GJB X-414 81

SAI 272-81-177-LJ

MASTER

82005557

National Uranium Resource Evaluation

ARES AUTOMATED RESPONSE FUNCTION CODE

USER MANUAL

By T. Maung
G. M. Reynolds

DO NOT MICROFILM
COVER

SCIENCE APPLICATIONS, INC.
La Jolla, California 92037

June 1981

PREPARED FOR U.S. DEPARTMENT OF ENERGY
Assistant Secretary for Nuclear Energy
Grand Junction Office, Colorado

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

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.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

This report is a result of work performed by Science Applications, Inc. through a Bendix Field Engineering Corporation subcontract, as part of the National Uranium Resource Evaluation. NURE is a program of the U.S. Department of Energy's Grand Junction, Colorado, Office to acquire and compile geologic and other information with which to assess the magnitude and distribution of uranium resources and to determine areas favorable for the occurrence of uranium in the United States.

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 in this report, or represents that its use would not infringe privately owned rights. Reference therein 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.

DO NOT MICROFILM
COVER

ARES
AUTOMATED RESPONSE FUNCTION CODE
USER MANUAL

T. Maung
G.M. Reynolds

SCIENCE APPLICATIONS, INC.
La Jolla, California
Report No. SAI-272-81-177-LJ

June 1981

PREPARED FOR THE U.S. DEPARTMENT OF ENERGY
ASSISTANT SECRETARY FOR NUCLEAR ENERGY
GRAND JUNCTION OFFICE, COLORADO
UNDER CONTRACT NO. DE-AC13-76GJ01664
AND BENDIX FIELD ENGINEERING CORPORATION
SUBCONTRACT NO. 79-340-L

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
PREFACE	
1. DESCRIPTION OF SOFTWARE	7
1.1 General Description.	7
1.2 Hardware Requirement	7
1.3 Computational Sequence	8
2. DETAILED DESCRIPTION OF PROGRAM INPUTS AND OUTPUTS . .	11
2.1 HPGAM.	11
2.2 UTLTY.	14
2.3 LSQVM.	14
2.4 REPIT.	15
2.5 RESPN.	15
2.6 PLTTR.	15
3. PROGRAM LOADING AND OPERATION	17
3.1 Loading Programs into HP-1000.	17
3.2 Test Runs.	17
3.3 Plot Options	19
4. LISTING OF PROGRAMS	21
4.1 HPGAM.	21
4.2 LSQVM.	45
4.3 UTLTY.	67
4.4 RFSPN.	75
4.5 PLTTR.	79
4.6 File Manipulating Routines	85
4.6.1 MERGE	85
4.6.2 MKFL2	87
4.6.3 MKFL3	91

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1	ARES Software Flow Diagram	9
2	Coordinates and Geometries for Detectors	12

LIST OF TABLES

<u>Table</u>		
1	Sample Input File	13

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

PREFACE

This ARES user's manual provides detailed instructions for a general understanding of the Automated Response Function Code and gives step by step instructions for using the complete code package on a HP-1000 system.

This code is designed to calculate response functions of NaI gamma-ray detectors, with cylindrical or rectangular geometries.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

1. DESCRIPTION OF SOFTWARE

1.1 General Description

The Automated Response Function (ARES) software is written in modularized form to fit the architecture of the HP-1000 minicomputer. The major components of the ARES package are:

I. HPGAM. A fast running Monte Carlo code to compute the energy depositions by gamma-rays in sodium iodide (NaI) crystals with either aluminum or stainless steel cladding. This program can handle either a cylindrical or a rectangular detector geometry. The program was originally coded by Evans, et al¹, at Los Alamos and was named GAMRES. We have modified the program extensively and the new HP version is called HPGAM.

II. LSQVM. A generalized multiparameter Least Square Minimizing program based on Variable Metric Method originally suggested by W. C. Davidson² of the Argonne National Laboratory. The present version of the program has been extensively modified to suit the specific purpose of producing analytic fits to the computed energy depositions as functions of gamma-ray energy and pulse height in the detector.

III. Utility codes for manipulating input/output and producing graphic plots.

1.2 Hardware Requirement

The software package is configured for a HP-1000 computer system with a 7920-M Disc Driver and an on-line 2608-A line printer. The code package is delivered on a 9 track 1600 BPI magnetic tape. A tape unit comparable to HP model 7970-E is therefore required for loading the program.

-
1. NDA Technology for Uranium Resource Evaluation, LA-6996-PR, October, 1977.
  2. Variable Metric Method for Minimization, ANL-5990, 1959.

1.3 Computational Sequence

The computational sequence of ARES from initial inputs to producing output response files or plots is shown in the flow diagram in Figure 1. The general flow sequence will be described here. The detailed input/output, file contents, and their structures will be presented in Section 2.

Three input files INPT01, INPT02, and INPT03 are required to run the complete software package and will be expected to be present on the disk. HPGAM uses the initial conditions from INPT01 and will produce an output file OUT01 of energy deposition values at the same time printing out the results on the line printer. The first utility program UTLTY combines the input file INPT02 and the output file OUT01 to create an input file LSQIN for the LSQVM routine. Running the LSQVM routine will produce an output file OUT02, containing the best parameters for the responses, at the same time printing the chi-squared values for the fit. Update file INPT02 with the results of OUT02 by running the utility program REPIT, which will repeat execution of UTLTY and LSQVM. After examination of the printed outputs from LSQVM and comparing the chi-squared values in the successive runs either repeat the above loop or proceed to the next program. If the consecutive chi-squared values are not different, proceed to the next program RESPN, otherwise repeat the above loop. It may be necessary to repeat the loop up to five times in some extreme cases when the statistics of the HPGAM produced results are poor.

Next, execute RESPN to produce detector responses for photon energies present in input file INPT03. The RESPN routine will create an output file OUTPT at the same time printing a list of the computed response functions. The responses from the output file OUTPT can be plotted on the graphic line printer by executing the PLTTR routine. PLTTR will smear the pulse height distributions with detector resolution before producing the plots.

Step by step execution of the program and I/O file structures and input/output parameters are described in detail in section 2.

ARES SOFTWARE FLOW DIAGRAM

SAI-81TM-2

Figure 1.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

2. DETAILED DESCRIPTION OF PROGRAM INPUTS AND OUTPUTS

2.1 HPGAM

The Monte Carlo transport code HPGAM uses two files, one for input and one for output. The input is called INPTØ1 and the output is OUTØ1. The structures of the various files are described below.

2.1.1 Input File INPTØ1

This file is an ASCII file expected to exist on the disk containing the following information:

- Line 1-3   Comment cards describing the computational setup and the detector and source geometry. Information to help the user for bookkeeping purposes only.
- Line 4   XB, YB, ZB1, ZB2, XBB, YBB, ZBB2, IGEO with format (7F10.4, I10) IGEO = 1 for rectangular detector IGEO = 2 for cylindrical detector. Other quantities are defined in Figure 2.
- Line 5   Position of the source along Z axis (negative) and the maximum opening angle (degrees) for the source. Format (2F10.4).
- Line 6   Two detector resolution values (absolute) to be folded in for printed outputs. Format (2F10.4).
- Line 7   ESTART = photon energy, ANEV = number of events to transport, MDUMP = number of events for which parameters will be printed on the line printer. Format (2E12.5, I10).
- Line 8-On   Repeat LINE 7 for different energy photons.
- Last Line   Set ANEV = 0. to terminate program.

