliminating the need to include calibration data and normalization parameters. In this chapter the format of these two types of input files will be discussed and illustrated to provide the user with detailed explanation of how to set up FERD-PC input.

FERD-PC assumes the input file consists of pulse-height data unless the prebinned data flag is set to 1 in the response file. This flag is the seventh control parameter in block 1 of the response file and is identified by the letter G. It can be changed by editing the response file directly or through the modify command at the beginning of FERD-PC.

5.2 CHANNEL PULSE-HEIGHT DATA INPUT

Channel pulse-height data input is assumed to be the default input file in FERD-PC. This type of input file contains the raw counts per channel from a pulse-height analyzer and must be binned into the proper pulse-height bins corresponding to the response matrix prior to the unfolding process. An example of this type of format used by FERD-PC is illustrated in Figure 5.1.

The first line of this file is considered to be a file label or title and is ignored by FERD-PC. The second line must contain the format identifier which specifies to FERD-PC the format of any data that follows. FERD-PC reads the format identifier in line two using a (9X,20A1) format. It then uses this format to read in the pulse-height data starting in line three. This way the data may be in any format the user desires starting in line three. The format identifier must

FILE: SAMPLE.FGD FOREGROUND PULSE HEIGHT DATA - PUBE SOURCE
FORMAT = (7X,8I8)

1	0	0	0	0	0	0	0	0
9	1	0	0	0	62	0	0	0
17	1	4	1	0	3	481	58108	70046
25	64210	59479	55192	51943	48653	46102	44153	41811
33	39939	38182	37004	35216	34118	33209	32585	31757
41	30715	29504	29298	28243	27906	26888	25965	25536
49	24973	24351	24112	23596	23306	22645	22384	21825
57	21594	21006	20628	20199	19883	20119	19731	19418
65	18990	18499	18131	18190	17985	17878	17683	17228
73	16999	16542	16399	16157	16081	15844	15589	15383
81	15124	14973	14906	14776	14466	14283	14022	13896
89	13757	13281	13117	13050	12820	12666	12509	12293
97	12219	12040	11922	11738	11739	11478	11552	11115
105	11011	10893	10811	10890	10429	10280	10312	10220
113	9959	9758	9613	9479	9426	9428	9247	9010
121	8963	8853	8731	8527	8550	8313	8345	8183
129	8039	8090	7849	7842	7595	7652	7393	7220
137	7063	7091	7106	7054	6916	6874	6628	6693
145	6584	6388	6504	6503	6220	6271	6075	6018
153	5965	5910	5927	5782	5748	5557	5549	5600
161	5542	5288	5362	5182	5379	5227	5172	5141
169	5151	4971	4944	4944	4807	4961	4751	4792
177	4560	4671	4617	4674	4515	4400	4406	4326
185	4278	4140	4245	4193	4171	4160	4140	3985
193	4038	3966	3909	3866	3827	3822	3811	3718
201	3671	3664	3614	3631	3601	3532	3551	3413
209	3409	3433	3338	3275	3298	3068	3223	3117
217	3175	3052	3140	2985	2890	2912	2997	2838
225	2849	2784	2812	2770	2716	2720	2756	2664
233	2624	2601	2557	2493	2484	2500	2361	2443
241	2352	2384	2340	2319	2301	2339	2197	2144
249	2082	2139	2134	2015	2107	2050	2056	2041
257	1955	1974	2010	1953	1955	1830	1927	1833
265	1856	1793	1853	1740	1714	1754	1745	1745
273	1701	1726	1635	1616	1643	1637	1520	1588
281	1623	1551	1508	1568	1559	1577	1479	1479
289	1356	1456	1461	1347	1383	1453	1373	1327
297	1438	1411	1295	1264	1300	1295	1298	1279
305	1347	1336	1266	1290	1252	1214	1265	1200
313	1175	1160	1125	1127	1149	1198	1148	1168
321	1125	1109	1145	1052	1105	1100	1098	1068
329	1092	1119	1051	1104	1070	1004	1073	1001
337	983	976	1028	940	991	949	984	961
345	952	923	915	934	897	909	875	904
353	858	833	862	878	852	861	802	815
361	799	757	808	861	772	734	732	752
369	749	784	799	691	699	750	731	674
377	760	687	643	651	698	654	672	682
385	687	622	619	650	631	652	611	581
393	635	617	582	632	589	560	599	522
401	543	602	540	538	560	588	509	516
409	486	503	519	492	477	478	479	473
417	446	473	439	440	466	404	441	436
425	427	425	419	393	421	370	388	366
433	374	378	363	316	368	371	353	366

Figure 5.1. Sample input file for channel pulse-height data.

specify an integer format or an error will occur when running FERD-PC. See the example in Figure 5.1.

The program will continue reading the pulse-height data until an end of file is encountered so caution should be taken not to add extraneous lines or data at the end of this file. A maximum of 2048 channels are allowed in the original pulse-height data array. This can be modified by changing the dimension statements for the variables IA,A,RSUM,XL in subroutine BIN; however, care must be taken not to exceed the computer's or compiler's memory allocation. The variable NMAX at the beginning of the same subroutine must also be changed to reflect the maximum possible value of the new data array.

When reviewing the pulse-height data that are output to the log file, one may not see the same number of channels that the input file contains because once all the data are read in from the input file, FERD-PC determines the maximum non-zero channel and uses this number as the maximum channel. This process improves the binning speed for data containing a large number of zero counts in the upper end of the spectrum.

If pulse-height data are used as the input file for FERD-PC, the program will automatically prompt the user for foreground and background file names. If a background file is not to be included the user enters a blank line when prompted for the file name. For each of these files the user will be prompted for the calibration channel, light units at the calibration channel, a zero channel, and a power normalization factor if the proper flag is set in the response file. The foreground and background files may have different calibration channels with overlapping gains since FERD-PC bins each file separately. The user will also be prompted for the number of smoothing iterations to be performed on each of the input files prior to the binning process. Smoothing is normally required to reduce large fluctuations in the unfolded results when counting statistics are very poor.

5.3 PREBINNED PULSE-HEIGHT DATA INPUT

The second type of input consists of previously binned data. This is a file that contains the pulse-height data already placed into the proper bins corresponding to the response matrix. It consists of the count rate per bin and the standard deviation for each bin with all calibration and normalization parameters having been taken into account. This file may be created through other software and then used in FERD-PC or it may be generated by FERD-PC when binning a pulse-height spectrum. This file may be a benefit when planning to unfold the same data set more than once, since this would eliminate the need to re-enter the calibration parameters each time. See Section 6.2 for more information on binned data output. To use this type of input file in FERD-PC, the prebinned data input flag must be set in the response file as described in Section 5.1.

FERD-PC assumes the first line of this file as the run title and will not prompt the user for a run title if prebinned input is used. The second line of this file must contain the format identifier which specifies to FERD-PC the format of any data that follows. FERD-PC reads the format identifier in line two using a (9X,20A1) format. It then uses this format to read in the following data starting in line three. This way the data may be in any format the user desires. An example of the type of format that one can use is illustrated in Figure 5.2. In this example the first column of data is nothing more than the bin number and is not used by FERD-PC. The second column is the center of the pulse-height bin in light units and must correspond directly to the center of each row in the response matrix. The third column contains the count rate in counts per second per bin width with the bin width given in light units. The fourth column contains the standard deviation for each bin and is given in the same units as column three. The standard deviation for each pulse-height bin must be a non-zero number or a floating point error will occur when running FERD-PC.

```
FILE:  SAMPLE.BIN                SAMPLE INPUT - PUBE SOURCE
FORMAT = (12X,3F15.0)
         1    1.0750000E-01  9.9553970E+00  4.6283260E-02
         2    1.2250000E-01  8.9122520E+00  4.3324060E-02
         3    1.3750000E-01  8.1666920E+00  4.1062240E-02
         4    1.5250000E-01  7.4834350E+00  3.9032770E-02
         5    1.6750000E-01  6.9360550E+00  3.7469160E-02
         6    1.8250000E-01  6.4461210E+00  3.6123280E-02
         7    2.0500000E-01  1.2065900E+01  4.8979070E-02
         8    2.3500000E-01  1.0981930E+01  4.6318830E-02
         9    2.6500000E-01  9.9830830E+00  4.3745260E-02
        10    2.9500000E-01  9.3517260E+00  4.1971430E-02
        11    3.2500000E-01  8.7530060E+00  4.0299710E-02
        12    3.5500000E-01  8.1750850E+00  3.8741560E-02
        13    3.8500000E-01  7.8675830E+00  3.7697010E-02
        14    4.1500000E-01  7.4016830E+00  3.6419800E-02
        15    4.5500000E-01  1.1658470E+01  4.5197050E-02
        16    5.0500000E-01  1.0764900E+01  4.2988260E-02
        17    5.5500000E-01  1.0025570E+01  4.1019220E-02
        18    6.0500000E-01  9.1918530E+00  3.8735270E-02
        19    6.5500000E-01  8.6289340E+00  3.7022090E-02
        20    7.0500000E-01  8.0335250E+00  3.5330470E-02
        21    7.5500000E-01  7.4243880E+00  3.3628030E-02
        22    8.0500000E-01  6.8866580E+00  3.2087460E-02
        23    8.5500000E-01  6.3501800E+00  3.0591680E-02
        24    9.0500000E-01  5.8640620E+00  2.9253490E-02
        25    9.7500000E-01  9.4058390E+00  3.6864970E-02
        26    1.0650000E+00  8.2260060E+00  3.4281740E-02
                                   .
                                   .
                                   .
        48    4.6150000E+00  9.1968770E-02  3.4938180E-03
        49    4.9250000E+00  2.4812710E-02  1.8222430E-03
        50    5.2350000E+00  4.9581390E-03  9.3261320E-04
        51    5.5450000E+00  2.4346010E-03  6.0566100E-04
        52    5.8550000E+00  1.0654220E-03  4.0549700E-04
        53    6.1650000E+00  1.2500000E-03  3.9528470E-04
        54    6.4750000E+00  5.0000000E-04  3.5355340E-04
        55    6.9100000E+00  1.7500000E-03  5.8630200E-04
        56    7.4700000E+00  0.0000000E+00  1.7677670E-04
        57    8.0300000E+00  0.0000000E+00  1.7677670E-04
        58    8.5900000E+00  0.0000000E+00  1.7677670E-04
        59    9.1500000E+00  0.0000000E+00  1.7677670E-04
        60    9.7100000E+00  0.0000000E+00  1.7677670E-04
        61    1.0270000E+01  0.0000000E+00  1.7677670E-04
        62    1.0830000E+01  0.0000000E+00  1.7677670E-04
        63    1.1390000E+01  0.0000000E+00  1.7677670E-04
        64    1.1950000E+01  0.0000000E+00  1.7677670E-04
        65    1.2510000E+01  0.0000000E+00  1.7677670E-04
        66    1.3070000E+01  0.0000000E+00  1.7677670E-04
        67    1.3630000E+01  0.0000000E+00  1.7677670E-04
        68    1.4190000E+01  0.0000000E+00  1.7677670E-04
```

Figure 5.2. Sample input file for pre-binned pulse-height data.

6. FERD-PC OUTPUT

6.1 MAIN OUTPUT FILE

Every time FERD-PC is run, the user will be asked to enter an output file name for the unfolded results. This output file is referred to as the main output file in this manual. It contains the unfolded solution and consists of two sections as shown in Figure 6.1.

The first section contains the unfolded spectrum and corresponds to each energy bin in the response matrix. It is output as a lower and upper limit which represents the confidence interval of the unfolded result at each energy. The energy given here is in units of MeV and represents the midpoint of each bin. The confidence interval corresponds to the statistical uncertainty which is calculated from the pulse-height data and propagated through the unfolding process. When unfolding pulse-height spectra, each interval is in units of flux per MeV, $n/(cm^2 \cdot s \cdot MeV)$. If the input data are normalized to a power level, which can be configured using the control parameters. The output is in units of $n/(cm^2 \cdot s \cdot MeV \cdot kW)$.

The second section contains the integrated output corresponding to each energy interval listed in block 2 of the response file. The first two columns represent the lower and upper bounds of each energy interval and are given units of MeV. The third column represents the unfolded spectrum integrated over the given energy interval and is in units of flux, $n/(cm^2 \cdot s)$, or flux per kilowatt, $n/(cm^2 \cdot s \cdot kW)$, which ever may apply. The fourth column is the statistical error in the integrated result and is also given in the same units as column three. The energy intervals output in this second section can be controlled by editing block 2 of the response data file. A maximum of 50 intervals may be designated here and may be in any order the user chooses.

6.2 BINNED DATA OUTPUT

If control parameter H is set to 1 in the response file, the user will be prompted for a file name for the output file of the binned

SAMPLE RUN - PUBE SOURCE

NEUTRON ENERGY (MeV)	FLUX (n/(cm2*s*MeV)) LOWER LIMIT	UPPER LIMIT	NEUTRON ENERGY (MeV)	FLUX (n/(cm2*s*MeV)) LOWER LIMIT	UPPER LIMIT
8.11E-01	5.74E+00	6.62E+00	5.94E+00	6.30E+00	6.49E+00
9.07E-01	7.90E+00	8.45E+00	6.25E+00	5.89E+00	6.14E+00
1.01E+00	7.62E+00	8.07E+00	6.55E+00	5.81E+00	6.02E+00
1.11E+00	6.39E+00	6.91E+00	6.84E+00	5.95E+00	6.10E+00
1.20E+00	6.29E+00	6.73E+00	7.24E+00	6.06E+00	6.19E+00
1.31E+00	6.27E+00	6.69E+00	7.74E+00	5.35E+00	5.49E+00
1.41E+00	6.09E+00	6.53E+00	8.24E+00	4.02E+00	4.13E+00
1.51E+00	6.01E+00	6.47E+00	8.76E+00	2.76E+00	2.83E+00
1.61E+00	5.96E+00	6.38E+00	9.26E+00	2.05E+00	2.10E+00
1.71E+00	5.96E+00	6.38E+00	9.74E+00	1.54E+00	1.58E+00
1.81E+00	6.21E+00	6.62E+00	1.03E+01	9.63E-01	9.96E-01
1.93E+00	6.92E+00	7.34E+00	1.08E+01	4.62E-01	4.84E-01
2.10E+00	7.82E+00	8.26E+00	1.12E+01	1.84E-01	1.99E-01
2.30E+00	7.78E+00	8.21E+00	1.18E+01	4.08E-02	4.99E-02
2.50E+00	8.52E+00	8.89E+00	1.24E+01	-1.20E-02	-5.17E-03
2.70E+00	1.03E+01	1.07E+01	1.32E+01	-1.37E-02	-8.73E-03
2.90E+00	1.18E+01	1.21E+01	1.40E+01	-5.83E-03	-1.91E-03
3.10E+00	1.25E+01	1.30E+01	1.48E+01	-1.86E-03	2.49E-03
3.30E+00	1.27E+01	1.30E+01	1.56E+01	-3.21E-03	-4.18E-05
3.50E+00	1.19E+01	1.24E+01	1.65E+01	-3.07E-03	-6.33E-04
3.71E+00	1.10E+01	1.13E+01	1.75E+01	-1.34E-03	8.30E-04
3.91E+00	1.06E+01	1.08E+01	1.85E+01	-7.75E-04	1.16E-03
4.15E+00	1.09E+01	1.12E+01	1.95E+01	-6.68E-04	1.15E-03
4.45E+00	1.15E+01	1.18E+01	2.05E+01	-1.46E-03	1.21E-03
4.75E+00	1.12E+01	1.14E+01	2.16E+01	-1.71E-03	1.08E-03
5.04E+00	9.87E+00	1.01E+01	2.26E+01	-1.07E-03	9.53E-04
5.34E+00	8.30E+00	8.50E+00	2.35E+01	-6.87E-04	9.85E-04
5.64E+00	7.05E+00	7.28E+00			

E1 (MeV)	E2 (MeV)	INTEGRAL (n/(cm2*s))	ERROR (n/(cm2*s))
.811	1.000	1.47E+00	5.57E-02
1.000	1.200	1.38E+00	4.90E-02
1.200	1.600	2.54E+00	8.70E-02
1.600	2.000	2.62E+00	8.32E-02
2.000	3.000	9.41E+00	2.03E-01
3.000	4.000	1.19E+01	1.89E-01
4.000	6.000	1.91E+01	2.34E-01
6.000	8.000	1.17E+01	1.67E-01
8.000	10.000	5.24E+00	6.92E-02
10.000	12.000	8.58E-01	1.98E-02
12.000	16.000	-1.55E-02	9.38E-03
1.500	15.000	6.15E+01	9.94E-01

Figure 6.1. Main output file from sample run.

data. The file created here will contain data generated by the binning section of FERD-PC and corresponds to each light unit bin of the response matrix. It is this data file that is used by the unfolding process in FERD-PC and may also be used as a prebinned input file, already discussed in Chapter 5. This binned data file is also output into the log file if the log output option is selected. An example of the binned data output is illustrated in Figure 6.2.

The first line of the binned output file is the title of the run which is input by the user at the beginning of FERD-PC. The second line is the format identifier used by FERD-PC. This is required if this file is to be used as prebinned input. The third line is the beginning of the binned data and contains four columns.

The first column is the data line number for use in cross referencing other files and the response matrix. The second column contains the center of each pulse-height bin and is given in light units. This number corresponds to the center of each bin in the response matrix. The third column is the pulse-height count rate and is given in counts per second per bin. If a background file is binned together with a foreground file, this column contains the difference between the two files. Column three is also multiplied by the detector normalization factor listed as line P in the control parameters. It is also divided by the power level if the user employs the power normalization option. The fourth and last column represents the standard deviation of column three determined by the statistical error in each pulse-height bin. Since the standard deviation must be a non-zero number, FERD-PC will always insert a non-zero term here. If the count rate is zero in column three, FERD-PC will use one half of the smallest standard deviation encountered up to that point. The deviation is given in the same units as column three.

6.3 LOG OUTPUT FILE

The log output file generated by FERD-PC contains a record of the various parameters which have been input and calculated by FERD-PC. This provides a method of recording all input and control parameters

SAMPLE RUN - PUBE SOURCE

FORMAT = (12X,3F15.0)

1	1.0750000E-01	9.9553970E+00	4.6283260E-02
2	1.2250000E-01	8.9122520E+00	4.3324060E-02
3	1.3750000E-01	8.1666920E+00	4.1062240E-02
4	1.5250000E-01	7.4834350E+00	3.9032780E-02
5	1.6750000E-01	6.9360550E+00	3.7469160E-02
6	1.8250000E-01	6.4461210E+00	3.6123290E-02
7	2.0500000E-01	1.2065900E+01	4.8979070E-02
8	2.3500000E-01	1.0981930E+01	4.6318830E-02
9	2.6500000E-01	9.9830830E+00	4.3745260E-02
10	2.9500000E-01	9.3517260E+00	4.1971430E-02
11	3.2500000E-01	8.7530060E+00	4.0299710E-02
12	3.5500000E-01	8.1750850E+00	3.8741560E-02
13	3.8500000E-01	7.8675830E+00	3.7697010E-02
14	4.1500000E-01	7.4016830E+00	3.6419800E-02
15	4.5500000E-01	1.1658470E+01	4.5197050E-02
16	5.0500000E-01	1.0764900E+01	4.2988260E-02
17	5.5500000E-01	1.0025570E+01	4.1019220E-02
18	6.0500000E-01	9.1918530E+00	3.8735270E-02
19	6.5500000E-01	8.6289340E+00	3.7022090E-02
20	7.0500000E-01	8.0335250E+00	3.5330470E-02
21	7.5500000E-01	7.4243880E+00	3.3628030E-02
22	8.0500000E-01	6.8866580E+00	3.2087460E-02
23	8.5500000E-01	6.3501800E+00	3.0591680E-02
24	9.0500000E-01	5.8640620E+00	2.9253490E-02
25	9.7500000E-01	9.4058390E+00	3.6864970E-02
26	1.0650000E+00	8.2260060E+00	3.4281740E-02
27	1.1550000E+00	7.2784230E+00	3.2158250E-02
		.	
		.	
		.	
46	3.9950000E+00	4.5107120E-01	7.6571850E-03
47	4.3050000E+00	2.4312960E-01	5.6076390E-03
48	4.6150000E+00	9.1968770E-02	3.4938180E-03
49	4.9250000E+00	2.4812710E-02	1.8222430E-03
50	5.2350000E+00	4.9581390E-03	9.3261320E-04
51	5.5450000E+00	2.4346010E-03	6.0566100E-04
52	5.8550000E+00	1.0654220E-03	4.0549700E-04
53	6.1650000E+00	1.2500000E-03	3.9528470E-04
54	6.4750000E+00	5.0000000E-04	3.5355340E-04
55	6.9100000E+00	1.7500000E-03	5.8630200E-04
56	7.4700000E+00	0.0000000E+00	1.7677670E-04
57	8.0300000E+00	0.0000000E+00	1.7677670E-04
58	8.5900000E+00	0.0000000E+00	1.7677670E-04
59	9.1500000E+00	0.0000000E+00	1.7677670E-04
60	9.7100000E+00	0.0000000E+00	1.7677670E-04
61	1.0270000E+01	0.0000000E+00	1.7677670E-04
62	1.0830000E+01	0.0000000E+00	1.7677670E-04
63	1.1390000E+01	0.0000000E+00	1.7677670E-04
64	1.1950000E+01	0.0000000E+00	1.7677670E-04
65	1.2510000E+01	0.0000000E+00	1.7677670E-04
66	1.3070000E+01	0.0000000E+00	1.7677670E-04
67	1.3630000E+01	0.0000000E+00	1.7677670E-04
68	1.4190000E+01	0.0000000E+00	1.7677670E-04

Figure 6.2. Binned data output from sample run. This file may also be used for prebinned input.

used by FERD-PC for each run and also contains some important data output from the unfolding process that is not included in the main output file. It also provides the user with a method of debugging certain aspects of FERD-PC and ensures correct input format for all data files. If the log output option is selected using the control parameters, the user will be prompted for a log output file name. The file is opened at the beginning of FERD-PC with data being written to the file throughout the binning and unfolding processes. Since the log output can be a very long file, the only illustration provided will be the last section. However a detailed description of each section is given below.

The first section of the log file contains the control parameters used for the current run. It is set up in a similar format as the response file with a short description following each parameter. The next section contains all input used in the binning process, which includes calibration data, power normalization, live time, and the number of smoothing iterations. The next section contains the counts as read from each pulse-height data file, with the result of any smoothing iterations following each file. Also included is the calculated gain, which is output as light units per channel. If the prebinned input option is selected, the above sections are not output.

Once the binning process is complete, the binned data is output to the log file in the same format as found in the binned data output file. At this point the response and window matrix would also be output to the log file if this additional option is selected using control parameter E. It should be noted here that the windows are output in matrix form and not as a Gaussian resolution. Thus, a considerable amount of space can be used for each log file output so the user has the option to repress the response and window output while still writing other critical parameters to the log file.

The next section contains the memory allocation for array D. This array defines most of the data area needed or used during the unfolding process. The memory allocated is defined by the variable LTH and should correspond to the length of array D in the main program. The 'Required Memory' output here is the amount of memory required by the

program based on the number of rows and columns in the response and window matrices. Subroutine ARRAY gives an approximation for the required length of array D. Normally, not all the memory allocated in this dimension statement will be required to conduct the unfolding process, but is reserved for larger response matrices. This enables the use of different size responses while optimizing the available memory of the computer.

Two parameters calculated by FERD-PC are output next. These are MU and the condition indicator. MU is a parameter described in reference 1 as a value which guarantees that the upper bounds for the unknown solution hold within a probability equal to or greater than 99.95 percent. The other parameter is referred to as the condition indicator. The condition indicator is a measure of the conditioning of the response matrix. It measures the difficulty in inverting the matrix relative to the number of bits of accuracy of the computer. Normally if the condition indicator is greater than 1.0, it suggests that the response matrix has been set up inappropriately, with the wrong interval between bins. If the condition indicator is larger than the value of parameter TAU, which is set by the user, then TAU is reset to the value of the condition indicator.[1]

The last section of the log output file contains the unfolded results. An example of this section is illustrated in Figure 6.3. It is organized into seven columns of data with the first column listing the bin or interval number. The second column contains the energy midpoint of each output interval given in units of MeV. The third and fourth columns contain the lower and upper limits of the unfolded spectrum. These limits are the same numbers as output in the main output file and the punch output file. The fifth column is the average of the lower and upper limits found in the previous two columns. The sixth column represents the confidence band based on the statistical part of the error in the lower biased estimate. The seventh and last column, called QSHOOT, is a measure of the total undershoot or overshoot error expressed as a percentage of the width of the confidence interval. It is calculated from the vector difference between the given window vector and the synthetic window vector. If

	ENERGY	PLO	PUP	PAVG	ERR	QSHOOT
1	2.347E+01	-6.8907E-04	9.8574E-04	1.4833E-04	1.6489E-03	-1.2727E+00
2	2.256E+01	-1.0696E-03	9.5285E-04	-5.8371E-05	2.0194E-03	7.7014E-01
3	2.157E+01	-1.7047E-03	1.0741E-03	-3.1532E-04	2.7344E-03	1.2199E+00
4	2.055E+01	-1.4605E-03	1.2077E-03	-1.2640E-04	2.6236E-03	3.0593E+00
5	1.949E+01	-6.6829E-04	1.1531E-03	2.4239E-04	1.8057E-03	-7.3222E-01
6	1.852E+01	-7.7431E-04	1.1548E-03	1.9026E-04	1.8974E-03	-1.5420E+00
7	1.753E+01	-1.3431E-03	8.3076E-04	-2.5615E-04	2.1625E-03	1.2778E-01
8	1.653E+01	-3.0614E-03	-6.3142E-04	-1.8464E-03	2.3741E-03	3.4420E-01
9	1.562E+01	-3.2125E-03	-4.1413E-05	-1.6270E-03	3.0940E-03	3.5726E-01
10	1.479E+01	-1.8550E-03	2.4892E-03	3.1711E-04	4.2453E-03	-3.1537E+00
11	1.402E+01	-5.8282E-03	-1.9135E-03	-3.8708E-03	3.8867E-03	6.1848E-02
12	1.320E+01	-1.3710E-02	-8.7311E-03	-1.1221E-02	4.9225E-03	4.6593E-02
13	1.239E+01	-1.2018E-02	-5.1782E-03	-8.5982E-03	6.6986E-03	3.8765E-02
14	1.175E+01	4.0765E-02	4.9895E-02	4.5330E-02	9.0056E-03	0.0000E+00
15	1.124E+01	1.8407E-01	1.9869E-01	1.9138E-01	1.4439E-02	-1.3695E-04
16	1.077E+01	4.6238E-01	4.8440E-01	4.7339E-01	2.1693E-02	-5.5862E-03
17	1.025E+01	9.6334E-01	9.9648E-01	9.7991E-01	3.2810E-02	-2.3166E-03
18	9.741E+00	1.5411E+00	1.5839E+00	1.5625E+00	4.2543E-02	-3.7021E-04
19	9.261E+00	2.0460E+00	2.1037E+00	2.0748E+00	5.7329E-02	-2.3512E-03
20	8.757E+00	2.7588E+00	2.8287E+00	2.7937E+00	6.9358E-02	-1.2408E-03
21	8.236E+00	4.0185E+00	4.1264E+00	4.0725E+00	1.0638E-01	-4.3024E-03
22	7.736E+00	5.3545E+00	5.4858E+00	5.4201E+00	1.3013E-01	-1.6634E-03
23	7.235E+00	6.0612E+00	6.1881E+00	6.1247E+00	1.2644E-01	-3.3759E-04
24	6.839E+00	5.9517E+00	6.1015E+00	6.0266E+00	1.4427E-01	-1.3477E-03
25	6.555E+00	5.8084E+00	6.0234E+00	5.9159E+00	2.0283E-01	-1.2324E-03
26	6.254E+00	5.8905E+00	6.1395E+00	6.0150E+00	2.3522E-01	-1.5913E-03
27	5.938E+00	6.2961E+00	6.4851E+00	6.3906E+00	1.8519E-01	-2.0995E-03
28	5.645E+00	7.0470E+00	7.2766E+00	7.1618E+00	2.1447E-01	-4.7232E-04
29	5.345E+00	8.2969E+00	8.5072E+00	8.4020E+00	1.9726E-01	-2.3900E-04
30	5.042E+00	9.8726E+00	1.0096E+01	9.9845E+00	2.0938E-01	-2.7355E-04
31	4.752E+00	1.1173E+01	1.1408E+01	1.1290E+01	2.3130E-01	-3.7037E-06
32	4.453E+00	1.1502E+01	1.1749E+01	1.1625E+01	2.4383E-01	-2.8946E-04
33	4.152E+00	1.0868E+01	1.1161E+01	1.1014E+01	2.8093E-01	-1.9696E-04
34	3.906E+00	1.0551E+01	1.0854E+01	1.0702E+01	2.8946E-01	-2.6290E-04
35	3.707E+00	1.1000E+01	1.1333E+01	1.1166E+01	3.2162E-01	-1.5956E-04
36	3.500E+00	1.1937E+01	1.2369E+01	1.2153E+01	3.8806E-01	-7.0565E-04
37	3.296E+00	1.2677E+01	1.3043E+01	1.2860E+01	3.4160E-01	-1.9318E-04
38	3.097E+00	1.2548E+01	1.2998E+01	1.2773E+01	4.0413E-01	-6.6653E-04
39	2.898E+00	1.1729E+01	1.2148E+01	1.1938E+01	3.7956E-01	-4.3731E-04
40	2.702E+00	1.0257E+01	1.0658E+01	1.0458E+01	3.7919E-01	-4.2537E-04
41	2.498E+00	8.5199E+00	8.8857E+00	8.7028E+00	3.6292E-01	-3.0639E-04
42	2.296E+00	7.7833E+00	8.2083E+00	7.9958E+00	4.1546E-01	-1.4287E-03
43	2.099E+00	7.8375E+00	8.2607E+00	8.0491E+00	4.1328E-01	-1.8802E-03
44	1.932E+00	6.9198E+00	7.3385E+00	7.1292E+00	4.1314E-01	-9.2729E-04
45	1.812E+00	6.1938E+00	6.6233E+00	6.4085E+00	4.0919E-01	-1.5448E-03
46	1.709E+00	5.9557E+00	6.3720E+00	6.1639E+00	3.9539E-01	-3.8497E-03
47	1.609E+00	5.9556E+00	6.3741E+00	6.1648E+00	3.9300E-01	-1.1319E-03
48	1.510E+00	6.0145E+00	6.4527E+00	6.2336E+00	4.1829E-01	-6.2080E-03
49	1.412E+00	6.0784E+00	6.5269E+00	6.3027E+00	4.2441E-01	-9.1896E-03
50	1.309E+00	6.2537E+00	6.6765E+00	6.4651E+00	4.0000E-01	-3.0612E-03
51	1.204E+00	6.2669E+00	6.6983E+00	6.4826E+00	4.0807E-01	-1.3675E-03
52	1.106E+00	6.3911E+00	6.9164E+00	6.6537E+00	4.5254E-01	-1.9459E-03
53	1.006E+00	7.6169E+00	8.0474E+00	7.8322E+00	4.0987E-01	-2.7143E-04
54	9.069E-01	7.8967E+00	8.4461E+00	8.1714E+00	4.9293E-01	-5.2455E-03
55	8.113E-01	5.7384E+00	6.6468E+00	6.1926E+00	8.3447E-01	-3.4426E-04

Figure 6.3. Log output from sample run (last section only).

QSHOOT exceeds 20 percent then the energy mesh points may need to be taken closer together.[1]

6.4 PUNCH OUTPUT FILE

If the punch file output is selected, the unfolded results will be output into a punch type file. The purpose of this file is to provide other programs with the capability to access the unfolded data more easily than the main output file. An example of this punch output file is illustrated in Figure 6.4. The first line in this file contains the title of the run and is then followed by three columns of data. The first column contains the center of each energy bin and is given in units of MeV. The second and third columns correspond to the lower and upper limits of the unfolded results and are in units of flux per MeV, $n/(cm^2 \cdot s \cdot MeV)$. If the data is normalized to a power level it is given in units of flux per MeV per kilowatt, $n/(cm^2 \cdot s \cdot MeV \cdot kW)$. The format that can be used to read in this data is (3E15.5). PLOTFERD is capable of reading either the main output file or the punch output file as is described in Appendix B.

6.5 SYNTHETIC WINDOW OUTPUT

The user has the option to output the synthetic window data FERD-PC generates during the unfolding process. The synthetic windows are output in a similar format as the response data file consisting of blocks 5 through 8 as illustrated in Figure 6.5. This file can then be used by PLOTFERD exactly like any other response file to graphically display the synthetic window matrix or the Gaussian FWHM resolutions. Block 5 in this output file is the midpoint of each energy bin corresponding to each column of the response and window matrices. Block 6 is the energy midpoint of each window vector corresponding to each row in the window matrix. The FWHM window resolutions in block 7 are calculated from the peak of each window assuming the response to be Gaussian in shape. Block 8 is the synthetic window matrix as determined by the unfolding process.

SAMPLE RUN - PUBE SOURCE

.81130E+00	.57384E+01	.66179E+01
.90690E+00	.78967E+01	.84461E+01
.10060E+01	.76166E+01	.80683E+01
.11060E+01	.63927E+01	.69148E+01
.12040E+01	.62864E+01	.67256E+01
.13090E+01	.62743E+01	.66904E+01
.14120E+01	.60926E+01	.65321E+01
.15100E+01	.60134E+01	.64692E+01
.16090E+01	.59647E+01	.63812E+01
.17090E+01	.59637E+01	.63768E+01
.18120E+01	.62096E+01	.66233E+01
.19320E+01	.69197E+01	.73389E+01
.20990E+01	.78214E+01	.82638E+01
.22960E+01	.77833E+01	.82100E+01
.24980E+01	.85200E+01	.88857E+01
.27020E+01	.10259E+02	.10656E+02
.28980E+01	.11753E+02	.12147E+02
.30970E+01	.12548E+02	.13000E+02
.32960E+01	.12666E+02	.13043E+02
.35000E+01	.11938E+02	.12370E+02
.37070E+01	.10981E+02	.11323E+02
.39060E+01	.10551E+02	.10849E+02
.41520E+01	.10869E+02	.11154E+02
.44530E+01	.11500E+02	.11753E+02
.47520E+01	.11173E+02	.11408E+02
.50420E+01	.98727E+01	.10089E+02
.53450E+01	.82972E+01	.85048E+01
.56450E+01	.70465E+01	.72759E+01
.59380E+01	.62976E+01	.64897E+01
.62540E+01	.58894E+01	.61395E+01
.65550E+01	.58087E+01	.60231E+01
.68390E+01	.59514E+01	.61014E+01
.72350E+01	.60611E+01	.61885E+01
.77360E+01	.53545E+01	.54872E+01
.82360E+01	.40184E+01	.41263E+01
.87570E+01	.27588E+01	.28286E+01
.92610E+01	.20459E+01	.21037E+01
.97410E+01	.15411E+01	.15838E+01
.10250E+02	.96338E+00	.99647E+00
.10770E+02	.46237E+00	.48439E+00
.11240E+02	.18407E+00	.19869E+00
.11750E+02	.40769E-01	.49897E-01
.12390E+02	-.12001E-01	-.51741E-02
.13200E+02	-.13708E-01	-.87331E-02
.14020E+02	-.58334E-02	-.19125E-02
.14790E+02	-.18559E-02	.24881E-02
.15620E+02	-.32142E-02	-.41778E-04
.16530E+02	-.30677E-02	-.63274E-03
.17530E+02	-.13436E-02	.83048E-03
.18520E+02	-.77459E-03	.11578E-02
.19490E+02	-.66824E-03	.11546E-02
.20550E+02	-.14597E-02	.12107E-02
.21570E+02	-.17074E-02	.10796E-02
.22560E+02	-.10703E-02	.95257E-03
.23470E+02	-.68700E-03	.98459E-03

Figure 6.4. Punch file output from sample run.