A sample input file INPTØ1 is shown in Table 1.

A. Rectangular Detector.

B. Cylindrical Detector.

SAI-81TM-1

Figure 2. Coordinates and Geometries for Detectors.

THIS IS A SET OF 16X16 INPUT DATA
 POINT SOURCE IS AT 27.5 INCHES FROM THE FRONT FACE
 FRONT FACE 0.2 CM, AND BACK SCATTERING REGION 0.50 CM

20.32	20.32	0.2	10.36	20.52	20.52	10.86	1
-69.85	14.0						
.09	.10						
0.100		90000		0			
0.2		90000		0			
0.3		90000		0			
0.4		90000		0			
0.5		90000		0			
0.6		90000		0			
0.7		90000		0			
0.8		90000		0			
0.9		90000		0			
1.0		90000		0			
1.1		90000		0			
1.2		90000		0			
1.21		90000		0			
1.4		90000		0			
1.6		90000		0			
1.8		90000		0			
2.0		90000		0			
2.2		90000		0			
2.4		90000		0			
2.6		90000		0			
2.8		90000		0			
3.0		90000		0			
2.614		0		0			

Table 1. Sample Input File INPT01.

2.1.2 Output File OUTØ1.

This is a binary file with two records for each gamma-ray energy. Record one has three words consisting of photon energy, number of photons transported, and number of photons interacted in the detector. Record two is an array with 100 elements giving the pulse height distribution.

2.2 UTLTY

This utility program combines two input files INPTØ1 and INPTØ2 to create an output file LSQIN.

2.2.1 INPTØ1. Structure of this file is described in section 2.1.2.

2.2.2 INPTØ2.

This input file contains the initial vlaues for the coefficients of the analytic functions used by the general fitting routine LSQVM. The file is created by running the file-creating program MKFL2. There are eight functions with 25 coefficients in each, and four functions with five coefficients in each, requiring a total of 220 initial values. A safe and convenient set of starting initial values can be created by repeatedly typing in a value of 1.0 for all coefficients.

2.2.3 Output File LSQIN.

This is a binary file created by combining the pulse height distributions as a function of photon energy from OUTØ1 and the initial values for the fitting parameters from INPTØ2, to be used as input to the fitting program LSQVM.

2.3 LSQVM.

This program uses the file LSQIN as input and creates an output file OUTØ2 with the best fit parameters. It also prints out the chi-squared values for the best fit.

2.4   REPIT

This is just a command file used to repeat the fitting process. It updates the initial values in INPT02 with the values from OUT02 and creates a new LSQIN input file. It then runs the LSQVM fitting routine to generate the most up-to-date fitting parameters.

2.5   RESPN

This program uses two files as input; namely, OUT02 and INPT03, and creates an output file OUTPT containing the responses.

2.5.1   INPT03.

INPT03 is a binary input file consisting of ENERGY, SLOPE, and EZERO for each detector response desired. This file is created by running file-creating program MKFL3.

2.5.2   OUT02.

OUT02 is described in section 2.3

2.5.3   OUTPT.

This binary output file contains the unsmeared pulse height distributions calculated using the best fit parameters from OUT02. It has two records for each energy. First record has three words ENERGY, SLOPE, and EZERO. The second record has the pulse height distribution, the number of elements depending on photon ENERGY, SLOPE, and EZERO, with a maximum of 256 elements.

The last record has entry ENERGY equal to 0.0.

2.6   PLTTR

This program uses disk file OUTPT as the input and plots the response curves on the HP printer. No output file is created.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

3. PROGRAM LOADING AND OPERATION

3.1 LOADING ARES PROGRAM INTO HP-1000

As described in Section 1, the ARES code package is provided on a 1600 BPI 9-track tape. The loading of the program is described here by assigning disk cartridge number 26 to the user.

3.1.1 Mount tape on the tape driver making sure that the density is set at 1600 BPI. Load tape and set it on line.

3.1.2 Transfer the control file *TAPE from tape to disk by typing the following file manager command:

```
ST, 8, *TAPE::26
```

3.1.3 Transfer control to the control file *TAPE by typing the command:

```
TR, *TAPE
```

The computer will now load the entire ARES source program package from the tape to disk cartridge 26.

3.2 TEST RUNS

To checkout the ARES program package follow the step-by-step running instructions listed below.

3.2.1 Create an input file INPT01 using the system EDITOR. Before creating this file rename and save the original INPT01 file copied from the program tape.

3.2.2 Compile the HPGAM program, create compile files, load and run the program by using the following commands:

```
TR, *XFGAM
```

```
TR, *XGAM
```

Note that HPGAM will create output file OUTØ1.

3.2.3 The ARES detector response generation package is set up to operate on a set of 22 separate input photon energies. HPGAM can be run with a few input energies at a time, recombining and renaming the output file OUTØ1 using the utility program MERGE until the set of 22 input energies, given in Table 1, has been run.

3.2.4 Compile and run the program UTLTY to create the input file LSQIN for the fitting routine LSQVM.

3.2.5 Compile and run the fitting program LSQVM by using the following commands:

TR, *XFCRV

TR, *XCRV

This program will create the output file OUTØ2.

3.2.6 Note the printed chi-squared values produced by LSQVM. Repeat the fitting process by using the command:

TR, *REPIT

This command will cause the computer to

Update INPTØ2 with OUTØ2

Purge OUTØ2

Run UTLTY to create a new LSQIN

Run LSQVM to produce the next set of fitting parameters.

Compare the new chi-squared values with the previous set, repeating the fitting process if necessary, or proceed to step 3.2.7.

3.2.7 The output file OUTØ2 contains the best fit coefficients for generating the detector response. Next, run the RESPN routine to generate unsmeared detector responses at gamma energies entered in INPTØ3.

The calculated detector responses will be written out in file OUTPT and also printed out on the line printer.

3.3 PLOT OPTIONS

The unsmeared energy depositions from output file OUTPT can be plotted after folding in the detector resolution. This is accomplished using the routine PLTTR.

3.3.1 Create a compile file for PLTTR using the command:

```
RU, FTN4, &PLTTR, 0, %PLTTR
```

3.3.2 Load and execute the PLTTR routing using the command:

```
TR, *PLTTR
```

The program will ask for the number of response curves to be plotted from the file OUTPT. The resolution of the detector and the energy dependence of the resolution (POWER) have to be typed in as input to the program.