SAMPLE RUN - PUBE SOURCE

(5) ENERGY BINS FOR SYNTHETIC WINDOW OUTPUT IN MeV

2.347E+01	2.256E+01	2.157E+01	2.055E+01	1.949E+01	1.852E+01	1.753E+01
1.653E+01	1.562E+01	1.479E+01	1.402E+01	1.320E+01	1.239E+01	1.175E+01
1.124E+01	1.077E+01	1.025E+01	9.741E+00	9.261E+00	8.757E+00	8.236E+00
7.736E+00	7.235E+00	6.839E+00	6.555E+00	6.254E+00	5.938E+00	5.645E+00
5.345E+00	5.042E+00	4.752E+00	4.453E+00	4.152E+00	3.906E+00	3.707E+00
3.500E+00	3.296E+00	3.097E+00	2.898E+00	2.702E+00	2.498E+00	2.296E+00
2.099E+00	1.932E+00	1.812E+00	1.709E+00	1.609E+00	1.510E+00	1.412E+00
1.309E+00	1.204E+00	1.106E+00	1.006E+00	9.069E-01	8.113E-01	

(6) ENERGY VECTORS FOR SYNTHETIC WINDOWS IN MeV

2.347E+01	2.256E+01	2.157E+01	2.055E+01	1.949E+01	1.852E+01	1.753E+01
1.653E+01	1.562E+01	1.479E+01	1.402E+01	1.320E+01	1.239E+01	1.175E+01
1.124E+01	1.077E+01	1.025E+01	9.741E+00	9.261E+00	8.757E+00	8.236E+00
7.736E+00	7.235E+00	6.839E+00	6.555E+00	6.254E+00	5.938E+00	5.645E+00
5.345E+00	5.042E+00	4.752E+00	4.453E+00	4.152E+00	3.906E+00	3.707E+00
3.500E+00	3.296E+00	3.097E+00	2.898E+00	2.702E+00	2.498E+00	2.296E+00
2.099E+00	1.932E+00	1.812E+00	1.709E+00	1.609E+00	1.510E+00	1.412E+00
1.309E+00	1.204E+00	1.106E+00	1.006E+00	9.069E-01	8.113E-01	

(7) SYNTHETIC WINDOWS IN GUASSIAN FWHM (%)

16.6290	16.7113	16.7395	16.6253	16.8601	16.9003	16.9681
16.9623	16.9129	17.1799	17.2234	17.3329	17.3925	17.5032
17.5854	17.6704	17.7485	17.8534	17.9525	18.0692	18.1994
18.3447	18.5085	18.6510	18.7564	18.8819	19.0477	19.1995
19.3734	19.5665	19.7727	20.0110	20.2795	20.5256	20.7513
20.9971	21.2916	21.5894	21.9326	22.3164	22.7621	23.2683
23.8500	24.4236	24.8864	25.3314	25.8095	26.3365	26.9138
27.5999	28.3964	29.2458	30.2583	31.4164	32.7962	

(8) SYNTHETIC WINDOW MATRIX

1	1	2.4071E-1	2.0673E-1	1.2676E-1	5.1854E-2	1.4517E-2	3.7231E-3	4.1046E-4
1	8	6.1110E-4	4.7484E-4	4.0036E-4	6.1161E-5	3.2754E-4	1.1098E-4	7.2049E-5
1	15	2.0122E-4	3.9468E-6	2.2945E-5	3.4401E-6	2.9620E-6	1.1111E-6	1.0850E-6
1	22	6.9390E-7	3.7411E-7	8.4884E-8	1.1837E-7	7.2344E-8	8.8500E-8	5.6924E-8
1	29	4.8555E-8	7.7291E-8	6.1737E-8	7.3566E-8	7.6310E-8	8.5691E-8	8.9990E-8
1	36	1.0069E-7	1.0439E-7	1.1912E-7	1.3703E-7	1.4970E-7	1.6981E-7	1.9910E-7
1	43	2.3290E-7	2.6299E-7	3.0538E-7	3.3400E-7	3.5793E-7	4.1153E-7	3.7704E-7
1	50	3.9277E-7	4.0127E-7	3.7008E-7	2.2695E-7	7.6402E-8	3.3766E-8	0.0000E+0
2	1	2.1230E-1	2.4918E-1	2.0586E-1	1.1433E-1	3.9816E-2	1.1071E-2	2.4790E-3
2	8	6.6303E-4	7.1261E-4	1.1452E-5	1.6842E-4	1.5654E-5	6.7986E-5	3.1051E-5
2	15	1.1154E-6	1.1175E-5	2.3501E-6	1.9939E-6	8.5810E-7	4.7133E-7	2.2167E-7
2	22	1.9093E-7	4.1260E-8	2.3426E-8	1.2895E-8	1.3170E-8	1.0461E-8	1.1363E-8
2	29	1.2306E-8	8.9635E-9	1.4304E-8	1.5872E-8	1.7119E-8	1.6373E-8	1.8349E-8
2	36	1.8437E-8	1.6319E-8	2.2440E-8	3.0626E-8	2.9370E-8	3.4204E-8	4.0372E-8
2	43	3.9459E-8	3.7421E-8	6.7167E-8	5.6558E-8	5.9588E-8	8.3516E-8	4.1778E-8
2	50	3.7742E-8	6.2535E-8	6.0402E-8	3.6465E-8	9.2708E-9	5.3740E-9	0.0000E+0
3	1	1.2250E-1	2.1440E-1	2.6018E-1	2.1263E-1	1.0472E-1	3.6772E-2	9.8109E-3

.
.
.

Figure 6.5. Synthetic window output from sample run.

6.6 UNFORMATTED RESPONSE FILE

The user can create an unformatted response file containing the window and response matrices using a non-ASCII format. Once this file is created it can be used to provide the window vectors and response matrix to FERD-PC at a substantially faster rate than using the standard ASCII response file. This procedure is highly recommended if one is going to be unfolding several sets of data using the same response each time. This option is selected using the second control parameter B in block 1 of the response data file. The file name used to open the unformatted file is listed as control parameter Q in the response file. The options to create and use an unformatted response file can not be selected at the same time or a message will be generated indicating a control parameter error and the program will be terminated. The unformatted response file only replaces the need for blocks (5) through (9) of the response data file and does not eliminate the need for the file, RESPONSE.DAT, which still provides the control parameters and binning information for FERD-PC. Further information on the response data file is provided in Chapter 4.

6.7 TEMPORARY SCRATCH FILE

At the beginning of FERD-PC, a temporary scratch file is created in the default directory. This scratch file is used to hold the response matrix during the use of subroutine GINV, where the response matrix is inverted and the original array is overwritten. It also holds the window matrix which is used during each loop to calculate the window estimation. While creating a scratch file is not necessary, and can be eliminated if the computer has the additional memory, it has proved useful. After inverting the matrix, the disk access light will occasionally go on indicating that program is still running. If the program ends successfully with a beep and the message, 'UNFOLDING COMPLETE!!!', the scratch file is automatically deleted from the disk. If the program terminates early through a control break or an error, this scratch file will not be deleted and will appear in the default

directory with a numerical name, such as 'ZZ040688'. If this file is created, the user may delete this file since it will not be used by FERD-PC in later runs.

REFERENCES

1. B. W. Rust, D. T. Ingersoll, and W. R. Burrus, *A User's Manual for the FERDO and FERD Unfolding Codes*, ORNL/TM-8720, 1983.

2. H. Kendrick and S. M. Sperling, *An Introduction to the Principles and Use of the FERDOR Unfolding Code*, GA-9882, 1970.

3. V. D. Bogert and W. R. Burrus, "The SLOP code for unfolding of instrument response," *Neutron Physics Div. Annual Progress Report for the Period Ending Sept. 1, 1962*, ORNL-3360.

4. R. H. Johnson, *A User's Manual for COOLC and FORIST*, PNE-75-107, 1975.

APPENDIX A
PLOTFERD

A.1 DESCRIPTION

PLOTFERD is a utility program written with Microsoft's Quickbasic compiler to display graphically FERD-PC input and output. With this program the user can plot FERD-PC files to the screen, compare and overlap against other files, and output the display to a HP 7475A or equivalent plotter. PLOTFERD supports enhanced color graphics to include CGA and monochrome modes. It has the capability to overlap up to six plots, each in a different color, with the ability to smooth, normalize, and subtract from each plot. These and other functions of PLOTFERD will be described in detail in this appendix.

A.2 SYSTEM REQUIREMENTS

PLOTFERD has been complied and tested using Microsoft's Quickbasic compiler version 3.0. Modifications to this program will require a compatible compiler. At a minimum the computer system must have the following hardware components:

* An IBM PC or PC-compatible that runs DOS version 2.0 or later.
* An enhanced graphics (EGA) or color graphics (CGA) adapter.
* Any display monitor compatible with the above graphic adapters.
* A math co-processor (8087/80287/80387)
* A minimum of 320K (kilobytes) of available user memory.
* A HP 7475A Graphics Plotter is optional for graphics output.

The software will automatically determine the type of graphics adapter and display monitor installed, thus no initial program setup is required. The optional HP 7475A graphics plotter requires a serial port which must be set to a baud rate of 2400, using COM PORT #1, with 8 data bits, and no parity check. Plotters normally require a special serial port cable configured to the specifications of the plotter.

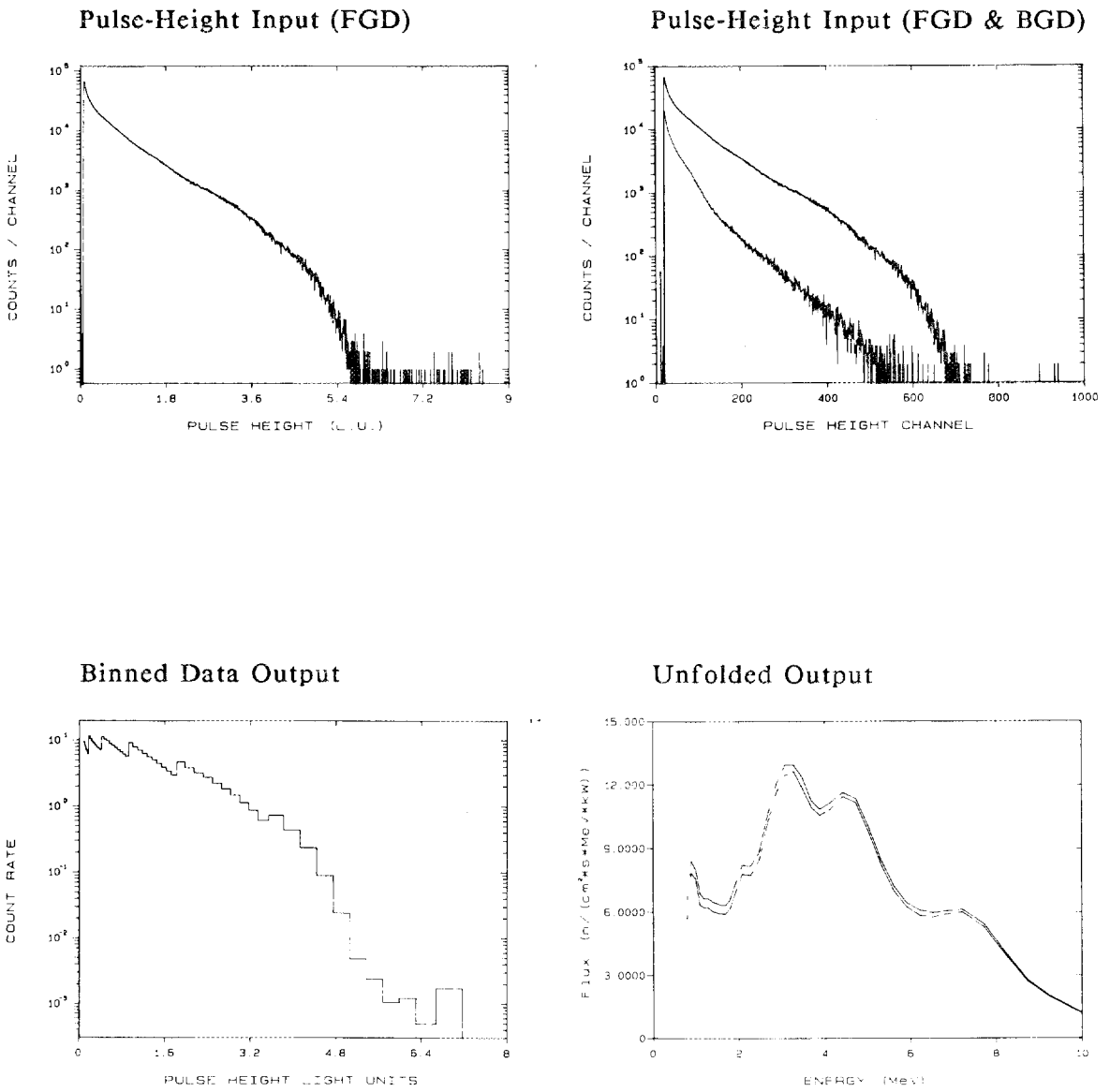

Figure A.1. Plot output from sample run.

This requirement can be obtained in the user manual for the specific plotter.

A.3 CHOOSING THE TYPE OF PLOT

The user has the option to plot FERD-PC input or output, with the type of format selected at the beginning of the program. After running PLOTFERD the screen will display the input options. Using the up and down arrow keys the user may select the type of file he wishes to plot. Hitting the enter key will accept the highlighted choice. The user will then be prompted for the file and path name. To exit the program from this menu the escape (ESC) key should be pressed. Each file is assumed to use the exact format as illustrated in each sample file of this manual. The format identifier used by FERD-PC for some input files is not used by PLOTFERD, thus any deviation from the sample files will require modification to this plotting program. Each type of file listed on the initial menu is described in detail below.

UNFOLDED OUTPUT: This is the main output file that is always generated by FERD-PC. If this option is selected, the user will be prompted to enter the file and path name of the unfolded data file. If the program is unable to locate the file, it will ask for the filename again. If the file exists, the program will open and read the data assuming the format is the same as shown in Figure 6.1. Any other format may cause the program to terminate or result in an incorrect plot. The program uses the first section of this file and does not use the integrated output. An example of plotting the main output file for the sample run is illustrated in Figure A.1.

CHANNEL PULSE-HEIGHT DATA: This selection is used to plot a pulse-height data file. If selected, the user will be prompted to enter the file and path name of the pulse-height file. This file must use the same format as is illustrated in Figure 5.1. After the file is opened the user will be asked if the data is to be

plotted as a function of light units. If (Y)es is entered here the user will be asked to enter calibration data using the same procedure as in the binning process for FERD-PC. If the user chooses not to plot it as a function of light units, the data will be displayed as a function of pulse-height channel only. Examples of both types of display are illustrated in Figure A.1.

BINNED PULSE-HEIGHT DATA FILE: This option will plot a binned data output file from FERD-PC. It must be in the same format as the binned data file listed in Figure 6.2. It is initially plotted as a histogram since each segment corresponds to count rate integrated over the bin width. An example of a binned data plot is illustrated in Figure A.1.

WINDOW DATA FILE: This option displays the window data listed in the response data file or the synthetic window file generated by FERD-PC. If selected, the user will be prompted for the file name and then asked:

DO YOU WANT TO PLOT THE WINDOW (1=FWHM or 2=MATRIX) ?

Entering 1 will plot block 7, which contains the FWHM resolutions of the window matrix. Entering 2 will plot a specific energy window from block 8. If the matrix is chosen, the user will be asked which window response is to be plotted with the question:

PLOT WINDOWS AT WHICH ENERGY (0.81 to 23.47) ? 15.0

This is entered as an energy value with the program selecting the nearest window response which corresponds to the same energy. The program will then display the following:

LOADING DATA ... WINDOW NUMBER 10 = 14.79 MeV

This informs the user which window number is being plotted along

with the corresponding energy. Up to six windows may be plotted together from the same or a different file. The energy and label for each window will be displayed at the top the plot. An example of both types of window plots is illustrated in Figure A.2.

RESPONSE MATRIX: This option displays the response matrix listed in the response data file. This must be an ASCII formatted file and cannot be used with an unformatted response file. The format is assumed to be the same as illustrated in Figure 4.1. After entering the file name, the user will be asked which energy response is to be plotted with the question:

PLOT RESPONSE AT WHICH ENERGY (0.81 to 23.47 MeV) ? 15.0

The program will select the nearest response corresponding to the entered energy then ask the user:

PLOT RESPONSE (1=PER LIGHT UNIT, 2=ABSOLUTE) ? 1

Entering 1 will plot the response with each interval divided by the pulse-height binning width. Entering 2 will plot the response exactly as listed in the block 9 of the file. Examples of the two types of plot are shown in Figure A.2.

PUNCH TYPE FILE: This option will plot the punch style file that FERD-PC is capable of outputting. An example of this file format is given in Figure 6.4.

A.4 FUNCTION KEYS

After the first file is loaded into memory, it will be graphically displayed onto the screen along with several function keys at the bottom of the display. At this point the user has several options available which can be selected by pressing the appropriate function key. Since there are as many as 18 functions available, not all

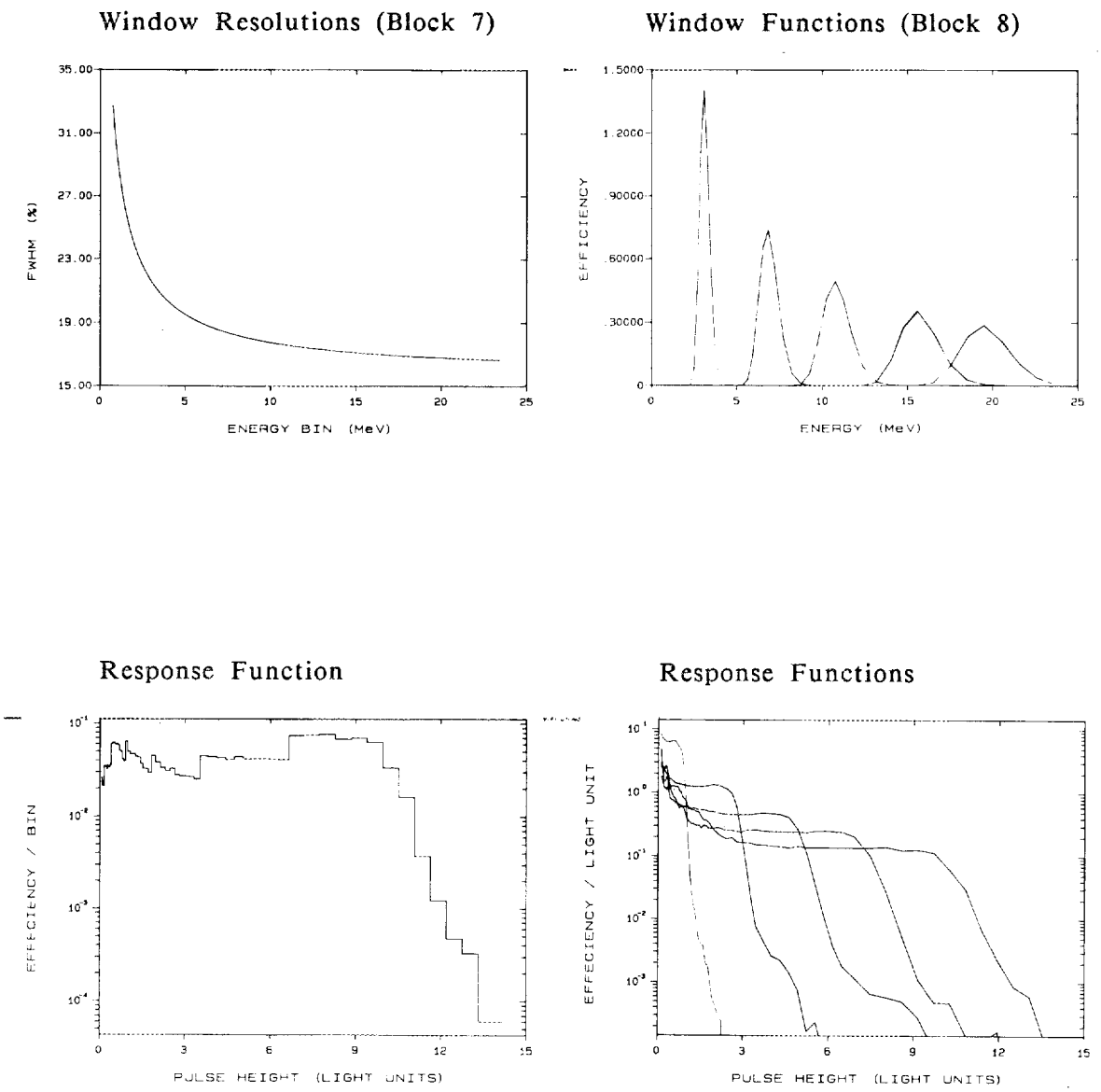

Figure A.2. Plot output from PLOTFERD using response data file.

options can be displayed at the same time. By pressing the space bar the user can toggle between function key menus or include a blank line. The blank line may be useful to avoid printing out the function keys when using a screen dump. This section will provide a description of each function key and its use.

F1 = LOAD: The F1 key enables the user to overlap and compare similar files. If selected, PLOTFERD will prompt the user for a file name and any other data as was required for the first plot. It will only plot a similar type file using the same format as the first loaded file. Each new plot will be assigned a different color if using a color display. The scale will remain the same until changed by the user, thus secondary files consisting of data outside of the scaled display may not show up until the plot is re-scaled by the user.

F2 = SAVE: The F2 key will save a plot to the disk. This may be used to save a file after subtracting a background, normalizing, or smoothing. If selected, the user will be prompted for a file and path name. If the file name already exists, it will be over written. The file is saved in a special format that can be reloaded under any other format.

F3 = Y-AXIS: The F3 key allows the user to re-scale the Y axis. The user will be prompted for the minimum and maximum scale with the default given in parenthesis. To accept the default, the user can press the enter key without entering any numbers.

F4 = X-AXIS: The F4 key allows the user to re-scale the X axis. The user will be prompted for the minimum and maximum scale with the default given in parenthesis. To accept the default, the user can press the enter key without entering any numbers.

F5 = LIN-Y / LOG-Y: The F5 key will re-scale the display using a linear or log scale for the Y axis. If the current plot is using

a vertical log scale, it will re-plot it using a linear scale. If the current plot is using a linear vertical scale, it will re-plot it using a log scale.

F6 = LIN-X / LOG-X: The F6 key will re-scale the display using a linear or log scale for the X axis. If the current plot is using a horizontal log scale, it will re-plot it using a linear scale. If the current plot is using a linear horizontal scale, it will re-plot it using a log scale.

F7 = PRINT: The F7 key enables the user to output all data points of a single plot to the screen or printer. If selected, the user will be prompted for which output device is to be used. While data is being printed, the user can temporarily halt the output by pressing the space bar and then resume the output by hitting any key. The output can be terminated at any time by pressing the escape (ESC) key.

F8 = SMOOTH: The F8 key provides a smoothing routine. This option enables the user to smooth the displayed plot using a five point averaging process.

F10 = CLR: The F10 key will clear the display and restart the program from the beginning. This enables the user to clear all plots and switch to another type of format or input.

ESC = EXIT: The ESC key is used to terminate and exit from PLOTFERD. This key is only valid when displaying a plot and no other function keys are being used. The ESC key is also used to return PLOTFERD to the display when some function keys are in use.

^F1 = CURSOR: Control F1 will display a cursor which can be moved along each data point by pressing the right or left arrow keys. The coordinates of each axis are displayed in the upper right portion of the screen. Pressing the control key with the

appropriate arrow key will move the cursor in increments of ten.

^F2 = NORM-Y: Control F2 is used to normalize a plot along the vertical or Y axis. The user will be prompted for a number which will be used to multiply each coordinate along the vertical or Y axis. The plot will then be re-plotted using its new data points.

^F3 = NORM-X: Control F3 is used to normalize a plot along the horizontal or X axis. The user will be prompted for a number which will be used to multiply each data point along the X axis. The plot will then be re-plotted using its new data points.

^F4 = SUM: Control F4 is used to sum every data point along the X axis. The user will be prompted for the plot number when multiple plots are currently displayed. The sum is then displayed at the top of the screen along with the plot number.

^F5 = AVER: Control F5 is used to average the plot over a given number of data points. The user will be prompted for the plot and number of points to be averaged. The chosen plot will then be summed and averaged over each interval determined by the number of points entered.

^F6 = PLOT: Control F6 is used to output all displayed plots to a HP 7475A or equivalent plotter. The user will be prompted for a X and Y axis label and a title, which is displayed at the top of the plot. This option uses Com Port #1 with the plotter set to a baud rate of 2400, with 8 data bits, and no parity check.

^F7 = SYMBOL: Control F7 is used to defined the type of line displaying each plot. With this option, the user can selectively display each plot as a line, histogram, or a line with each data point marked with a square. This may be useful in identifying plots that closely overlap each other.

^F8 = SUB: Control F8 is used to subtract one plot from another plot. The user will be prompted first for the plot that is to be subtracted. The plot is selected from one of the displayed numbers corresponding to the order in which all plots were loaded. The user will then be prompted for a second plot number. The first plot will then be subtracted from the second plot. The plot that is subtracted will be cleared from the screen. Any negative values which result from the subtraction will be reset to a value of zero. This option can only be used on multiple plots with the same X axis coordinates. This option may also be used to selectively clear any plot from the screen. To do this, the user selects the same plot number for both inputs. The order and color of some of the plots may change when using this option.