The response curves are plotted on one graph by superimposing all the desired distributions.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

4. LISTING OF PROGRAMS

4.1 HPGAM LISTING

&HPGAM T=00004 IS ON CR00026 USING 00045 BLKS R=0000

```

0001  FTN4,L
0002      PROGRAM HPGAM
0003  C  MONTECARLO CODE FOR SODIUM IODIDE,  REVISED MAY 1,1980.
0004  C  MAIN ROUTINE.
0005  C  PROGRAM IS NOW SET UP TO RUN WITH NAI DETECTORS WHICH HAS
0006  C  ALUMINUM CLADDING.
0007  C  MAKE SURE TO USE PROPER ALUMINUM OR IRON DENSITIES AND
0008  C  CROSS SECTIONS BY MAKING APPROPRIATE CHANGES IN THE
0009  C  FOLLOWING SUBROUTINES.
0010  C.....1.  BLOCK DATA
0011  C.....2.  SIGTT
0012  C
0013      DIMENSION LT(200,6),LCON(9),LHI(9),RESSIG(3)
0014      DIMENSION ISIZE(2)
0015      DIMENSION IOUT(144),INPUT(144),NIN(3),NOUT(3),IBUF(128),ITIME(5)
0016      DIMENSION ERGY(33),S(8),BUF(128)
0017      COMMON/BORDR/EP$
0018      COMMON/ANIHL/XANN,YANN,ZANN,CXANN,CYANN,CZANN,IMATAN
0019      COMMON/VARPR/A,PIB2,PIB4,U
0020      COMMON/RNDM/X1,X2
0021      COMMON/SCIN/ATDENS(2),RHO(2),IMAT
0022      COMMON/CONST/PI,TWOPI
0023      COMMON/CHECK/ESTART,ELOW
0024      COMMON/BNDS/XB,YB,ZB1,ZB2,XBB,YBB,ZBB2
0025      COMMON/SQUAR/XBSQ,XBBSQ
0026      COMMON/CORD/X,Y,Z,CX,CY,CZ
0027      COMMON/GEOM/IGEO,NCT(50),NEG
0028      COMMON/DUMP/NDUMP,MDUMP,EL(9)
0029      COMMON/VAR/B,B2,B3,AXX,A2,G,G2,G3,UX
0030      DIMENSION COMMNT(36)
0031      INTEGER COMMNT
0032      REAL LT,LCON
0033      DOUBLE PRECISION X1,X2
0034      DATA ERGY/.001,.0015,.002,.003,.004,.005,.006,.008,.01,.015,.02,
0035      * .03,.04,.05,.06,.08,.1,.15,.2,.3,.4,.5,.6,.8,1.,1.5,2.,3.,4.,5.,
0036      * 6.,8.,10./
0037      DATA ISIZE/16,0/
0038      DATA NIN/2HIN,2HPT,2H01/
0039      DATA NOUT/2HOU,2HT0,2H1 /
0040      ISEC=0
0041      ITYPE=4
0042      ICATR=26
0043      IDCBZ=128
0044      IOPT=0
0045      MS=0
0046      CALL CREAT(IOUT,IERR,NOUT,ISIZE,ITYPE,ISEC,ICATR,IDCBZ)
0047      IF(IERR.GE.0)GO TO 703
0048      WRITE(1,901)IERR
0049      901  FORMAT(' CREAT ERROR=',I6)
0050      GO TO 9900
0051      703  CALL OPEN(INPUT,IERR,NIN,IOPT,ISEC,ICATR,IDCBZ)
0052      IF(IERR.GE.0)GO TO 705
0053      WRITE(1,704)IERR
0054      704  FORMAT(' OPEN ERROR=',I6)
0055      GO TO 9900
0056  C  INITIATE RANDOM NUMBER GENERATOR
0057      705  CALL INIT
0058  C

```



```

0059 C
0060 C***** SECTION A ** DATA INPUT *****
0061 C (FOR MORE DETAIL SEE DESCRIPTION OF DATA DECK)
0062 C
0063     WRITE(6,54)
0064     DO 972 KR=1,3
0065     CALL READF(INPUT,IERR,IBUF,36,LEN)
0066     DO 971 I=1,LEN
0067     COMMNT(I)=IBUF(I)
0068     971 CONTINUE
0069     WRITE(6,56)(COMMNT(I),I=1,LEN)
0070     54 FORMAT(10H1DATA DECK)
0071     56 FORMAT(1H0/1X,36A2)
0072     972 CONTINUE
0073 C
0074 C*****READ SCINTILLATOR AND CAN BOUNDARIES, AND RESOLUTION VALUES
0075 C
0076 C SCINTILLATOR IS ZB CMS DEEP AND HAS TRANSVERSE DIMENSIONS
0077 C OF 2*XB AND 2*YB
0078 C IF SCINTILLATOR IS A CYLINDER, ITS RADIUS IS XB (=YB)
0079 C THE PARAMETER IGEO IDENTIFIES THE SCINTILLATOR GEOMETRY
0080 C IGEO=1 FOR BOX, IGEO=2 FOR CYLINDER(PHOTONS ENTER FLAT FACE)
0081 C
0082     WRITE(6,101)
0083     101 FORMAT(//,5X,'SYSTEM BOUNDARIES',//,11X,'XB',13X,'YB',12X,'ZB1',
0084     A 12X,'ZB2')
0085     CALL READF(INPUT,IERR,IBUF,40,LEN)
0086     CALL CODE(80)
0087     READ(IBUF,5)XB,YB,ZB1,ZB2,XBB,YBB,ZBB2,IGEO
0088     5 FORMAT(7F10.4,I10)
0089     WRITE(6,6)XB,YB,ZB1,ZB2
0090     6 FORMAT(4(5X,F10.4))
0091     WRITE(6,701)
0092     701 FORMAT(1H0,9X,'XBB',12X,'YBB',12X,'ZBB2',9X,'IGEO')
0093     WRITE(6,702)XBB,YBB,ZBB2,IGEO
0094     702 FORMAT(3(5X,F10.4),I10)
0095     XBSQ=XB*XB
0096     XBSQ=XBB*XBB
0097 C READ SOURCE Z POSITION AND MAXIMUM ANGLE FOR PHOTONS, IN DEGREES
0098 CALL READF(INPUT,IERR,IBUF,10,LEN)
0099 CALL CODE(20)
0100 READ(IBUF,5)ZSORCE,ANGLMX
0101 WRITE(6,7701)ZSORCE,ANGLMX
0102 7701 FORMAT(23H0PHOTON SOURCE IS AT Z=,F7.2,4H CMS/
0103 124H0MAXIMUM SOURCE ANGLE IS,F7.1,8H DEGREES)
0104 COSMAX=1.0-COS(ANGLMX*PI/180.0)
0105 COSMAX=-COSMAX
0106 ZSORCE=ABS(ZSORCE)
0107 C*****NOTE THIS ANGLE MAX IS USED FOR THE COLLIMATOR IN OUR SET-UP
0108 C
0109 C READ THE TWO DESIRED (NONZERO) VALUES OF DETECTOR RESOLUTION (AS A
0110 C FRACTION.
0111     CALL READF(INPUT,IERR,IBUF,10,LEN)
0112     CALL CODE(20)
0113     READ(IBUF,915)(RESSIG(I),I=2,3)
0114     915 FORMAT(2F10.4)
0115 C DO 34 I=1,2
0116 C KSIG=I
0117 C IF(I.EQ.1) WRITE(6,102)
0118 C 102 FORMAT(//,5X,'CROSS SECTIONS FOR SODIUM IODIDE IN BARNS/ATOM',/)