APPENDIX B
FERD-PC FORTRAN CODE

```
C     ================================================================
C     =                                                              =
C     =                    FERD-PC UNFOLDING CODE                    =
C     =                                                              =
C     =                    VERSION - MAY 11, 1988                    =
C     =                                                              =
C     ================================================================

C.... This program was written for MICROSOFT's FORTRAN compiler version 4.01

C.... This program requires the DOS config.sys file to include FILES = 10

C.... This program requires a external response file named RESPONSE.DAT

C-----------------------------------------------------------------------

C.... MAIN PROGRAM

      DIMENSION D(15000),ID(1),DPC(100),ICNTRL(25),RLW(50),RHI(50)
      EQUIVALENCE (D(1),ID(1))
      CHARACTER*30 RFILE,BFILE,LFILE,PFILE,TFILE,BNDAT1,BNDAT2,SFILE
      CHARACTER COM(25)*68,COMM*80,TITLE*80,QUES*1
      CHARACTER FMTB*20,FMT*20,FORM*1,NBLK*3,TF(700)*80
      COMMON /PPU1/ TITLE,PFILE,TFILE
      COMMON /PPU2/ N7,N10,NPW
      COMMON /BINN/ PC(100),R(100),DR(100),N2
      COMMON /UNIT/ N3,N8,N4,N9,N11

      DATA RFILE /'RESPONSE.DAT'/
      LTH = 15000
      IFL = 0

C.... INPUT/OUTPUT UNITS

      N1  = 1
      N2  = 2
      N3  = 3
      N4  = 4
      N7  = 7
      N8  = 8
      N9  = 9
      N10 = 10
      N11 = 11
      N12 = 12
```

```
C.... OPEN RESPONSE FILE (UNIT=N2) AND CHECK FOR BLOCK (1)

      WRITE (*,*)
      WRITE (*,*) 'PROGRAM FERD-PC'
      WRITE (*,*)
 100  OPEN (UNIT=N2,FILE=RFILE,STATUS='OLD',ERR=110)
      ICHAN = 0
      GO TO 180
 110  WRITE (*,111)
 111  FORMAT(/' Response file not found ...^G')
 150  WRITE (*,151)
 151  FORMAT(' Enter new file name of response data ..... ? ',\)
      READ (*,'(30A)') RFILE
      IF (RFILE.EQ.' ') GOTO 150
      WRITE (*,*)
      IFL = 1
      GO TO 100
 180  DO 181 I=1,10
      READ (N2,'(A3)',ERR=185) NBLK
      IF (NBLK.EQ.'(1)') GOTO 191
 181  CONTINUE
 185  WRITE (*,190) RFILE
 190  FORMAT(/' Did not find block (1) in - ^G',30A)
      GOTO 150

C.... READ CONTROL PARAMETERS AND COMMENTS FROM RESPONSE FILE

 191  IBI = I
      READ (N2,*)
      DO 192 I=1,14
 192  READ (N2,'(6X,I2,11X,60A)')  ICNTRL(I),COM(I)
      READ (N2,'(6X,F5.3,8X,60A)') TAU,COM(15)
      READ (N2,'(6X,F5.3,8X,60A)') DNM,COM(16)
      READ (N2,'(6X,A12,1X,60A)')  BFILE,COM(17)
      READ (N2,'(6X,F5.3,8X,60A)') RES,COM(18)
      IF (IFL.EQ.1) GOTO 250
      IF (ICNTRL(1).EQ.0) GOTO 250
      CLOSE (UNIT=N2)
      GOTO 150

C.... DISPLAY OR CHANGE CONTROL PARAMETERS IN RESPONSE FILE

  250 WRITE (*,251)
  251 FORMAT(' Display or Modifiy Control Parameters (Y/N)  ? ',\)
      READ (*,'(A1)') QUES
      IF (QUES.EQ.'N'.OR.QUES.EQ.'n') GOTO 299
      IF (QUES.EQ.'Y'.OR.QUES.EQ.'y') GOTO 252
      IF (QUES.EQ.' ') GOTO 299
      GOTO 250
```

```
252 WRITE (*,'(/)')
    WRITE (*,*) '       Parameter    Desciption'
    WRITE (*,*) '       -----------  ----------------------',
   +'-----------------------------------'
    DO 253 I=1,11
    K = I+64
253 WRITE(*,'(1H ,A1,A1,4X,I1,12X,A60)') CHAR(K),')',ICNTRL(I),COM(I)
    WRITE(*,'(1H ,A1,A1,4X,F5.3,8X,A60)') CHAR(79),')',TAU,COM(15)
    WRITE(*,'(1H ,A1,A1,4X,F5.3,8X,A60)') CHAR(80),')',DNM,COM(16)
    WRITE(*,'(1H ,A1,A1,4X,A12,1X,A60)') CHAR(81),')',BFILE,COM(17)
    WRITE(*,'(1H ,A1,A1,4X,F5.3,8X,A60)') CHAR(82),')',RES,COM(18)
    IF (ICHAN.EQ.1) WRITE (*,*)
254 WRITE (*,255)
255 FORMAT (/,' Enter corresponding letter to change parameter',
   +        /,' Then hit Enter (blank = no more changes) ... ? ',\)
    READ (*,'(A1)',ERR=254) QUES
    IF (QUES.EQ.' ') GOTO 257
    I = ICHAR(QUES)
    IF (I.LT.65) GOTO 254
    IF (I.GT.96) I = I-96
    IF (I.GT.64) I = I-64

259 IF (I.GT.11.AND.I.LT.15) GOTO 278
    ICHAN = 1
    IF (I.GT.11) GOTO 261
    ITMP = ICNTRL(I)
    IF (ITMP.EQ.0) ICNTRL(I) = 1
    IF (ITMP.EQ.1) ICNTRL(I) = 0
    IF (ICNTRL(5).EQ.1.AND.I.EQ.5) ICNTRL(4) = 1
    IF (ICNTRL(4).EQ.0.AND.I.EQ.4) ICNTRL(5) = 0
    IF (ICNTRL(2).EQ.1.AND.ICNTRL(3).EQ.1) GOTO 300
    IF (I.NE.1.OR.ICNTRL(I).NE.1) GOTO 278
289 WRITE (*,279)
279 FORMAT (/' Switch to new response file immediately (Y/N)? ',\)
    READ (*,'(A1)') QUES
    IF (QUES.EQ.'N'.OR.QUES.EQ.'n') GOTO 278
    IF (QUES.EQ.'Y'.OR.QUES.EQ.'y') GOTO 280
    GOTO 289
280 CLOSE (UNIT=N2)
    WRITE (*,'(/)')
    GOTO 150
300 ICNTRL(I) = 0
    WRITE (*,301)
301 FORMAT(/' Can not create and use unformatted response file at'
   1,' same time ...^G')
    GOTO 254
278 WRITE (*,290)
290 FORMAT (//,' FERD-PC Control Parameters',/)
    GOTO 252
261 IF (I.GT.15) GOTO 263
    WRITE (*,262)
```

```
  262 FORMAT (' Enter new value for TAU ..................... ? ',\)
      READ (*,*) TAU
      WRITE (*,290)
      GOTO 252
  263 IF (I.GT.16) GOTO 265
      WRITE (*,264)
  264 FORMAT (' Enter new detector normalization ........... ? ',\)
      READ (*,*) DNM
      WRITE (*,290)
      GOTO 252
  265 IF (I.GT.17) GOTO 267
      WRITE (*,266)
  266 FORMAT (' Enter new file name for unformated file .... ? ',\)
      READ (*,'(30A)') BFILE
      IF (BFILE(1:1).EQ.' ') GOTO 265
      WRITE (*,290)
      GOTO 252
  267 IF (I.GT.18) GOTO 257
      WRITE (*,268)
  268 FORMAT (' Enter new normalization parameter for blk 7  ? ',\)
      READ (*,*) RES
      WRITE (*,290)
      GOTO 252

  257 IF (ICHAN.EQ.0) GOTO 299
      WRITE (*,*)
  269 WRITE (*,270)
  270 FORMAT (' Make these changes permanent (Y/N) ......... ? ',\)
      READ (*,'(A1)') QUES
      IF (QUES.EQ.'N'.OR.QUES.EQ.'n') GOTO 299
      IF (QUES.EQ.'Y'.OR.QUES.EQ.'y') GOTO 271
      GOTO 269

C.... WRITE CHANGES TO RESPONSE FILE

  271 REWIND (N2)
      WRITE (*,282)
      IRZ = 1
  282 FORMAT (/,' Please wait ...')
      J=0
  294 J = J + 1
      IF (J.LE.700) GOTO 287
      WRITE (*,285) 43
  285 FORMAT (1A1,'                          ')
      WRITE (*,288)
  288 FORMAT (/' Response file too large to make permanent changes...^G')
      REWIND (N2)
      GOTO 299
  287 READ (N2,'(A78)',ERR=283) TF(J)
      GOTO 294
  283 REWIND (N2)
      DO 295 I=1,IBI+1
```

```
  295 WRITE (N2,'(A78)') TF(I)
      DO 272 I=1,11
      K = I+64
  272 WRITE (N2,'(1X,2A1,3X,I1,12X,A59)')  CHAR(K),')',ICNTRL(I),COM(I)
      DO 273 I=12,14
      K = I+64
  273 WRITE (N2,'(1X,2A1,3X,I2,11X,A59)')  CHAR(K),')',ICNTRL(I),COM(I)
      WRITE (N2,'(1X,A2,3X,F5.3,8X,A59)') 'O)',TAU,COM(15)
      WRITE (N2,'(1X,A2,3X,F5.3,8X,A59)') 'P)',DNM,COM(16)
      WRITE (N2,'(1X,A2,3X,A12,1X,A59)')  'Q)',BFILE,COM(17)
      WRITE (N2,'(1X,A2,3X,F5.3,8X,A59)') 'R)',RES,COM(18)
      DO 274 I=IBI+20,J-1
  274 WRITE (N2,'(A78)') TF(I)
      REWIND (N2)
      WRITE (*,286) 43
  286 FORMAT (1A1,\)

  299 IRF = ICNTRL(1)
      NBF = ICNTRL(2)
      NBM = ICNTRL(3)
      NLG = ICNTRL(4)
      NRR = ICNTRL(5)
      NPW = ICNTRL(6)
      NBN = ICNTRL(7)
      NBT = ICNTRL(8)
      NPC = ICNTRL(9)
      NSW = ICNTRL(10)
      NFW = ICNTRL(11)
      NRP = ICNTRL(12)
      NCL = ICNTRL(13)
      NRW = ICNTRL(14)
      IF (NLG.EQ.0)  N8 = 0

C.... CHECK UNFORMATTED RESPONSE FILE IF FLAG IS SET

      IF (NBM.EQ.0) GOTO 400
      OPEN(UNIT=N9,FILE=BFILE,ERR=350,STATUS='OLD',FORM='UNFORMATTED')
      GOTO 400
  350 WRITE (*,351) BFILE,RFILE
  351 FORMAT(/' Error reading unformatted file --------------- ',A30/
     +' Will use current ASCII response file --------- ^G',A30/)
      NBM = 0

C.... ENTER TITLE NAME FOR THIS RUN

 400  IF (IRZ.EQ.0) WRITE (*,402)
 402  FORMAT (/,\)
      IF (NBN.EQ.1) GOTO 500
      WRITE (*,401)
 401  FORMAT('Enter a title for this run ................. ? ',\)
      READ (*,'(80A)') TITLE
```

```
C.... NAME INPUT/OUTPUT FILES

  500 IF (NBN.EQ.0) GO TO 620
  600 WRITE (*,601)
  601 FORMAT('Enter file name of binned data input ....... ? ',\)
      READ (*,'(A30)') BNDAT1
      OPEN (UNIT=N1,FILE=BNDAT1,ERR=750,STATUS='OLD')
      GO TO 620
  750 WRITE (*,*)
      WRITE (*,*) 'File not Found ...^G '
      WRITE (*,751)
  751 FORMAT (1H ,\)
      GO TO 600
  620 IF (N8.EQ.0) GOTO 900
      WRITE (*,621)
  621 FORMAT(' Enter file name for log output ............. ? ',\)
      READ (*,'(30A)') LFILE
      IF (LFILE.EQ.' ') GOTO 620
  900 IF (NBT.EQ.0) GO TO 950
  910 WRITE (*,911)
  911 FORMAT(' Enter file name for binned data output ..... ? ',\)
      READ (*,'(30A)') BNDAT2
      IF (BNDAT2.EQ.' ') GOTO 910
  950 IF (NPC.EQ.0) GOTO 952
      WRITE (*,951)
  951 FORMAT(' Enter file name for punch output ........... ? ',\)
      READ (*,'(30A)') PFILE
      IF (PFILE.EQ.' ') GOTO 950
  952 IF (NSW.EQ.0) GOTO 960
      WRITE (*,953)
  953 FORMAT(' Enter file name for synthetic window output. ? ',\)
      READ (*,'(30A)') SFILE
      IF (SFILE.EQ.' ') GOTO 952
  960 WRITE (*,961)
  961 FORMAT(' Enter file name for unfolded output ........ ? ',\)
      READ (*,'(30A)') TFILE
      IF (TFILE.EQ.' ') GOTO 960
      OPEN(UNIT=N4,STATUS='SCRATCH',FORM='UNFORMATTED')

      IF (NBN.EQ.0) GOTO 965
      WRITE (*,5001)
 5001 FORMAT(/,' Unfolding data ...')

C.... READ PREVIOUSLY BINNED DATA FILE IF FLAG IS SET (UNIT=N1)

  965 IF (NBN.EQ.0) GO TO 966
      CALL PREBIN(N1,TITLE,NRP,PC,R,DR)
      CLOSE(UNIT=N1)

C.... OPEN LOG FILE IF FLAG IF SET (UNIT=N8)
```

```
  966 IF (N8.EQ.0) GOTO 1110
      OPEN (UNIT=N8,FILE=LFILE,ERR=970,STATUS='OLD')
      CLOSE (N8,STATUS='DELETE')
  970 OPEN (UNIT=N8,FILE=LFILE,STATUS='NEW')
      WRITE (N8,'(79A)') TITLE
      WRITE (N8,980) RFILE
  980 FORMAT(//'CONTROL PARAMETERS FROM RESPONSE FILE - ',30A)
      WRITE (N8,*)
      DO 982 I=1,11
  982 WRITE (N8,'(6X,I1,12X,A60)')  ICNTRL(I),COM(I)
      DO 983 I=12,14
  983 WRITE (N8,'(6X,I2,11X,A60)')  ICNTRL(I),COM(I)
      WRITE (N8,'(6X,F5.3,8X,A60)') TAU,COM(15)
      WRITE (N8,'(6X,F5.3,8X,A60)') DNM,COM(16)
      WRITE (N8,'(6X,A12,1X,A60)')  BFILE,COM(17)
      WRITE (N8,'(6X,F5.3,8X,A60)') RES,COM(18)
      WRITE (N8,*)

C.... OPEN FILE FOR SYNTHETIC WINDOWS IF FLAG IS SET (UNIT=N11)

 1110 IF (NSW.EQ.0) GOTO 990
      OPEN (UNIT=N11,FILE=SFILE,ERR=1120,STATUS='OLD')
      CLOSE (N11,STATUS='DELETE')
 1120 OPEN (UNIT=N11,FILE=SFILE,STATUS='NEW')
      WRITE (N11,'(79A)') TITLE

C.... TRANSFER CONTROL PARAMETERS TO ARRAY ICNTRL

  990 IF (N8.EQ.0) NRR = 0
      DO 991 I=1,25
  991 ICNTRL(I)  = 0
      ICNTRL(1)  = NRP
      ICNTRL(2)  = NCL
      ICNTRL(3)  = NRW
      ICNTRL(5)  = NBF
      ICNTRL(6)  = NBM
      ICNTRL(23) = NPC
      ICNTRL(8)  = NRR
      ICNTRL(10) = NSW
      ICNTRL(11) = NFW
      ICNTRL(12) = N11

C.... READ INTEGRAL OUTPUT TABLE BLOCK (2) (UNIT=N2)

      CALL INTEV(N2,RFILE,NSI,RLW,RHI)
 1000 IF (NBN.EQ.1) GOTO 1100

C.... READ BINNING PARAMETERS (UNIT=N2)
      CALL PHS(N2,RFILE,NRP,PC,DPC)

C.... BIN RAW MCA COUNTS INTO PROPER PULSE HEIGHT BINS
      CALL BIN(N1,N12,N8,NPW,DNM,PC,DPC,R,DR,NRP)
```

```
C.... OUTPUT BINNED DATA IF FLAG IS SET (UNIT=N1)
 1100 IF (N8.GT.0) CALL BINLOG(NRP,PC,R,DR)
      IF (NBT.EQ.0) GO TO 1200
      CALL BINOUT(N1,BNDAT2,TITLE,NRP,PC,R,DR)

C.... GETTIM IS A MICROSFT FORTRAN LIBRARY SUBROUTINE
 1200 CALL GETTIM(IHR1,IMIN1,ISEC1,IHSEC1)

C.... MAIN UNFOLDING CODE
      CALL ARRAY(D,ID,LTH,ICNTRL,TAU,RFILE,BFILE,LELAB,LPLO,LPUP,RES)

C.... INTEGRATION ROUTINE
      CALL INTP(D(LELAB),D(LPLO),D(LPUP),ICNTRL(3),RLW,RHI,NSI)
      CALL GETTIM(IHR2,IMIN2,ISEC2,IHSEC2)

      WRITE (*,5999) 43
 5999 FORMAT(1A1,'UNFOLDING COMPLETE!!!^G',/)
      I1 = (IHR2-IHR1)*3600
      I2 = (IMIN2-IMIN1)*60
      I3 = (ISEC2-ISEC1) + I1 + I2
      WRITE (*,6000) I3
 6000 FORMAT (' RUN TIME =',I4,' SECONDS',/)
      IF (N8.GT.0) WRITE(N8,6001)
 6001 FORMAT(1H /1H ,'FERD UNFOLDING JOB COMPLETED!!'/)
      END

C-----------------------------------------------------------------------

      SUBROUTINE ARRAY(D,ID,LTH,ICNTRL,TAU,RFILE,BFILE,LELAB,LPLO,LPUP,
     +RES)

C.... ALLOCATES SPACE FOR ARRAYS AND CHECKS LENGTH

      DIMENSION D(1),ID(1),ICNTRL(25)
      COMMON /UNIT/ N3,N8,N4,N9,N11
      CHARACTER RFILE*30,BFILE*30
      LENGTH = LTH

      NR   = ICNTRL(1)
      NC   = ICNTRL(2)
      NW   = ICNTRL(3)
      NRC  = MAX0(NR,NC)
      NRCW = MAX0(NR,NC,NW)
      LBLAB = 1
      LB    = LBLAB + NR
      LS    = LB    + NR
      LUT   = LS    + NR
      LLLO  = LUT   + NR
      LLUP  = LLLO  + NR
      LVLO  = LLUP  + NR
      LVUP  = LVLO  + NR
      LFORT = LVUP  + NR
```

```
      LW    = LFORT + NR
      LQ    = LW    + NC
      LATMP = LQ    + NC
      LELAB = LATMP + NC
      LISTR = LELAB + NC
      LPLO  = LISTR + NC
      LPUP  = LPLO  + NW
      LPLAB = LPUP  + NW
      LWINW = LPLAB + NW
      LUTMP = LWINW + NW
      LALO  = LUTMP + NRCW*10
      LHT   = LALO  + NR*NC
      LXTRA = LHT   + NR*NC
      LNTH  = LXTRA + MAX0(0,NW*NC-2*NR*NC)
      LWDOS = LALO

      IF (N8.GT.0) WRITE(N8,*)
      IF (N8.GT.0) WRITE(N8,*) 'FERD MEMORY ALLOCATION:'
      IF (N8.GT.0) WRITE(N8,603) LENGTH,LNTH
  603 FORMAT(1H ,/I7,' LOCATIONS ALLOCATED',
     + /I7,' LOCATIONS REQUIRED'/)
      IF(LNTH.GT.LENGTH)  GO TO 99

      CALL CNTRL(D(LALO),D(LB),D(LS),D(LW),D(LHT),D(LQ),D(LUT),
     1 D(LATMP),D(LPLO),D(LPUP),D(LBLAB),D(LPLAB),D(LELAB),D(LWINW),
     2 ID(LISTR),NR,NC,NW,TAU,D(LWDOS),D(LUTMP),NRCW,
     3 ICNTRL,RFILE,BFILE,RES)

      RETURN

   99 WRITE(*,603) LENGTH,LNTH
      WRITE(*,604)
  604 FORMAT(' **** DIMENSION "D" NOT LARGE ENOUGH ****^G'/)
      STOP
      END

C-----------------------------------------------------------------------

      SUBROUTINE CNTRL (ALO,B,S,W,HT,Q,UT,ATEMP,PLO,PUP,BLAB,
     1 PLAB,ELAB,WINW,ISTAR,MR,MC,KR,TAW,WINDOS,UTMP,NRCW,
     2 ICNTRL,RFILE,BFILE,RES)

C.... MAIN CONTROL SUBROUTINE FOR  F E R D - P C

      REAL*4 MU
      DIMENSION ALO(MR,MC),UTMP(NRCW,10)
      DIMENSION ATEMP(MC),PLO(KR),PUP(KR)
      DIMENSION B(MR),S(MR),W(MC),HT(MR,MC),Q(MC),UT(MR)
      DIMENSION BLAB(MR),PLAB(KR),ELAB(MC),ISTAR(MC),WINW(KR)
      DIMENSION WINDOS(KR,MC),ICNTRL(25),FWHM(100),ERRR(100)
      DIMENSION IDENT(20),IAPLT(7),IWPLT(7),RSP(10),FLUX(100)
      CHARACTER RFILE*30,NBLK*3,BFILE*30
```

```
      COMMON /BINN/ PC(100),R(100),DR(100),N2
      COMMON /UNIT/ N3,N8,N4,N9,N11

      NR = MR
      NC = MC
      NW = KR
      NBITS = 22
      ICS = ICNTRL(10)

C.... SET VALUE OF MU FOR 99.95 PCT. CONFIDENCE IN Q-BOUNDS
      TWONM1 = 2*NR - 1
      MU = (1.0/SQRT(2.0))*(3.270 + SQRT(TWONM1) )
      IF (N8.GT.0) WRITE (N8,150) MU
  150 FORMAT (/' *****   MU  =',1P,E11.3,/)

C.... READ IN ENERGY BINS FOR RESPONSE & WINDOW MATRICIES FROM RESPONSE FILE

      IF(ICNTRL(6).EQ.1)  GO TO 300
      DO 151, I=1,100
      READ (N2,'(A3)',ERR=149) NBLK
  151 IF (NBLK.EQ.'(5)') GOTO 153
  149 WRITE (*,152) RFILE
  152 FORMAT(/' Did not find block (5) in - ^G',30A)
      STOP
  153 READ (N2,*)
      READ (N2,'(5X,7F10.0)') (ELAB(I), I=1,NC)

C.... READ IN ROW ENERGIES FOR WINDOW MATRIX FROM RESPONSE FILE

      DO 154, I=1,100
      READ (N2,'(A3)',ERR=149) NBLK
  154 IF (NBLK.EQ.'(6)') GOTO 156
      WRITE (*,155) RFILE
  155 FORMAT(/' Did not find block (6) in - ^G',30A)
      STOP
  156 READ (N2,*)
      READ (N2,'(5X,7F10.0)') (PLAB(I), I=1,NW)

C.... READ IN WINDOW FHHW(%) USED FOR GAUSSIAN CALCULATIONS

  300 IF(ICNTRL(11).EQ.0) GOTO 160
      DO 361, I=1,50
      READ (N2,'(A3)',ERR=361) NBLK
  361 IF (NBLK.EQ.'(7)') GOTO 363
      WRITE (*,362) RFILE
  362 FORMAT(/' Did not find block (7) in - ^G',30A)
      STOP
  363 READ (N2,*)
      READ (N2,'(5X,7F10.0)') (WINW(I), I=1,NC)
      DO 364 I=1,NC
  364 WINW(I) = WINW(I) * RES
      GOTO 22
```

```
C.... READ IN WINDOW MATRIX FROM RESPONSE FILE

  160 IF(ICNTRL(6).EQ.1) GOTO 22
      DO 161, I=1,50
      READ (N2,'(A3)',ERR=161) NBLK
  161 IF (NBLK.EQ.'(8)') GOTO 163
      WRITE (*,162) RFILE
  162 FORMAT(/' Did not find block (8) in - ^G',30A)
      STOP
  163 READ (N2,*)
  164 READ (N2,'(1X,2I3,1X,7F10.0)') IR,IC,(RSP(K),K=1,7)
      IF (IR+IC.LE.0) GOTO 22
      DO 165, I=1,7
  165 WINDOS(IR,IC+I-1) = RSP(I)
      GOTO 164

C.... READ UNFORMATTED WINDOW MATRIX FROM RESPONSE.BIN

   22 IF(ICNTRL(6).EQ.0) GOTO 200
      READ(N9) PLAB
      READ(N9) ELAB