```

```

0119 C      IF(I.EQ.2) WRITE(6,103)
0120 CC103 FORMAT(///,5X,'CROSS SECTIONS FOR AL IN BARNS/ATOM',/)
0121 C 103 FORMAT(///,5X,'CROSS SECTIONS FOR IRON IN BARNS/ATOM',/)
0122 C      WRITE(6,104)
0123 C 104 FORMAT(8X,'ENERGY (MEV)',6X,'COMPTON',6X,'PHOTOELECTRIC',4X,
0124 C      A 'PAIR PRODUCTION')
0125 C      DO 36 J=1,33
0126 C      ERG=ERGY(J)
0127 C      CALL SIGTT(ERG,S,KSIG)
0128 C      WRITE(6,105)ERG,S(1),S(2),S(3)
0129 C 105 FORMAT(11X,F6.3,6X,E11.4,5X,E11.4,7X,E11.4)
0130 C 36 CONTINUE
0131 C 34 CONTINUE
0132 C
0133 C
0134 C ***** SECTION B ** GET UP FOR EACH GROUP OF HISTORIES *****
0135 C
0136 C INITIALIZE HISTOGRAM ARRAYS AND COUNTERS
0137 C
0138 C 1000 DO 10 I=1,6
0139 C      DO 9 J=1,200
0140 C      9 LT(J,I)=0.
0141 C      LCON(I)=0.
0142 C      10 LHI(I)=0
0143 C      DO 8 I=1,50
0144 C      8 NCT(I)=0
0145 C      ANDET=0.
0146 C      NDUMP=0
0147 C      RESSIG(1)=0.
0148 C
0149 C FOR EACH SET OF HISTORIES DESIRED, READ ONE CARD CONTAINING
0150 C THE INITIAL PHOTON ENERGY ESTART
0151 C THE NUMBER OF HISTORIES TO BE RUN, ANEV
0152 C THE DESIRED BIN WIDTH (MEV ELECTRON EQUIVALENT) FOR THE OUTPUT
0153 C PULSE HEIGHT HISTOGRAMS, BINW, IS ESTART/100.
0154 C
0155 C
0156 C      CALL READF(INPUT,IERR,TRUF,15,LEN)
0157 C      CALL CODE(30)
0158 C      READ(1,15)ESTART,ANEV,NDUMP
0159 C      15 FORMAT(F10.3,F10.0,I10)
0160 C      BINW=ESTART/100.
0161 C      WRITE(1,791)ESTART,ANEV,NDUMP
0162 C      791 FORMAT(2E12.5,I10)
0163 C      ELOW=ESTART-.01
0164 C      IF(ANEV.EQ.0.)GO TO 4000
0165 C
0166 C
0167 C
0168 C ***** SECTION C ** CHOOSE INITIAL POSITION X(I), DIRECTION
0169 C COSINES C(I), AND ENERGY E OF EACH PHOTON *****
0170 C NOTE THAT COORDINATE ORIGIN IS AT CENTER OF FRONT FACE OF CAM
0171 C
0172 C
0173 C*****START EVENT TRACKING
0174 C
0175 C      CALL EXEC(11,ITIME)
0176 C      WRITE(6,4001)ITIME(4),ITIME(3),ITIME(2)
0177 C      4001 FORMAT(///,5X,'START EVENT TRACKING AT TIME = ',2(I2,1H:),I2,///)
0178 C      2002 AIN=0.

```

```

0179      ISCUP=0
0180 2000 AIN=AIN+1.
0181      ISCUP=ISCUP+1
0182      COSTH=COSMAX* $\text{RAN}(\text{DMMY})+1.$ 
0183      SINTH= $\text{SQRT}(1.-\text{COSTH}*\text{COSTH})$ 
0184      PHI= $2\text{WOPI}*\text{RAN}(\text{DUMMY})$ 
0185      CX= $\text{SINTH}*\text{COS}(\text{PHI})$ 
0186      CY= $\text{SINTH}*\text{SIN}(\text{PHI})$ 
0187      CZ=COSTH
0188      RVEC=ZSORCE/COSTH
0189      X=RVEC*CX
0190      Y=RVEC*CY
0191      Z=0.001
0192      E=ESTART
0193      DO 40 I=1,6
0194 40 EL(I)=0.
0195 C
0196 C***CALL SUBROUTINE FOLGM, WHICH FOLLOWS THE GAMMA RAY UNTIL IT EITHER
0197 C ESCAPES FROM THE SYSTEM OR DROPS IN ENERGY BELOW 10 KEV
0198 C
0199 C FOLGM RETURNS THE ARRAY EL(I) WITH ELEMENTS I=1 TO 6 CONTAINING THE
0200 C LIGHT OUTPUT (IN MEV ELECTRON EQUIVALENT) IN EACH OF THE REACTION
0201 C CHANNELS 1 TO 6 LISTED ABOVE, AND THE TOTAL LIGHT OUTPUT IN EL(7).
0202 C IT ALSO RETURNS THE NUMBER OF INTERACTIONS MS DURING THE HISTORY.
0203 C
0204      CALL FOLGM(E,0.010,MS)
0205      IF(EL(4).LE.0.) GO TO 46
0206 C
0207 C
0208 C
0209 C ***** SECTION D ** SIMULATE FINITE PULSE HEIGHT RESOLUTION AND
0210 C UPDATE PULSE HEIGHT HISTOGRAMS *****
0211 C
0212 C
0213 C THE ARRAY ELEMENT EL(8) AND EL(9) CONTAIN THE TOTAL LIGHT OUTPUT
0214 C EL(7) FOR THE EVENT, SHIFTED RESPECTIVELY BY TWO GAUSSIAN-DISTRIBUTED
0215 C RANDOM NUMBERS WITH STANDARD ERRORS SIG WHICH DEPEND ON THE (INPUT
0216 C DATA) DETECTOR RESOLUTION VALUES ELOPE(2) AND ELOPE(3)
0217 C
0218      DO 44 J=2,3
0219      K=J+3
0220 C MAKE SURE TO PUT IN CORRECT SLOPE FOR THE RESOLUTION, IT IS NOW 0.38
0221      SIG=(RESSIG(J)*EL(4)/2.35482)*(0.662/EL(4))*0.38
0222 44 EL(K)=EL(4)+SIG* $\text{SQRT}(-2.*\text{ALOG}(\text{RAN}(\text{DMMY}))*\text{COS}(2\text{WOPI}*\text{RAN}(\text{DMMY})))$ 
0223 C
0224 C
0225 C THE ARRAY LT(I,J), I=1 TO 200, J=1 TO 9 CONTAINS 9 200-BIN HISTO-
0226 C GRAMS, EACH HAVING BIN WIDTH BINW (COMPUTED), ACCUMULATED FROM THE 9
0227 C ELEMENTS OF EL(J) RETURNED FOR EACH DETECTED PHOTON. THUS LT(I,J)
0228 C J=1 TO 6, CONTAINS A DIFFERENTIAL PULSE HEIGHT SPECTRUM (DPHS) FOR
0229 C REACTION CHANNEL J, ASSUMING PERFECT RESOLUTION. LT(I,J), J=7 TO 9,
0230 C CONTAINS A DPHS FOR THE TOTAL LIGHT OUTPUT, ASSUMING 1 PE LEVELS OF
0231 C ZERO, ELOPE(1), AND ELOPE(3) RESPECTIVELY.
0232 C
0233 C THE ARRAY LCON(J), J=1 TO 9, IS INCREMENTED EACH TIME ANY ELEMENT I
0234 C OF LT(I,J) IS INCREMENTED. IT THUS CONTAINS THE TOTAL IN EACH
0235 C HISTOGRAM.
0236 C THE ARRAY LHI(J), J=1 TO 9, CONTAINS THE BIN NUMBER I OF THE HIGHEST
0237 C NONZERO BIN OF LT(I,J)
0238 C

```

```

0239      46 DO 50 I=1,6
0240          L=IFIX(EL(I)/BINW+0.5)
0241          IF(L.LT.1)GO TO 50
0242          IF(L.GT.200)L=200
0243          LT(L,I)=LT(L,I)+1.
0244          LCON(I)=LCON(I)+1.
0245          LHI(I)=MAX0(LHI(I),L)
0246      50 CONTINUE
0247          IF(EL(4).GT.0.)ANDET=ANDET+1.
0248          IF(ISCUP.LT.1000)GO TO 795
0249          WRITE(1,902)AIN
0250      902 FORMAT(' FINISHED TRACKING',F7.0,' PHOTONS')
0251          ISCUP=0
0252      795 CONTINUE
0253          IF(AIN.GE.ANEV)GO TO 3000
0254          GO TO 2000
0255      C
0256      C
0257      C ***** SECTION E **OUTPUT *****
0258      C
0259      3000 CONTINUE
0260      C
0261      C*****EVENT TRACKING IS COMPLETE
0262      C
0263          CALL EXEC(11,ITIME)
0264          WRITE(6,4002)ITIME(4),ITIME(3),ITIME(2)
0265      4002 FORMAT(///,5X,'EVENT TRACKING COMPLETED AT TIME = ',
0266          1 2(I2,1H:),I2,///)
0267          WRITE(6,300) ANEV,ESTART
0268      300 FORMAT(1H1/F10.0,20H PHOTONS OF ENERGY ,F6.3,' MEV'/)
0269          WRITE(6,3007)
0270      3007 FORMAT(36H DIFFERENTIAL PULSE HEIGHT SPECTRA /)
0271          WRITE(6,3008)
0272      3008 FORMAT(40H LIGHT OUTPUT IN MEV ELECTRON EQUIVALENT)
0273          DO 350 I=1,3
0274              NC=LHI(I)
0275              WRITE(6,302)I,BINW,LCON(I)
0276              IF(LCON(I).EQ.0)GO TO 350
0277      302 FORMAT(9H0CHANNEL ,13,5H ONLY,5X,11H BIN WIDTH ,F6.3,6X,F8.0,
0278          1 8H EVENTS )
0279      350 WRITE(6,304)(LT(L,I),L=1,NC)
0280      304 FORMAT(10F7.0)
0281          DO 360 I=1,3
0282              K=I+3
0283              WRITE(6,306)RESSIG(I),BINW,LCON(K)
0284              NC=LHI(K)
0285              IF(I.NE.1) GO TO 360
0286              BUF(1)=ESTART
0287              BUF(2)=ANEV
0288              BUF(3)=ANDET
0289              CALL WRITF(IOUT,IERR,BUF,6)
0290              DO 975 L=1,NC
0291                  BUF(L)=LT(L,K)
0292      975 CONTINUE
0293              CALL WRITF(IOUT,IERR,BUF,2*NC)
0294              IF(IERR.LT.0) WRITE(1,904) IERR
0295      904 FORMAT(' WRITE ERROR',I6)
0296      C
0297      306 FORMAT(33H0LIGHT OUTPUT FROM ALL CHANNELS ,5X,
0298          1 28H DETECTOR RESOLUTION VALUE =,F7.3,/,

```

```

0299      2 11H BIN WIDTH ,F6.3,5H MEV ,6X,F7.0,8H EVENTS >
0300      360 WRITE(6,304)(LT(L,K),L=1,NC)
0301      GO TO 1000
0302      4000 CALL LOCF(IOUT,IERR,IREF,IRB,IOFF,JSEC)
0303      ITRUN=JSEC/2-IRB-1
0304      CALL CLOSE(IOUT,IERR,ITRUN)
0305      CALL CLOSE(INPUT)
0306      9900 STOP
0307      END
0308      C
0309      C
0310      C
0311      BLOCK DATA
0312      COMMON/BORDR/EP5
0313      COMMON/SCIN/ATDENS(2),RHO(2),IMAT
0314      COMMON/CONST/PI,TWOPI
0315      COMMON/CHECK/ESTART,EL0W
0316      COMMON/BNDS/XB,YB,ZB1,ZB2,XBB,YBB,ZBB2
0317      COMMON/SQUAR/XBSQ,XBBSQ
0318      COMMON/CORD/X,Y,Z,CX,CY,CZ
0319      COMMON/GEOM/IGEO,NCT(50),NEG
0320      COMMON/DUMP/NDUMP,MDUMP,EL(9)
0321      COMMON/ANIHL/XANN,YANN,ZANN,CXANN,CYANN,CZANN,IMATAN
0322      COMMON/VAR/B,B2,B3,A,A2,G,G2,G3,U
0323      COMMON/VARPR/ANEW,PIB2,PIB4,UNEW
0324      COMMON/RNDM/X1,X2
0325      DOUBLE PRECISION X1,X2
0326      C NEXT CARD IS FOR RUNNING WITH IRON CLADDING.
0327      C DATA RHO/3.667,7.86/
0328      C NEXT CARD IS FOR RUNNING WITH ALUMINUM CLADDING
0329      C DATA RHO/3.667,2.70/
0330      C****ATOMIC DENSITIES ARE IN 10**24 ATOMS/BARN-CM
0331      C FOR MAI, ATDENS(1)=0.602252*RHO(1)/(22.9898+126.9044)
0332      C DATA ATDENS(1)/1.47334E-2/
0333      C FOR ALUMINUM, ATDENS(2)=0.602252*RHO(2)/26.98
0334      C FOR IRON ATDENS(2)=0.602252*RHO(2)/55.847
0335      C DATA ATDENS(2)/6.02698E-2/
0336      C DATA PI,TWOPI/3.14159,6.28318/
0337      C DATA PIB2,PIB4/1.57079,0.785398/
0338      C DATA EPS/0.0001/
0339      C DATA NCT/50*0/
0340      C DATA NEG/0/
0341      C DATA NDUMP/0/
0342      END
0343      END*

```

&RAN T=00004 IS ON CR00026 USING 00004 BLKS R=0000

```
0001 FTH4,L
0002 FUNCTION RAN(U)
0003 C GENERATE A NON-ZERO RANDOM NUMBER BETWEEN 0. AND 1.0
0004 COMMON/RNDM/X1,X2
0005 DOUBLE PRECISION E,X,X1,X2
0006 INTEGER IX(4)
0007 EQUIVALENCE (X,IX)
0008 DATA E/2.7182818284590452D+00/
0009 C TRANSLATE FORWARD
0010 100 X1=X1+1.0D+00
0011 X2=X2+E
0012 C SINE SUPERPOSITION
0013 X=DSIN(X1)+DSIN(X2)
0014 C CONVERT TO RANDOM NUMBER
0015 U=IX(3)/65536+.5
0016 IF(U.EQ.0.)GO TO 100
0017 RAN=U
0018 C RETURN TO CALLING ROUTINE
0019 RETURN
0020 END
0021 SUBROUTINE INIT
0022 C INITIALIZE RANDOM NUMBER GENERATOR
0023 COMMON/RNDM/X1,X2
0024 DOUBLE PRECISION X1,X2
0025 DIMENSION ITIME(5)
0026 DATA ICODE/11/
0027 C PICK UP TIME OF DAY IN THOUSAND SECONDS
0028 CALL EXEC(ICODE,ITIME)
0029 X1=3600.*ITIME(4)+60.*ITIME(3)+ITIME(2)+.01*ITIME(1)
0030 X1=.001*X1
0031 X2=X1
0032 C RETURN TO CALLING ROUTINE
0033 RETURN
0034 END
```