      READ(N9) WINDOS

C.... CALCULATE GAUSSIAN WINDOW FUNCTIONS

  200 IF(ICNTRL(11).EQ.0) GOTO 23
      DO 20 I=1,NW
      CALL WREAD(W,NC,KR,I,ELAB,PLAB,WINW)
      DO 20 J=1,NC
   20 WINDOS(I,J) = W(J)

C.... WRITE WINDOW FUNCTIONS TO UNFORMATTED RESPONSE FILE

   23 IF(ICNTRL(5).EQ.0)  GO TO 24
      OPEN(UNIT=N9,FILE=BFILE,ERR=231,STATUS='OLD',FORM='UNFORMATTED')
      CLOSE (UNIT=N9,STATUS='DELETE')
  231 OPEN(UNIT=N9,FILE=BFILE,STATUS='NEW',FORM='UNFORMATTED')
      WRITE(N9) PLAB
      WRITE(N9) ELAB
      WRITE(N9) WINDOS

   24 IF (ICNTRL(8).EQ.0) GOTO 27
      WRITE(N8,*)
      WRITE(N8,*) 'WINDOW MATRIX:'
      WRITE(N8,*)
      DO 70 I=1,NW
      DO 70 J=1,NC,7
      SUM = 0
      DO 68 K=0,6
   68 SUM = SUM + WINDOS(I,J+K)
      IF (SUM.EQ.0) GOTO 70
      WRITE (N8,'(1X,2I3,1X,1P,7E10.4E1)') I,J, (WINDOS(I,J+K), K=0,6)
   70 CONTINUE
```

```
   27 DO 26 I=1,NW
      DO 25 J=1,NC
      W(J) = WINDOS(I,J)
   25 WINDOS(I,J) = 0
   26 WRITE(N4) W

C.... READ IN RESPONSE MATRIX FROM RESPONSE FILE

  180 IF(ICNTRL(6).EQ.1) GOTO 28
      DO 181, I=1,300
      READ (N2,'(A3)',ERR=181) NBLK
  181 IF (NBLK.EQ.'(9)') GOTO 183
      WRITE (*,182) RFILE
  182 FORMAT(/' Did not find block (9) in - ^G',30A)
      STOP
  183 READ (N2,*)
  184 READ (N2,'(2I3,8F9.0)',END=30) IR,IC,(RSP(K), K=1,8)
      IF (IR+IC.LE.0) GOTO 30
      DO 185, I=1,8
  185 ALO(IR,IC+I-1) = RSP(I)
      GOTO 184

C.... READ IN UNFORMATTED RESPONSE MATRIX FROM RESPONSE.BIN

   28 READ(N9) ALO
      CLOSE (UNIT=N9)

C.... WRITE RESPONSE MATRIX INTO UNFORMATTED FILE RESPONSE.BIN
   30 IF(ICNTRL(5).EQ.0) GOTO 13
      WRITE(N9) ALO
      CLOSE (UNIT=N9)

   13 IF (ICNTRL(8).EQ.0) GOTO 15
      WRITE(N8,*)
      WRITE(N8,*) 'RESPONSE MATRIX:'
      WRITE(N8,*)
      DO 80 I=1,NR
      DO 80 J=1,NC,8
      SUM = 0
      DO 78 K=0,7
   78 SUM = SUM + ALO(I,J+K)
      IF (SUM.EQ.0) GOTO 80
      WRITE (N8,'(2I3,1P,8E9.2)') I,J,(ALO(I,J+K), K=0,7)
   80 CONTINUE

   15 WRITE(N4) ALO
      BACKSPACE N4

C.... INSERT PREBINNED DATA
      DO 111, I=1,NR
      BLAB(I) = PC(I)
```

```
      B(I) = R(I)
  111 S(I) = DR(I)

C.... UNFOLD DATA
      CALL FERD (ALO,B,S,W,HT,Q,UT,UTMP,ATEMP,ISTAR,ELAB,PLAB,
     + MR,NR,NC,NW,NBITS,TAW,KR,MU,PLO,PUP,NRCW,CONIND,QTMAX,ICS)

C.... PRINT PUNCH DATA IF FLAG IS SET

      IF(ICNTRL(23).GT.0) CALL PUNCH(ELAB,PLO,PUP,NW)

      RETURN
      END

C-----------------------------------------------------------------------------

      SUBROUTINE FERD(ALO,B,S,W,HT,Q,UT,UTMP,ATEMP,ISTAR,ELAB,PLAB,
     1 MR,NR,NC,NW,NBITS,TAW,KR,MU,PLO,PUP,NRCW,CONIND,QTMAX,ICS)

      REAL*4 MU
      DIMENSION ALO(MR,NC),B(NR),S(NR),W(NC),HT(MR,NC),PLAB(NW),
     1 Q(NC),UT(NR),UTMP(NRCW,10),ATEMP(NC),ISTAR(NC),ELAB(NC),
     2 PLO(KR),PUP(KR)
      COMMON /UNIT/ N3,N8,N4,N9,N11

C.... SYNTHETIC WINDOW STORAGE ARRAY
      DIMENSION WIN(100,100),SYNWLO(100)
      NT1 = 100
      NT2 = 100

C.... CALCULATE Q VECTOR
      DO 20 J=1,NC
      ISTAR(J) = 0
      Q(J)=1.0E35
      DO 20 I=1,NR
      B(I) = AMAX1(B(I),0.0)
      IF(ALO(I,J))10,20,10
   10 QMIN=(B(I)+MU*S(I))/ALO(I,J)
      IF(Q(J)-QMIN)20,20,15
   15 Q(J)=QMIN
      ISTAR(J)=I
   20 CONTINUE

C.... ADDITION FOR ZERO COLUMNS IN MATRIX   5/7/80
      DO 22 J=1,NC
      IF(ISTAR(J).EQ.0.0) Q(J) = 0.0
   22 CONTINUE

C.... NORMALIZE MATRIX (USING AVERAGE MATRIX Q, S, AND UT FROM ABOVE)
      ENORM=0.0
      DO 40 I=1,NR
        DO 40 J=1,NC
```

```
          HT(I,J)= Q(J) * ALO(I,J) / S(I)
          ENORM=ENORM+HT(I,J)*HT(I,J)
   40 CONTINUE
      ENORM=SQRT(ENORM)
      CONIND=ENORM * 10.0 * 0.5**NBITS

C.... INCREASE TAW FOR LARGE CONDITION NUMBER
      TAW=AMAX1(TAW,CONIND)
      IF (N8.GT.0) WRITE(N8,905) CONIND

C.... CALCULATE TRANSFORMATION MATRIX
      CALL GINV(HT,MR,NR,NC,TAW/SQRT(FLOAT(NC)),ALO,ATEMP)
      READ(N4) ALO
      REWIND N4

C.... UNNORMALIZE TRANSFORMED MATRIX
      DO 50 I=1,NR
        DO 50 J=1,NC
          HT(I,J)=Q(J)*HT(I,J)/S(I)
   50 CONTINUE

C.... BEGIN WINDOW LOOP
      DO 200 K=1,NW
      READ(N4) W

C.... CALCULATE UT=WTH
      DO 60 I=1,NR
   60 UT(I)=0.0
      DO 100 J=1,NC
      IF(W(J))70,100,70
   70 DO 80 I=1,NR
      UT(I)=UT(I)+W(J)*HT(I,J)
   80 CONTINUE
  100 CONTINUE

C.... LOWER ESTIMATE FOR WINDOW
      CALL LOOP(W,B,S,Q,ALO,MR,NR,NC,PHILO,ERR,ISTAR,UT,UTMP(1,1),
     +UTMP(1,2),0,QUE,ICS,SYNWLO)
      UTMP(K,6)=ERR
      PLO(K) = PHILO
      QLO = QUE

C.... UPPER ESTIMATE FOR WINDOW
      DO 150 J=1,NC
  150 W(J)=-W(J)
      DO 160 I=1,NR
  160 UT(I)=-UT(I)
      CALL LOOP(W,B,S,Q,ALO,MR,NR,NC,PHIUP,ERR,ISTAR,UT,UTMP(1,1),
     +UTMP(1,2),1,QUE,ICS,SYNWLO)
      UTMP(K,6)=UTMP(K,6)+ERR
      PUP(K) = -PHIUP
      QUP = QUE
      UTMP(K,10)=100.0*(QUP+QLO)/(PHIUP-PHILO)
```

```
C.... OPTIONAL PLOT OF UPPER SYNTHETIC WINDOW
      IF (ICS.EQ.0) GOTO 200
      DO 165 IL=1,NC
  165 WIN(K,IL) = SYNWLO(IL)

  200 CONTINUE

C.... OPTIONAL OUTPUT OF SYNTHETIC WINDOW
      IF (ICS.EQ.1) CALL SYNWIN(NC,NW,ELAB,PLAB,WIN,NT1,NT2)

  700 IF (N8.GT.0) WRITE(N8,900)
      IF (N8.GT.0) WRITE(N8,902)
      QTMAX = 0
      DO 750 I=1,NW
      PAVG = (PUP(I)+PLO(I)) / 2.0
      IF (ABS(UTMP(I,10)).GT.QTMAX) QTMAX=ABS(UTMP(I,10))
  750 IF (N8.GT.0) WRITE(N8,901) I,ELAB(I),PLO(I),PUP(I),PAVG,
     +UTMP(I,6),UTMP(I,10)
  800 RETURN

  900 FORMAT(10X,6HENERGY,5X,3HPLO,9X,3HPUP,9X,4HPAVG,8X,3HERR,
     +9X,6HQSHOOT,/)
  902 FORMAT(10H          ,69(1H-))
  901 FORMAT(1H ,I5,1X,1P,E12.3,1P,5E12.4)
  905 FORMAT(/,' *****   CONDITION INDICATOR = ',E15.7,//)
      END

C-----------------------------------------------------------------------------

      SUBROUTINE LOOP(W,B,S,Q,ALO,MR,NR,NC,PHILO,ERR,ISTAR,UT,ULO,
     +ELO,ITU,QUE,ICS,SYNWLO)

C.... LOOP FOR SUBROUTINE FERD

      DIMENSION W(NC),B(NR),S(NR),Q(NC),ALO(MR,NC),ISTAR(NC),
     +UT(NC),ULO(NC),ELO(NC),SELO(100),SYNWLO(NC)
      COMMON /UNIT/ N3,N8,N4,N9,N11

      DO 100 I=1,NR
  100 ULO(I)=UT(I)

      IF (ITU.EQ.0) GOTO 125
      DO 110 J=1,NC
  110 ELO(J) = SELO(J)
      GOTO 210

  125 DO 200 J=1,NC
      UTA=0.0
      DO 180 I=1,NR
      IF (ULO(I)) 175,180,175
  175 UTA=UTA+ULO(I)*ALO(I,J)
  180 CONTINUE
```

```
      ELO(J)=W(J)-UTA
      SELO(J) = -ELO(J)
  200 CONTINUE

C.... FIND LARGEST NEGATIVE COMPONENT, EJMIN, OF ELO * Q, AND ITS INDEX JMIN
  210 EJMIN=0.0
      DO 240 J=1,NC
      EJQJ=ELO(J)*Q(J)
      IF(EJQJ)220,240,240
  220 IF(EJQJ-EJMIN)230,240,240
  230 EJMIN=EJQJ
      JMIN=J
  240 CONTINUE
      IS=ISTAR(JMIN)
      IF (EJMIN) 255,340,340

C.... REMOVE ISTAR ROW FROM WINDOW ERROR, ELO
  255 DO 270 J = 1,NC
  270 ELO(J) = ELO(J) + ULO(IS)*ALO(IS,J)

C.... COMPUTE NEW ULO(IS) TO MAKE ELO(JMIN) = 0.0
      IF (ELO(JMIN)) 275,290,280
  275 ULO(IS) = ELO(JMIN)/ALO(IS,JMIN)*1.001
      GOTO 290
  280 ULO(IS) = ELO(JMIN)/ALO(IS,JMIN)*0.999

C.... PUT NEW ISTAR ROW BACK INTO ELO
  290 DO 320 J = 1,NC
  320 ELO(J) = ELO(J) - ULO(IS)*ALO(IS,J)

C.... CONTINUE THIS PROCESS UNTIL THE WINDOW INEQUALITY IS SATISFIED
      GO TO 210

C.... STORE NEW SYNTHETIC WINDOW IF FLAG IS SET
  340 IF (ICS.EQ.0) GOTO 380
      DO 375 J=1,NC
      SUM1 = 0
      DO 370 I=1,NR
  370 IF (ULO(I).NE.0) SUM1 = SUM1 + ULO(I) * ALO(I,J)
      SYNWLO(J) = ABS(SUM1)
  375 IF (SYNWLO(J).LT.1.0E-9) SYNWLO(J) = 0.0

C.... CALCULATE PHI AND (U*S)**2
  380 ULOSQ=0.0
      PHILO=0.0
      DO 400 I=1,NR
      PHILO=PHILO+ULO(I)*B(I)
  400 ULOSQ=ULOSQ+(ULO(I)*S(I))**2
      ERR = SQRT(ULOSQ)
      PHILO=PHILO-ERR
      IF (N8.GT.0) QUE = QSHOOT(ELO,Q,NC)
      RETURN
      END
```

```
C-----------------------------------------------------------------------

      FUNCTION QSHOOT(D,Q,NC)
      DIMENSION D(NC),Q(NC)

C.... FUNCTION QSHOOT GIVES AN ESTIMATE OF ERROR DUE TO
C     QUADRATIC UNDERSHOOT OF THE SYNTHETIC WINDOW BETWEEN
C     THE COMPARISON POINTS .

      QSHOOT=0.0
      NN=NC-1
      DO 40 I=2,NN
      DER1=0.5*(D(I+1)-D(I-1))
      DER2=D(I-1)-2.0*D(I)+D(I+1)

C.... SECOND DERIVATIVE MUST BE POSITIVE FOR A MINIMUM TO OCCUR
      IF(DER2)40,40,10
   10 XMIN=-DER1/DER2

C.... DOES MINIMUM OCCUR WITHIN RANGE
      IF(ABS(XMIN)-1.0)20,40,40
   20 VALUE=D(I)-(DER1**2)/(2.0*DER2)

C.... HAS AN UNDERSHOOT OCCURRED I. E., IS MINIMUM NEGATIVE
      IF(VALUE)30,40,40
   30 QSHOOT=QSHOOT-VALUE*Q(I)
   40 CONTINUE
      QSHOOT=4.0/9.0*QSHOOT
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE WREAD(W,NC,NW,NTRY,ELAB,PLAB,PCTWID)

C.... ASSUMES THAT PLAB AND ELAB ARE IN UNITS OF  MEV

      DIMENSION W(1),ELAB(1),PLAB(1),PCTWID(1)

C.... THE FOLLOWING IS THE ORDINARY DEFINITION OF THE GAUSS FUNCTION
      GAUSS(X) = 0.3989422804014 * EXP(-0.5*X**2)

C.... TO ESCAPE FROM PC COMPILER WARNING
      NW=NW

      SIGMA = (.01 * PCTWID(NTRY) * PLAB(NTRY))/(2.0 * 1.17741002251)
      DO   10   I = 1,NC
           DIFOS = ABS(ELAB(I) - PLAB(NTRY))/SIGMA
           IF   (DIFOS - 5.5)   3,2,2
    2           W(I) = 0.0
                     GO TO 10
    3           W(I) = GAUSS(DIFOS)/SIGMA
```

```
   10 CONTINUE
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE GINV(A,MR,NR,NC,TAU,U,ATEMP)

C     UPON ENTRY   A = A MATRIX WITH NR ROWS AND NC COLUMNS
C     AFTER THE RETURN THE ORIGINAL A IS DESTROYED AND THE ARRAY A
C     CONTAINS THE TRANSPOSE OF  T   THE TRANSFORMATION MATRIX
C        (X = T*B) WHICH SOLVES THE PAIR OF EQUATIONS
C
C                  A*X = B    AND    TAU I*X = 0
C
C          IN THE LEAST SQUARE SENSE.
C
C     MR = 1ST DIMENSION NO. OF ARRAY A IN THE CALLING PROGRAMS
C     TAU IS A NONNEGATIVE CONSTANT WHICH CONTROLS THE ERROR
C     PROPIGATION OF THE TRANSFORMATION MATRIX.   IF TAU = 0.   THEN THE
C     RESULTING TRANSFORMATION MATRIX IS THE ORDINARY LEAST SQUARES
C     TRANSFORMATION MATRIX.   (THE INVERSE IF NR = NC)
C     ATEMP AND U ARE USED FOR WORKING SPACE BY THE ALGORITHM AND DO NOT
C     NECESSARILY CONTAIN ANY RELAVANT NUMBERS AT THE CONCLUSION.
C     ATEMP MUST BE DIMENSIONED AT LEAST NC BY THE CALLING PROGRAMS
C     U MUST BE DIMENSIONED AT LEAST NC*NC BY THE CALLING PROGRAMS
C     NO. OF MULTIPLICATIONS = NC**2 (5/2 NR + 2/3 NC)

      DIMENSION A(MR,NC),U(NC,NC),ATEMP(NC)
      TAUSQ = TAU**2

C.... PLACE UNIT MATRIX IN U
      DO  5  I = 1,NC
          DO  4  J = 1,NC
    4         U(I,J) = 0.0
    5     U(I,I) = 1.0

C     ORTHOGONALIZE COMBINED MATRIX (A ABOVE U) BY GRAMM-SCHMIDT-HILBERT
C        METHOD WITH FIRST NR ROWS WEIGHTED WITH 1  AND THE OTHER
C        NC ROWS WEIGHTED WITH 1/TAU.  THEN REORTHOGONALIZE TO
C        LESSEN ROUNDOFF ERROR.

      DO  20  I = 1,NC
          II = I - 1
          IF  (II)  2,11,2
    2     DO  10  LL = 1,2
              DO  10  J = 1,II
              DOT = 0.0
              DOT2 = 0.0
              DO  3  K = 1,NR
    3             DOT = A(K,I)*A(K,J) + DOT
              DO  6  K = 1,J
```

```
    6             DOT2 = U(K,I)*U(K,J) + DOT2
            DOT = DOT + DOT2*TAUSQ
            DO  8  K = 1,J
    8             U(K,I) = U(K,I) - DOT*U(K,J)
            DO  10  K = 1,NR
   10             A(K,I) = A(K,I) - DOT*A(K,J)

C.... NORMALIZE THE COLUMN  I OF THE COMBINED MATRIX
   11     DOT = 0.0
          DOT2 = 0.0
          DO  12  K = 1,NR
   12         DOT = A(K,I)*A(K,I) + DOT
          DO  14  K = 1,I
   14         DOT2 = U(K,I)*U(K,I) + DOT2
          DOT = DOT + DOT2*TAUSQ
          DOT = SQRT(DOT)
          DO 17  K = 1,I
   17         U(K,I) = U(K,I)/DOT
          DO  19  K = 1,NR
   19         A(K,I) = A(K,I)/DOT
   20 CONTINUE

C.... CALCULATION OF THE TRANSPOSE OF THE TRANSFORMATION
C.... MATRIX   T(TRANS.) = A * U(TR.)
      DO  50  I = 1,NR
          DO  45  J = 1,NC
              ATEMP(J) = 0.0
              DO  45  K = J,NC
                  ATEMP(J) = A(I,K)*U(J,K) + ATEMP(J)
   45         CONTINUE
          DO  50  J = 1,NC
              A(I,J) = ATEMP(J)
   50         CONTINUE
      RETURN
      END

C-----------------------------------------------------------------------------

      SUBROUTINE PUNCH(ELAB,PLO,PUP,NW)

C.... OUTPUTS PUNCH FILE DATA

      DIMENSION ELAB(NW),PLO(NW),PUP(NW)
      CHARACTER TITLE*80,PFILE*30,TFILE*30
      COMMON /PPU1/ TITLE,PFILE,TFILE
      COMMON /PPU2/ N7,N10,NPW

      OPEN (UNIT=N7,FILE=PFILE,ERR=952,STATUS='OLD')
      CLOSE(UNIT=N7,STATUS='DELETE')
  952 OPEN (UNIT=N7,FILE=PFILE,STATUS='NEW')
```

```
      WRITE(N7,'(80A)') TITLE
      WRITE(N7,'(1P,3E15.5)') (ELAB(I),PLO(I),PUP(I), I=NW,1,-1)
      CLOSE(UNIT=N7)
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE SYNWIN(NC,NW,ELAB,PLAB,WINDOS,NT1,NT2)

C.... OUTPUTS SYNTHETIC WINDOW DATA

      DIMENSION ELAB(NC),PCTWIN(100),PLAB(NW),WINDOS(NT1,NT2)
      COMMON /UNIT/ N3,N8,N4,N9,N11

      GAUS = 0.3989422804014

      WRITE (N11,140)
  140 FORMAT (/'(5) ENERGY BINS FOR SYNTHETIC WINDOW OUTPUT IN MeV',/)
      WRITE (N11,'(5X,1P,7E10.3)') (ELAB(J), J=1,NC)

      WRITE (N11,141)
  141 FORMAT (/'(6) ENERGY VECTORS FOR SYNTHETIC WINDOWS IN MeV',/)
      WRITE (N11,'(5X,1P,7E10.3)') (PLAB(J), J=1,NW)

      WRITE (N11,150)
  150 FORMAT (/'(7) SYNTHETIC WINDOWS IN GUASSIAN FWHM (%)',/)
      DO 170 J=1,NC
      IF (WINDOS(J,J).EQ.0.OR.ELAB(J).EQ.0) GOTO 170
      SIGMA = GAUS/WINDOS(J,J)
      PCTWIN(J) = SIGMA * (2.0 * 1.17741002251) / .01 / ELAB(J)
  170 CONTINUE
      WRITE (N11,'(3X,7F10.4)') (PCTWIN(K), K=1,NC)

      WRITE (N11,151)
  151 FORMAT (/'(8) SYNTHETIC WINDOW MATRIX',/)
      DO 70 I=1,NW
      DO 70 J=1,NC,7
      SUM = 0
      DO 68 K=0,6
   68 SUM = SUM + WINDOS(I,J+K)
      IF (SUM.EQ.0) GOTO 70
      WRITE (N11,'(1X,2I3,1X,1P,7E10.4E1)') I,J, (WINDOS(I,J+K), K=0,6)
   70 CONTINUE

      WRITE (N11,'(//)')
      CLOSE (UNIT=N11)
      RETURN
      END
```

```
C-----------------------------------------------------------------------

      SUBROUTINE PHS(N2,RFILE,NRP,PC,DPC)

C.... READ PULSE HEIGHT UNITS AND BINNING WIDTH FROM RESPONSE FILE

      DIMENSION PC(1),DPC(1)
      CHARACTER NBLK*3,RFILE*30

      DO 260 I=1,50
      READ (N2,'(A3)',ERR=260) NBLK
  260 IF (NBLK.EQ.'(3)') GOTO 270
      WRITE (*,261) RFILE
  261 FORMAT(/' Did not find block (3) in - ^G',30A)
      STOP
  270 READ (N2,*)
      READ (N2,'(5X,8F9.0)') (PC(I), I=1,NRP)

      DO 360 I=1,50
      READ (N2,'(A3)',ERR=360) NBLK
  360 IF (NBLK.EQ.'(4)') GOTO 370
      WRITE (*,361) RFILE
  361 FORMAT(/' Did not find block (4) in - ^G',30A)
      STOP
  370 READ (N2,*)
      READ (N2,'(5X,8F9.0)') (DPC(I), I=1,NRP)

      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE PREBIN(N1,TITLE,KR,PC,R,DR)

C.... READ IN PREVIOUSLY BINNED DATA FILE IF FLAG IS SET

      DIMENSION PC(1),R(1),DR(1)
      CHARACTER TITLE*80,FMTB*20

      READ (N1,'(79A)') TITLE
      READ (N1,'(9X,A20)') FMTB
      READ (N1,FMTB) (PC(I),R(I),DR(I), I=1,KR)
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE BINOUT(N1,BNDAT2,TITLE,KR,PC,R,DR)

C.... OUTPUT BINNED DATA IF FLAG IS SET
```

```
      DIMENSION PC(1),R(1),DR(1)
      CHARACTER BNDAT2*30,TITLE*80

 1200 OPEN (UNIT=N1,FILE=BNDAT2,ERR=1210,STATUS='OLD')
      CLOSE (N1,STATUS='DELETE')
 1210 OPEN (UNIT=N1,FILE=BNDAT2,STATUS='NEW')
      WRITE (N1,'(80A)') TITLE
      WRITE (N1,1211)
 1211 FORMAT ('FORMAT = (12X,3F15.0)')
      WRITE (N1,1255) (I,PC(I),R(I),DR(I), I=1,KR)
 1255 FORMAT (6X,I4,2X,1P,3E15.7)
      WRITE (N1,*) ' '
      CLOSE (UNIT=N1)
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE BINLOG(KR,PC,R,DR)

      DIMENSION PC(1),R(1),DR(1)
      COMMON /UNIT/ N3,N8,N4,N9,N11

      WRITE (N8,*) 'BINNED DATA:'
      WRITE (N8,*)
      WRITE (N8,1100)
      WRITE (N8,1101)
 1100 FORMAT('                LIGHT UNITS    COUNT RATE     ERROR')
 1101 FORMAT('                -------------  -------------  ------------')
      DO 1110 I = 1, KR
        WRITE (N8,1108) I, PC(I), R(I), DR(I)
 1108   FORMAT (6X,I4,2X,1P,3E15.7)
 1110 CONTINUE
      WRITE (N8,*)
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE INTEV(N2,RFILE,NSI,RLW,RHI)

C.... READ INTEGRAL OUTPUT TABLE FROM RESPONSE DATA FILE

      DIMENSION RLW(1),RHI(1)
      CHARACTER NBLK*3,RFILE*30

      DO 1510 I=1,50
      READ (N2,'(A3,A60)') NBLK
 1510 IF (NBLK.EQ.'(2)') GOTO 1535
      WRITE (*,1520) RFILE
 1520 FORMAT(/' Did not find block (2) in - ^G',30A)
      STOP
```

```