&SIGTT T=00004 IS ON CR00026 USING 00013 BLKS R=0000

```

0001 FTN4,L
0002 SUBROUTINE SIGTT(EMEV,S,IMAT)
0003 C J=1 INCOHERENT COMPTON SCATTERING
0004 C J=2 PHOTOELECTRIC EFFECT
0005 C J=3 PAIR PRODUCTION
0006 C
0007 DIMENSION S(8),A1BAR(8),A2BAR(8),A3BAR(8),A4BAR(8)
0008 DIMENSION CPP(2),APP(14),BATON(2)
0009 DATA BATON/248.9115,44.8049/
0010 C DATA BATON/248.9115,92.7383/
0011 C***** 1971 BIGGS AND LIGHTHILL PARAMETERIZATION
0012 C NEXT 8 CARDS ARE FOR IRON
0013 C DATA A1BAR/-2.168E1,-3.065E0,1.884E0,9.303E-1,2.151E-1,2.151E-1,
0014 C * 3.511E-1,1.107E-1/
0015 C DATA A2BAR/2.201E3,1.572E3,1.555E2,1.487E3,-2.680E2,-2.680E2,
0016 C * -1.633E2,1.252E2/
0017 C DATA A3BAR/1.373E5,1.458E6,1.593E6,9.772E5,2.341E5,2.341E5,
0018 C * 2.090E5,1.008E5/
0019 C DATA A4BAR/-2.150E5,-1.393E7,-1.821E7,7.620E7,-6.063E5,-6.063E5,
0020 C * 7.131E5,1.273E7/
0021 C NEXT 8 CARDS ARE FOR ALUMINUM
0022 C DATA A1BAR/-2.168E1,-3.065E0,1.884E0,9.303E-1,4.158E-1,4.158E-1,
0023 C * 1.125E-2,9.245E-3/
0024 C DATA A2BAR/2.201E3,1.572E3,1.555E2,1.487E3,-1.351E2,-1.351E2,
0025 C * 2.747E0,1.188E1/
0026 C DATA A3BAR/1.373E5,1.458E6,1.593E6,9.772E5,2.716E4,2.716E4,
0027 C * 1.174E4,4.104E3/
0028 C DATA A4BAR/-2.150E5,-1.393E7,-1.821E7,7.620E7,3.723E2,3.723E2,
0029 C * 5.695E5,2.352E6/
0030 C DATA BIND/1022.012/
0031 C DATA CPP/3.072E-9,7.336E-10/
0032 C DATA CPP/3.072E-9,1.508E-9/
0033 C DATA APP/-7.960E-5,4.421E-6,5.721E-8,-1.356E-9,8.531E-12,
0034 C*****NEXT TWO CARDS ARE FOR RUNNING WITH ALUMINUM
0035 C * -1.174E-14,3.800E-17,5.930E-6,-1.598E-6,1.059E-7,-1.528E-9,
0036 C * 9.374E-12,-2.460E-14,3.460E-17/
0037 C NEXT TWO CARDS ARE FOR RUNNING WITH IRON CLADDING
0038 C * -1.714E-14,3.800E-17,-6.900E-6,-1.309E-6,1.500E-7,-2.235E-9,
0039 C * 1.358E-11,-3.410E-14,5.030E-17/
0040 C CALL CMPST(0,EMEV,SIGT,DUM,DUM,DUM,DUM)
0041 C IF(IMAT.GT.1) GO TO 5
0042 C BDCORR=1.-.492123*EXP(-EMEV/.00879656)-.449453*EXP(-EMEV/.0389331)
0043 C * -.0889303*EXP(-EMEV/.212995)
0044 C S(1)=SIGT*64.*BDCORR
0045 C GO TO 6
0046 C*****NEXT THREE CARDS ARE FOR RUNNING WITH ALUMINUM CLADDING
0047 C 5 BDCORR=1.-.456609*EXP(-EMEV/.00572223)-.479465*EXP(-EMEV/.0177749)
0048 C * -.0849173*EXP(-EMEV/.0918095)
0049 C S(1)=SIGT*13.*BDCORR
0050 C NEXT THREE CARDS ARE FOR RUNNING WITH IRON CLADDING
0051 C 5 BDCORR=1.-.611797*EXP(-EMEV/.00894953)-.349226*EXP(-EMEV/.0352814)
0052 C * -.0664580*EXP(-EMEV/.197318)
0053 C S(1)=SIGT*26.*BDCORR
0054 C 6 CONTINUE
0055 C E=EMEV*1000.
0056 C J=1
0057 C IF(E.GT.33.170) J=2
0058 C IF(E.GT.100.) J=3

```

```

0059      IF(E .GT. 500.) J=4
0060      J=J+IMAT*4-4
0061      E1=1./E
0062      E2=E1/E
0063      E3=E2/E
0064      E4=E3/E
0065      S(2)=A1BAR(J)*E1+A2BAR(J)*E2+A3BAR(J)*E3+A4BAR(J)*E4
0066      S(2)=S(2)*BATON(IMAT)
0067 C      THIS TEST PUT IN JUST TO ENSURE POSITIVE S(2) AT LOW ENERGIES
0068      IF(S(2).LT.0.0)S(2)=ABS(S(2))
0069      IF(E .GT. BIND) GO TO 10
0070      S(3)=0.
0071      RETURN
0072      10 IF(E .GT. 1500.) GO TO 20
0073      S(3)=BATON(IMAT)*CPP(IMAT)*(E-BIND)**2
0074      RETURN
0075      20 V=SQRT(E-BIND)
0076      V2=V*V
0077      V3=V2*V
0078      V4=V3*V
0079      V5=V4*V
0080      V6=V5*V
0081      V7=V6*V
0082      K=IMAT*7-6
0083      S(3)=APP(K)*V+APP(K+1)*V2+APP(K+2)*V3+APP(K+3)*V4+APP(K+4)*V5+
0084      * APP(K+5)*V6+APP(K+6)*V7
0085      S(3)=BATON(IMAT)*S(3)/(1.+1.7E-13*V6)
0086      RETURN
0087      END