 1535 READ (N2,*)
      READ (N2,*)
      READ (N2,*)
      DO 1540 NSI=1,50
      READ (N2,'(6X,2F8.0)') RLW(NSI),RHI(NSI)
 1540 IF (RLW(NSI).EQ.0.AND.RHI(NSI).EQ.0) GOTO 1545
 1545 NSI = NSI - 1
      RETURN
      END

C-----------------------------------------------------------------------------

      SUBROUTINE BIN(N1,N12,N8,NR,XNOR4,PC,DPC,R,DR,KR)

C.... THIS IS THE MAIN SUBROUTINE FOR THE BINNING CODE USED TO
C     PREPARE RAW MCA DATA FOR USE IN FERD UNFOLDING CODE

      DOUBLE PRECISION DRL,DRH,RL,RH
      DOUBLE PRECISION RSUM(2048)
      DIMENSION NCOM(20),IA(2048),XL(2048)
      DIMENSION R(100),DR(100),PC(100),DPC(100),A(2048)
      DIMENSION XR(100),SXR(100),HXR(100),SHXR(100)
      CHARACTER NF01*30,NF02*30,COM*78,FMTC*20,QUES*1
      DATA IPN /0/
      NMAX = 2048

      IF (N8.GT.0) WRITE(N8,2900)
 2900 FORMAT(/' INPUT USING BINNING SUBROUTINE:'/)
      DO 10 I = 1, KR
      XR(I) = 0.0
      HXR(I) = 0.0
      SXR(I) = 0.0
      SHXR(I) = 0.0
      R(I)  = 0.0
   10 DR(I) = 0.0

C.... USER INPUT FROM KEYBOARD

      KNOR3 = 1
      IPN = IPN + 1
      IF (BNN.EQ.1) GOTO 660
      WRITE (*,*)
   71 WRITE(*,72)
   72 FORMAT(' Enter file name of foreground data ............... ? ',\)
      READ(*,'(30A)') NF01
      IF (NF01.EQ.' ') GOTO 71
      OPEN(UNIT=N1,FILE=NF01,ERR=2100,STATUS='OLD')
      IF (N8.GT.0) WRITE(N8,3000) NF01
 3000 FORMAT(' FOREGROUND FILE NAME ............... ',30A)
      GOTO 58
```

```
2100 WRITE (*,*)
     WRITE (*,*) 'File not found ... Try again'
     IF (KNOR3) 117,71,71

  58 WRITE(*,59)
  59 FORMAT(' Light units at calibration point ................. ? ',\)
     READ(*,*) EGP
     GOTO 75

  74 KNOR3 = -1
     IF (BNN.EQ.1) GOTO 660
     BNN = 1
     WRITE(*,*)
 117 WRITE(*,2000)
2000 FORMAT(' Enter file name of background data (blank = none). ? ',\)
     READ(*,'(30A)') NF02
     IF (NF02.EQ.' ') NBNN = -1
     IF (NF02.EQ.' ') KNOR3 = 1
     IF (NF02.EQ.' ') WRITE (*,*)
     IF (NBNN.EQ.-1) GOTO 660
     OPEN(UNIT=N12,FILE=NF02,ERR=2100,STATUS='OLD')
     IF (N8.GT.0) WRITE(N8,3100) NF02
3100 FORMAT(//' BACKGROUND FILE NAME ............. ',30A)

  75 WRITE(*,9071)
9071 FORMAT(' Calibration channel ............................ ? ',\)
     READ(*,*) DPG
     IF (KNOR3.EQ.1)  FDPG = DPG
     IF (KNOR3.EQ.-1) BDPG = DPG
     IF (N8.GT.0) WRITE(N8,8051) DPG
8051 FORMAT(' CALIBRATION CHANNEL ............... ',1P,1E8.3E1)

     WRITE(*,60)
  60 FORMAT(' Zero Intercept (Channel) ......................... ? ',\)
     READ(*,*) XINTX
     IF (KNOR3.EQ.1)  FXINTX = XINTX
     IF (KNOR3.EQ.-1) BXINTX = XINTX
     IF (N8.GT.0) WRITE(N8,8009) XINTX
8009 FORMAT(' ZERO INTERCEPT ................... ',1P,1E8.3E1)

     WRITE(*,61)
  61 FORMAT(' Live Time (sec) ................................ ? ',\)
     READ(*,*) XNOR1
     IF (KNOR3.EQ.1)  FXNOR1 = XNOR1
     IF (KNOR3.EQ.-1) BXNOR1 = XNOR1
     IF (N8.GT.0) WRITE(N8,8011) XNOR1
8011 FORMAT(' LIVE TIME (SEC) .................. ',1P,1E8.3E1)

     XNOR2 = 0
     IF (NR.EQ.1) WRITE(*,9871)
9871 FORMAT(' Reactor Power (kW) ............................ ? ',\)
     IF (NR.EQ.1) READ(*,*) XNOR2
```

```
      IF (KNOR3.EQ.1)  FXNOR2 = XNOR2
      IF (KNOR3.EQ.-1) BXNOR2 = XNOR2
      IF (NR.EQ.0) GOTO 960
      IF (N8.GT.0) WRITE(N8,8012) XNOR2
 8012 FORMAT(' REACTOR POWER (kW) ................ ',1P,1E8.3E1)
  960 IF (N8.GT.0) WRITE(N8,8014) EGP
 8014 FORMAT(' LIGHT UNITS AT CALIBRATION POINT .. ',1P,1E8.3E1)

      WRITE(*,961)
  961 FORMAT(' Number of smoothing iterations (0 to 8) .......... ? ',\)
      READ(*,*) NSMOTH
      IF (KNOR3.EQ.1)  FNSMTH = NSMOTH
      IF (KNOR3.EQ.-1) BNSMTH = NSMOTH
      IF (KNOR3.EQ.-1) WRITE (*,*)
      IF (KNOR3.EQ.-1) KNOR3=1
      IF (N8.GT.0) WRITE(N8,8013) NSMOTH
 8013 FORMAT(' NUMBER OF SMOOTHING ITERATIONS .... ',I2,/)

  660 IF (BNN.EQ.0) GOTO 74
      IF (KNOR3.EQ.1) GOTO 670
      IF (NBNN.EQ.-1) GOTO 5003
      NSMOTH = BNSMTH
      XNOR2  = BXNOR2
      XNOR1  = BXNOR1
      DPG    = BDPG
      XINTX  = BXINTX
      NN = N12
      GOTO 680
  670 NSMOTH = FNSMTH
      XNOR2  = FXNOR2
      XNOR1  = FXNOR1
      DPG    = FDPG
      XINTX  = FXINTX
      NN = N1

  680 IF (MES.EQ.1) GOTO 681
      MES=1
      WRITE (*,*) 'Unfolding data ...'
  681 XINTP = XINTX - 1.0
      IF (XNOR1.EQ.0) XNOR1 = 1.0
      IF (XNOR2.EQ.0) XNOR2 = 1.0
      IF (XNOR4.EQ.0) XNOR4 = 1
      DPG = EGP/(DPG - XINTX)

      XNOR = XNOR1 * XNOR2 * KNOR3
      XNOR = XNOR4 / XNOR

C.... CALCULATE LIGHT UNITS OF EACH CHANNEL

      K = 1
      XL(1) = (0.5 - XINTP)*DPG
      DO 200 N = 2, NMAX
```

```
      K = K + 1
  200 XL(K) = XL(K-1) + DPG
      XLMAX = XL(K)

C.... READ IN COUNTS FROM FILE (UNIT=1)

      DO 701 I = 1, NMAX
      IA(I) = 0
  701 A(I) = 0.0
      J = 0
      READ (NN,'(80A)') COM
      READ (NN,'(9X,20A)') FMTC
      READ (NN,FMTC,END=702) (IA(I),I=1,NMAX)
  702 DO 704 I=NMAX,1,-1
  704 IF (IA(I).GT.0) GOTO 705
  705 NCHL = I+1
      DO 703 I = 1, NCHL
  703 A(I) = IA(I)

      CLOSE (UNIT=NN)
      IF (N8.EQ.0) GOTO 345
      IF (KNOR3.EQ.1)  WRITE(N8,'(A19,/)') ' FOREGROUND COUNTS:'
      IF (KNOR3.EQ.-1) WRITE(N8,'(A19,/)') ' BACKGROUND COUNTS:'
      IF (N8.GT.0.AND.IPN.LE.1) WRITE(N8,8071) DPG
 8071 FORMAT(/,' CALCULATED GAIN = ',F9.7, '  LIGHT UNITS / CHANNEL',/)
      WRITE(N8,'(80A)') COM
      DO 750 I=1,NCHL,8
  750 WRITE(N8,8016) I,(IA(L),L=I,I+7)
 8016 FORMAT(1X,I4,2X,8I8)
      WRITE (N8,'(/)')

C.... SMOOTH DATA NSMOTH ITERATIONS

  345 IF (NSMOTH.EQ.0) GO TO 370
      KB = 1
      KT = NCHL + KB - 1
      DO 350 KS = 1, NSMOTH
  350 CALL SMOOTH (A,KB,KT)
      KB = KB + NCHL
      IF (N8.EQ.0.AND.IPN.LE.1) GOTO 370
      WRITE(N8,351)
  351 FORMAT (' SMOOTHED COUNTS:',/)
      DO 352 I=1,NCHL,7
  352 WRITE(N8,355) I,(A(L),L=I,I+6)
  355 FORMAT(1X,I4,1X,7F10.1)
      WRITE (N8,'(/)')

C.... CAL. OF RSUM, DOUBLE PRECISION

  370 RSUM(1) = DBLE(A(1) * XNOR)
      DO 400 I = 2, NCHL
      RSUM(I) = RSUM(I-1) + DBLE(A(I) * XNOR)
```

```
  400 CONTINUE
      DO 600 I = 1, KR
        IF (PC(I).GT.XLMAX) GO TO 600
        AUX = PC(I) - DPC(I)*0.5
        TEMP = PC(I) + DPC(I)*0.5
        CALL DTERP(XL,RSUM,NCHL,2,AUX,RL)
        CALL DTERP(XL,RSUM,NCHL,2,TEMP,RH)
        R(I) = SNGL(RH - RL)
        XR(I) = XR(I) + R(I)
  600 CONTINUE
      CALL DTERP(XL,RSUM,NCHL,0,TEMP,RH)

C.... CAL. OF DRSUM, DOUBLE PRECISION, USING SAME AREA FOR RSUM

      RSUM(1) =  DBLE(A(1)*XNOR**2)
      DO 401 I = 2, NCHL
        RSUM(I) = RSUM(I-1) + DBLE(A(I) * XNOR**2)
  401 CONTINUE
      DO 601 I = 1, KR
        IF (PC(I).GT.XLMAX) GOTO 601
        AUX = PC(I) - DPC(I)*0.5
        TEMP = PC(I) + DPC(I)*0.5
        CALL DTERP(XL,RSUM,NCHL,2,AUX,DRL)
        CALL DTERP(XL,RSUM,NCHL,2,TEMP,DRH)
        DR(I) = SNGL(DABS(DRH - DRL))
        SXR(I) = SXR(I) + DR(I)
  601 CONTINUE
      CALL DTERP(XL,RSUM,NCHL,0,TEMP,DRH)
      IF (KNOR3.EQ.1) GOTO 74

 5003 SXRMIN = 10000000
      DO 4500 I = 1, KR
        SXR(I) = SQRT(SXR(I))
        IF (XR(I).LT.0) XR(I) = 0
        IF (SXR(I).LT.SXRMIN.AND.SXR(I).GT.0) SXRMIN=SXR(I)
        IF (XR(I).LE.0)  SXR(I) = SXRMIN/2
        IF (SXR(I).GT.4000000) SXR(I) = 1.0E-4
        R(I)  = XR(I)
        DR(I) = SXR(I)
 4500 CONTINUE

 9999 RETURN
      END

C-----------------------------------------------------------------------------

      SUBROUTINE SMOOTH (Y,KB,NPTS)

      DIMENSION Y(1)

      IMAX = NPTS - 1
      YI = Y(KB)
```

```
      DO 24 I = KB, IMAX
      YNEW = (YI + 2.0 * Y(I) + Y(I+1)) / 4.0
      YI = Y(I)
   24 Y(I) = YNEW
      Y(NPTS) = (YI + 3.0 * Y(NPTS)) / 4.0
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE DTERP(X,Y,NPTS,NTERMS,XIN,YOUT)

      DOUBLE PRECISION YOUT,DELTAX
      DOUBLE PRECISION Y(2048),DELTA(2),A(2)
      DIMENSION X(1)
      DATA ISTART/1/

      IF (NTERMS.GT.0) GOTO 11
      ISTART = 1
      GOTO 61

C.... SEARCH FOR APPROPRIATE VALUE OF X(1)

   11 DO 19 I = ISTART, NPTS
      IF (XIN - X(I)) 12, 17, 19
   12 I1 = I
   13 I1 = I1 - 1
      IF (I1.LE.0) GOTO 14
      IF (XIN - X(I1)) 13, 17, 21
   14 I1 = 1
      GOTO 21
   17 YOUT = Y(I)
      ISTART = I
      GO TO 61
   19 CONTINUE
      I1 = NPTS - 1
   21 I2 = I1 + 1
      IF (NPTS - I2) 23, 31, 31
   23 I2 = NPTS
      I1 = I2 - 1

C.... EVALUATE DEVIATIONS DELTA
   31 ISTART = I2
      DENOM = X(I2) - X(I1)
      DELTAX = DBLE((XIN - X(I1))/DENOM)
      DO 35 I = 1, 2
      IX = I1 + I - 1
   35 DELTA(I) = DBLE((X(IX) - X(I1))/DENOM)

C.... ACCUMULATE COEFFICIENTS AND SUM OF EXPANSION
      A(2) = (Y(I2)-Y(I1))/(DELTA(2)-DELTA(1))
      YOUT = Y(I1) + A(2) * (DELTAX - DELTA(1))
```

```
   61 RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE INTP(E,PLO,PUP,IEC,E1,E2,NL)

      DOUBLE PRECISION XL,XU
      DIMENSION E(1),PLO(1),PUP(1)
      DIMENSION ER(100),PL(100),PU(100)
      DIMENSION E1(1), E2(1)
      CHARACTER TITLE*80,TFILE*30,PFILE*30
      COMMON /PPU1/ TITLE,PFILE,TFILE
      COMMON /PPU2/ N7,N10,NPW

C.... INVERT ARRAYS
      DO 50, I=1,IEC
      ER(I) = E(IEC-I+1)
      PL(I) = PLO(IEC-I+1)
   50 PU(I) = PUP(IEC-I+1)

      OPEN (UNIT=N10,FILE=TFILE,ERR=60,STATUS='OLD')
      CLOSE (N10,STATUS='DELETE')
   60 OPEN (UNIT=N10,FILE=TFILE,STATUS='NEW')

C.... WRITE OUT UNFOLDED DATA
      WRITE(N10,'(80A)') TITLE
      IF (NPW.EQ.1) WRITE(N10,350)
      IF (NPW.EQ.0) WRITE(N10,351)
      WRITE(N10,360)
      N = INT(IEC/2)
      IF (IEC.GT.(N*2)) GOTO 100
      WRITE(N10,370)(ER(I),PL(I),PU(I),ER(I+N),PL(I+N),PU(I+N),I=1,N)
      GOTO 150
  100 N = N + 1
      WRITE(N10,370) (ER(I),PL(I),PU(I),ER(I+N),PL(I+N),PU(I+N),I=1,N-1)
      WRITE(N10,370)  ER(N),PL(N),PU(N)
  350 FORMAT (/,1H ,2('    NEUTRON     FLUX(N/(CM2*S*MeV*kW))'))
  351 FORMAT (/,1H ,2('    NEUTRON     FLUX (N/(CM2*S*MeV))  '))
  360 FORMAT (1H ,2('    ENERGY      LOWER       UPPER     ')/1H ,
     +           2('    ( MeV )     LIMIT       LIMIT     ')/1H ,
     +           2('    --------    --------    --------  '))
  370 FORMAT (1H ,1P,3E12.2,2X,3E12.2)

C.... WRITE OUT INTEGRAL OUTPUT
  150 WRITE (N10,380)
      IF (NPW.EQ.1) WRITE (N10,390)
      IF (NPW.EQ.0) WRITE (N10,391)
      WRITE (N10,392)
  380 FORMAT (1H ,//,1H ,15X,' E1       E2   ',6X,'INTEGRAL',8X,'ERROR')
  390 FORMAT (16X,'(MeV)   (MeV)',3X,'(N/(CM2*S*kW))',2X,
```

```
     +'(N/(CM2*S*kW))')
  391 FORMAT (16X,'(MeV)   (MeV)',3X,'  (N/(CM2*S)) ',2X,
     +'  (N/(CM2*S)) ')
  392 FORMAT (16X,'-----   -----',3X,' -------------',2X,
     +' -------------')

      DO 400 I = 1, NL
      IF (E2(I).LE.ER(1))   GO TO 400
      AUX = E1(I)
      IF (E1(I).LT.ER(1))   AUX = ER(1)
      CALL XINT(ER,PL,IEC,3,AUX,E2(I),XL)
      CALL XINT(ER,PU,IEC,3,AUX,E2(I),XU)
      XAV = 0.5*(XL + XU)
      DX = 0.5*(XU - XL)
      WRITE (N10,410) AUX, E2(I), XAV, DX
  410 FORMAT (1H ,14X,F6.3,F8.3,1P,2E15.2)
  400 CONTINUE
      CLOSE (UNIT=N10)
      RETURN
      END

C-----------------------------------------------------------------------------

      SUBROUTINE XINT (E,F,NPTS,NTERMS,E1,E2,SUM)

      DIMENSION E(1), F(1), X(300), Y(300)
      DOUBLE PRECISION SUM
      DATA IST1/1/,IST2/1/

      DO 100 I = IST1,NPTS
  100 IF (E1.LT.E(I)) GOTO 200
  200 I = I - 1
      IF (I.EQ.0) GOTO 250
      IF (E1.LT.E(I)) GOTO 200
  250 IL = I + 1
      IST1 = IL
      SUM = 0.0
      DO 300 I = IST2, NPTS
  300 IF (E2.LT.E(I)) GOTO 400
  400 I = I - 1
      IF (I.EQ.0) GOTO 450
      IF (E2.LT.E(I)) GOTO 400
  450 IH = I
      IST2 = IH

      IF (IH.LT.IL) GOTO 700
      ILA = IL - NTERMS/2
      IF (ILA.LE.0)   ILA = 1
      IF (ILA + NTERMS - 1.GT.NPTS)   ILA = NPTS - NTERMS + 1
      IHA = IH - NTERMS/2
      IF (IHA.LT.1)   IHA = 1
      IF (IHA + NTERMS - 1.GT.NPTS)   IHA = NPTS - NTERMS + 1
```

```
      CALL ITEG (E,F,NTERMS,ILA,E1,E(IL),SUM)
      CALL ITEG (E,F,NTERMS,IHA,E(IH),E2,SUM)
      GO TO 800
  700 ILA = IL - NTERMS/2
      IF (ILA.LE.0)   ILA = 1
      IF (ILA + NTERMS - 1.GT.NPTS)   ILA = NPTS - NTERMS + 1
      CALL ITEG (E,F,NTERMS,ILA,E1,E2,SUM)
  800 CONTINUE
      K = 0
      DO 500 I = IL, IH
      K = K + 1
      X(K) = E(I)
  500 Y(K) = F(I)
      IF (K.LE.1)   GO TO 900
      NPTSA = K
      SUM = SUM + AREA(X,Y,NPTSA,NTERMS)
  900 CONTINUE
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE ITEG (X, Y, NTERMS, I1, X1, X2, SUM)

C.... CONSTRUCT SQUARE MATRIX AND INVERT

      DOUBLE PRECISION XJK, ARRAY, A, DENOM, DELTAX, SUM
      DIMENSION X(1), Y(1)
      DIMENSION ARRAY (15,15)

   11 DO 17 J = 1, NTERMS
      I = J + I1 - 1
      DELTAX = X(I) - X(I1)
      XJK = 1.
      DO 17 K = 1, NTERMS
      ARRAY(J,K) = XJK
   17 XJK = XJK*DELTAX
   21 CALL MATINV (ARRAY,NTERMS,DET)
      IF (DET) 31, 23, 31
   23 IMID = I1 + NTERMS/2
      SUM = SUM + Y(IMID)*(X2 - X1)
      GO TO 40

C.... EVALUATE COEFFICIENTS AND INTEGRATE

   31 DX1 = X1 - X(I1)
      DX2 = X2 - X(I1)
   33 DO 39 J = 1, NTERMS
      A = 0.
      DO 37 K = 1, NTERMS
      I = K + I1 - 1
   37 A = A + Y(I)*ARRAY(J,K)
```

```
      DENOM = J
   39 SUM = SUM + (A/DENOM)*(DX2**J - DX1**J)
   40 RETURN
      END
C-----------------------------------------------------------------------

      FUNCTION AREA (X,Y,NPTS,NTERMS)

      DOUBLE PRECISION SUM
      DIMENSION X(1), Y(1)
   11 SUM = 0.
      IF (NPTS - NTERMS) 21, 21, 13
   13 NEVEN = 2*(NTERMS/2)
      IDELTA = NTERMS/2 - 1
      IF (NTERMS - NEVEN) 31, 31, 51

C.... FIT ALL POINTS WITH ONE CURVE
   21 X1 = X(1)
      X2 = X(NPTS)
   23 CALL ITEG(X,Y,NPTS,1,X1,X2,SUM)
      GO TO 71

   31 X1 = X(1)
      J = NTERMS - IDELTA
      X2 = X(J)
      CALL ITEG (X,Y,NTERMS,1,X1,X2,SUM)
      I1 = NPTS - NTERMS + 1
      J = I1 + IDELTA
      X1 = X(J)
      X2 = X(NPTS)
   39 CALL ITEG (X,Y,NTERMS,I1,X1,X2,SUM)
      IF (I1 - 2) 71, 71, 41
   41 IMAX = I1 - 1
      DO 46 I = 2, IMAX
      J = I + IDELTA
      X1 = X(J)
      X2 = X(J+1)
   46 CALL ITEG (X,Y,NTERMS,I,X1,X2,SUM)
      GO TO 71

   51 X1 = X(1)
      J = NTERMS - IDELTA
      X2 = (X(J) + X(J-1))/2.
      CALL ITEG (X,Y,NTERMS,1,X1,X2,SUM)
      I1 = NPTS - NTERMS + 1
      J = I1 + IDELTA
      X1 = (X(J) + X(J+1))/2.
      X2 = X(NPTS)
   59 CALL ITEG (X,Y,NTERMS,I1,X1,X2,SUM)
      IF (I1 - 2) 71, 71, 61
   61 IMAX = I1 - 1
```

```
      DO 66 I = 2, IMAX
      J = I + IDELTA
      X1 = (X(J+1) + X(J))/2.
      X2 = (X(J+2) + X(J+1))/2.
   66 CALL ITEG (X,Y,NTERMS,I,X1,X2,SUM)
   71 AREA = SUM
      RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE MATINV (ARRAY, NORDER, DET)

C.... FIND LARGEST ELEMENT ARRAY(I,J) IN REST OF MATRIX

      DOUBLE PRECISION ARRAY, AMAX, SAVE
      DIMENSION ARRAY(15,15), IK(15), JK(15)
   10 DET = 1.
   11 DO 100 K = 1, NORDER

      AMAX = 0.
   21 DO 30 I = K, NORDER
      DO 30 J = K, NORDER
   23 IF (DABS(AMAX) - DABS(ARRAY(I,J))) 24, 24, 30
   24 AMAX = ARRAY (I,J)
      IK(K) = I
      JK(K) = J
   30 CONTINUE

C.... INTERCHANGE ROWS AND COLUMNS TO PUT AMAX IN ARRAY(K,K)
   31 IF (AMAX) 41, 32, 41
   32 DET = 0.
      GO TO 140
   41 I = IK(K)
      IF (I - K) 21, 51, 43
   43 DO 50 J = 1, NORDER
      SAVE = ARRAY(K,J)
      ARRAY(K,J) = ARRAY(I,J)
   50 ARRAY(I,J) = -SAVE
   51 J = JK(K)
      IF (J - K) 21, 61, 53
   53 DO 60 I = 1, NORDER
      SAVE = ARRAY (I,K)
      ARRAY (I,K) = ARRAY(I,J)
   60 ARRAY (I,J) = -SAVE

C.... ACCUMULATE ELEMENTS OF INVERSE MATRIX
   61 DO 70 I = 1, NORDER
      IF (I - K) 63, 70, 63
   63 ARRAY(I,K) = -ARRAY(I,K) / AMAX
   70 CONTINUE
   71 DO 80 I = 1, NORDER
```

```
      DO 80 J = 1, NORDER
      IF (I - K) 74, 80, 74
   74 IF (J - K) 75, 80, 75
   75 ARRAY (I,J) = ARRAY(I,J) + ARRAY(I,K)*ARRAY(K,J)
   80 CONTINUE
   81 DO 90 J = 1, NORDER
      IF (J - K) 83, 90, 83
   83 ARRAY(K,J) = ARRAY(K,J) / AMAX
   90 CONTINUE
      ARRAY(K,K) = 1. / AMAX
  100 DET = DET * AMAX

C.... RESTORE ORDERING OF MATRIX
  101 DO 130 L = 1, NORDER
      K = NORDER - L + 1
      J = IK(K)
      IF (J - K) 111, 111, 105
  105 DO 110 I = 1, NORDER
      SAVE = ARRAY(I,K)
      ARRAY(I,K) = - ARRAY(I,J)
  110 ARRAY(I,J) = SAVE
  111 I = JK(K)
      IF (I - K) 130, 130, 113
  113 DO 120 J = 1, NORDER
      SAVE = ARRAY(K,J)
      ARRAY(K,J) = -ARRAY(I,J)
  120 ARRAY(I,J) = SAVE
  130 CONTINUE
  140 RETURN
      END
```

DISTRIBUTION

1. R. G. Alsmiller
2. B. R. Appleton
3. J. M. Barnes
4. K. Chatani
5. J. K. Dickens
6. J. D. Drischler
7. W. W. Engle
8. C. Y. Fu
9. T. A. Gabriel
10. D. T. Ingersoll
11. J. O. Johnson
12. D. C. Larson
13. F. C. Maienschein
14. F. J. Muckenthaler
15. J. V. Pace
16. R. W. Peelle
17. F. G. Perey

18-22. B. D. Rooney

23. R. W. Roussin
24. R. T. Santoro
25. C. O. Slater
26. R. R. Spencer
27. Central Research Library

28-32. EPMD Reports Office

33. ORNL Y-12 Technical Library Document Reference Section

34-36. Laboratory Records

37. ORNL Patent Office

38-47. RSIC

EXTERNAL DISTRIBUTION

48. Office of Assistant Manager for Energy Research and Development, DOE-ORO, P.O. Box 2001, Oak Ridge, TN 37831-8600.

49-58. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831.

59. Dr. John J. Dorning, Department of Nuclear Engineering and Engineering Services, Thornton Hall, University of Virginia, Charlottesville, VA 22901.

60. Dr. Robert M. Haralick, Department of Electrical Engineering, University of Washington, Seattle, WA 98195.