```

&CMPST T=00004 IS ON CR00026 USING 00008 BLKS R=0000

```

0001  FTN4,L
0002      SUBROUTINE CMPST(NTRY,EGAM,SIGT,ELT,EGAMP,COSGAM,PHI).
0003      COMMON/CONST/PI,TWOPI
0004      COMMON/CHECK/ESTART,ELOW
0005      COMMON/COORD/X,Y,Z,CX,CY,CZ
0006  C
0007  C  SIMULATES THE COMPTON SCATTERING OF A GAMMA OF ENERGY EGAM BY AN
0008  C  ELECTRON. FOR NTRY=0,CMPST RETURNS ONLY THE TOTAL COMPTON CROSS-
0009  C  SECTION SIGT (BARNS) PER ELECTRON AT ENERGY EGAM. IF NTRY IS NONZERO
0010  C  CMPST ALSO CHOOSES BY MONTE CARLO THE RECOIL ELECTRON ENERGY EEL,
0011  C  THE NEW GAMMA ENERGY EGAMP,AND THE COSINE COSGAM OF THE LAB SCAT-
0012  C  TERING ANGLE OF THE GAMMA. SEE THE PROGRAM DESCRIPTION FOR MORE DETAI
0013  C
0014  C  COMPUTE TOTAL CROSS SECTION
0015  C
0016      EPS=.001
0017      G=EGAM/.5110
0018      G4=(G*G)**2
0019      A=1.+2.*G
0020      TH=2.*G*G/A
0021      B=(1.+G)/G**2
0022      S=B*(2.*(1.+G)/A -ALOG(A)/G) +ALOG(A)*.5/G -(1.+3.*G)/A**2
0023      SIGT=0.4989*S
0024      IF (NTRY .EQ. 0) RETURN
0025  C
0026  C  CHOOSE THE RECOIL ELECTRON ENERGY BY MONTE CARLO.
0027  C  P IS THE INTEGRAL DISTRIBUTION OF THE ELECTRON ENERGY E (IN UNITS
0028  C  OF THE ELECTRON REST MASS). THE EQUATION P(E)-U=0 IS SOLVED BY
0029  C  ITERATION,WHERE U IS A UNIFORMLY-DISTRIBUTED RANDOM VARIABLE.
0030  C
0031      U=RAN(DMMY)
0032      C=0.
0033      D=TH
0034      6 D=.5*D
0035      E=C+D
0036      F=G/(G-E)
0037      FL=ALOG(F)/G
0038      P=( B*(B*.5*E-FL) +.5*FL +.5*E*(F-E*G*.5)/G4)/S
0039      IF (ABS(U-P)-EPS) 20,20,10
0040      10 IF (U-P) 6,20,12
0041      12 C=E
0042      GO TO 6
0043      20 EEL=0.5110*E
0044      EGAMP=EGAM-EEL
0045      COSGAM=1.-(EGAM/EGAMP-1.)/G
0046      IF(COSGAM .GE. 1.0) COSGAM=0.999999
0047      PHI=TWOPI*RAN(DMMY)
0048      COSEL=1./SQRT(1.+(2./(1.-COSGAM)-1.)/(1.+G)**2)
0049      IF(COSEL .GT. 1.0) COSEL=1.0
0050  C*****CALL SUBROUTINE ELECT THAT TRACES THE ELECTRON PATH TO DETERMINE
0051  C*****HOW MUCH OF THE ELECTRON ENERGY IS DEPOSITED IN THE CRYSTAL
0052      PFLAG=-1.0
0053      CALL ELECT(COSEL,PHI,EEL,ELT,PFLAG)
0054      CALL SCATR(COSGAM,PHI)
0055      RETURN
0056  END

```

&PHOTO T=00004 IS ON CR00026 USING 00006 BLKS R=0000

```

0001  FTN4,L
0002      SUBROUTINE PHOTO(E,ELT,EGAM)
0003      COMMON/VAR/B,B2,B3,A,A2,G,G2,G3,U
0004      COMMON/CONST/PI,TWOPI
0005      COMMON/CHECK/ESTART,ELOW
0006      T=E/0.5110
0007      B=SQRT(T*(T+2.))/(T+1.)
0008      B2=B*B
0009      B3=B2*B
0010      A=1.-B
0011      A2=A*A
0012      G2=1.-B2
0013      G=SQRT(G2)
0014      G3=G2*G
0015      U=РАН(DMMY)
0016      DO 10 I=1,20
0017      XL=1.-(I-1)*0.1
0018      XU=XL-0.1
0019      FL=PEANG(XL)
0020      FU=PEANG(XU)
0021      SIGN=FL*FU
0022      IF(SIGN) 1,2,10
0023  10 CONTINUE
0024      1 XMID=(XL+XU)*0.5
0025      FMID=PEANG(XMID)
0026      IF(FL*FMID) 3,4,5
0027      3 XU=XMID
0028      GO TO 6
0029      4 COSEL=XMID
0030      GO TO 7
0031      5 XL=XMID
0032      6 IF((XL-XU).GT. 0.001) GO TO 1
0033      COSEL=(XL+XU)*0.5
0034      7 PHI=TWOPI*РАН(DMMY)
0035      EGAM=0.
0036      PFLAG=-1.0
0037      CALL ELECT(COSEL,PHI,E,ELT,PFLAG)
0038      RETURN
0039      2 COSEL=XL
0040      IF(FU.EQ. 0.) COSEL=XU
0041      GO TO 7
0042      END
0043      FUNCTION PEANG(COSTH)
0044      COMMON/VAR/B,B2,B3,A,A2,G,G2,G3,U
0045      DEL=1.-B*COSTH
0046      DEL2=DEL*DEL
0047      DEL3=DEL2*DEL
0048      DEL4=DEL3*DEL
0049      TOP=G*((3.*B2*COSTH*COSTH-3.*B*COSTH+G2)/(3.*B3*DEL3)-(1.-2.*B)/
0050      * (3.*B3*A2))
0051      FACT=(G2-2.*G+1.)/(2.*G2)
0052      TOP=TOP+FACT*((B2-4.*B+3.)/(2.*B3*A2)-(B2+3.-4.*B*COSTH)/
0053      * (2.*B3*DEL2)-ALOG(DEL/A)/B3)
0054      BOT=4./(3.*G3)+FACT*(2./(B2*G2)-ALOG((1.+B)/A)/B3)
0055      PEANG=TOP/BOT-U
0056      RETURN
0057      END

```

&PAIR T=00004 IS ON CR00026 USING 00012 BLKS R=0000

```

0001 FTH4,L
0002 SUBROUTINE PAIR(K,ELT,EGAM,GFLAG,COSTH1,PHI1,COSTH2,PHI2)
0003 COMMON/CHECK/ESTART,ELOW
0004 COMMON/VARPR/A,PIB2,PIB4,U
0005 COMMON/CONST/PI,TWOPI
0006 COMMON/CORD/X,Y,Z,CX,CY,CZ
0007 COMMON/ANIHL/XANN,YANN,ZANN,CXANN,CYANN,CZANN,IMATAN
0008 COMMON/SCIN/ATDEHS(2),RHO(2),IMAT
0009 LOGICAL GFLAG
0010 REAL K,MU,MU2,L
0011 DATA MU,MU2/0.511006,0.261127/
0012 C*****FIND THE POSITRON AND ELECTRON KINETIC ENERGIES
0013 E=0.5*K
0014 EKK2=K*K
0015 E2=E*E
0016 P2=E2-MU2
0017 P=SQRT(P2)
0018 EPS=2.*ALOG((E+P)/MU)
0019 L=2.*ALOG(2.*E2/(MU*K))
0020 P3=P2*P
0021 P4=P3*P
0022 A=-1.33333*P2-4.*E2+MU2*(2.*EPS*E/P-EPS*EPS)
0023 FACTL=EKK2*(E2*E2+P4)/P4-2.66667*E2-(MU2*K*0.5)*
0024 * (2.*MU2*EPS/P3+2.*K*E2/P4)
0025 A=(A+L*FACTL)/(K*K*K)
0026 U=РАН(DMMY)
0027 DO 10 I=1,20
0028 XL=(I-1)*0.05
0029 XU=XL+0.05
0030 FL=PAIRN(XL)
0031 FU=PAIRN(XU)
0032 SIGN=FL*FU
0033 IF(SIGN)1,2,10
0034 10 CONTINUE
0035 1 XMID=(XL+XU)*0.5
0036 FMID=PAIRN(XMID)
0037 IF(FL*FMID)3,4,5
0038 3 XU=XMID
0039 GO TO 6
0040 4 ERGPOS=XMID
0041 GO TO 7
0042 5 XL=XMID
0043 6 IF((XU-XL).GT.0.0005) GO TO 1
0044 ERGPOS=(XL+XU)*0.5
0045 GO TO 7
0046 2 ERGPOS=XL
0047 IF(FU.EQ.0.) ERGPOS=XU
0048 7 CONTINUE
0049 TWOMU=2.*MU
0050 TPOS=(K-TWOMU)*ERGPOS
0051 TNEG=K-TWOMU-TPOS
0052 C*****FIND THE POSITRON AND ELECTRON SCATTERING ANGLES
0053 EPOS=TPOS*MU
0054 PPOS=SQRT(EPOS*EPOS-MU2)
0055 COSTHP=((EPOS+(PPOS-EPOS))/(1.-2.*PPOS*РАН(DMMY)/(EPOS+PPOS)))/PPOS
0056 ENEG=TNEG*MU
0057 PNEG=SQRT(ENEG*ENEG-MU2)
0058 COSTHN=((ENEG+(PNEG-ENEG))/(1.-2.*PNEG*РАН(DMMY)/(ENEG+PNEG)))/PNEG

```

```

0059      PHIPOS=TWOPI*RAN(DMMY)
0060      PHINEG=TWOPI*RAN(DMMY)
0061 C*****TRACK THE ELECTRON AND THEN THE POSITRON
0062      PFLAG=-1.0
0063      CALL ELECT(COSTHN,PHINEG,TNEG,ELTNEG,PFLAG)
0064      ELT=ELT+ELTNEG
0065      PFLAG=1.0
0066      CALL ELECT(COSTHP,PHIPOS,TPOS,ELTPOS,PFLAG)
0067      ELT=ELT+ELTPOS
0068      IF(PFLAG.GT. 1.) GO TO 9
0069 C*****START THE TWO ANNIHILATION PHOTONS (INSIDE THE SYSTEM)
0070 C*****CHOOSE THE SCATTERING ANGLES OF THE TWO PHOTONS
0071      XANN=X
0072      YANN=Y
0073      ZANN=Z
0074      CXANN=CX
0075      CYANN=CY
0076      CZANN=CZ
0077      IMATAN=IMAT
0078      COSTH1=2.*RAN(DMMY)-1.
0079      COSTH2=-COSTH1
0080      PHI1=PI*RAN(DMMY)
0081      PHI2=PHI1+PI
0082 C*****TRACK EACH PHOTON INDEPENDENTLY
0083      EGAM=MU
0084      CALL SCATR(COSTH1,PHI1)
0085      GFLAG=.TRUE.
0086      RETURN
0087 9 EGAM=0.
0088      GFLAG=.FALSE.
0089      RETURN
0090      END
0091      FUNCTION PAIRN(X)
0092      COMMON/VARPR/A,PIB2,PIB4,U
0093      X2=X*X
0094      X3=X2*X
0095      X4=X3*X
0096      X5=X4*X
0097      ARG=2.*X-1.
0098      TOP=0.5*ARG*SQRT(X-X2)+0.25*(SIN(ARG)+PIB2)
0099      TOP=TOP+0.540*(A-0.52)*(0.5*X2-1.66667*X3+2.*X4-0.8*X5)
0100      BOT=PIB4+0.540*(A-0.52)/30.
0101      PAIRN=TOP/BOT-U
0102      RETURN
0103      END

```

&ELECT T=00004 IS ON CR00026 USING 00013 BLKS R=0000

```

0001  FTN4,L
0002      SUBROUTINE ELECT(COSEL,PHI,E,ELT,PFLAG)
0003      COMMON/CORD/X,Y,Z,CX,CY,CZ
0004      COMMON/SCIN/ATDENS(2),RHO(2),IMAT
0005      COMMON/GEOM/IGEO,NCT(50),NEG
0006      COMMON/DUMP/NDUMP,MDUMP,EL(9)
0007      COMMON/BORDR/EPS
0008  C*****SAVE COORDINATES AND DIRECTION COSINES FOR LATER GAMMA TRACING
0009      XOLD=X
0010      YOLD=Y
0011      ZOLD=Z
0012      CXOLD=CX
0013      CYOLD=CY
0014      CZOLD=CZ
0015      IMATOL=IMAT
0016      CALL SCATR(COSEL,PHI)
0017  C*****CALCULATE THE RANGE OF AN ELECTRON OF ENERGY E IN THE MEDIUM
0018      1 RANGE=(0.412*E**((1.265-0.0954*ALOG(E)))/RHO(IMAT)
0019      CALL GEOMT(INOUT,RINT,RANGE)
0020      IF(NDUMP.GT. MDUMP) GO TO 10
0021      WRITE(6,100) E,EL(4),MS,ELT,X,Y,Z,CX,CY,CZ,COSEL,PHI,RINT,RANGE,
0022      * INOUT,IMAT
0023      100 FORMAT (F6.3,1X,F6.2,13,F6.2,1X,3F8.4,1X, 3F8.4,1X,2F7.4,2F9.5,
0024      * 2I5,4X,1HE)
0025      10 CONTINUE
0026      IF(RINT.LT. 0.) NEG=NEG+1
0027      IF(IMAT.GT. 1) GO TO 2
0028  C*****ELECTRON ORIGINATED IN THE CRYSTAL
0029      IF(INOUT.GT. 1) GO TO 3
0030  C*****ELECTRON LOST ALL ITS ENERGY IN THE CRYSTAL
0031      ELT=E
0032      IF(PFLAG.GT. 0.) GO TO 5
0033      GO TO 1000
0034  C*****ELECTRON LOST PART OF ITS ENERGY IN THE CRYSTAL AND THEN
0035  C*****ENTERED THE CAN
0036      3 RANGEP=(RANGE-RINT)*RHO(IMAT)
0037      IF(RANGEP.LT.EPS)RANGEP=EPS
0038      POWER=(1.265-SQRT(1.60022+0.38160*(-0.886732-ALOG(RANGEP))))/
0039      * 0.19080
0040      EP=EXP(POWER)
0041      IF(INOUT.GT. 2) GO TO 7
0042      ELT=E-EP
0043      IF(ELT.LT.0.0)ELT=0.0
0044      IF(PFLAG.LT. 0.) GO TO 1000
0045      7 RINTP=RINT+EPS
0046      X=X+CX*RINTP
0047      Y=Y+CY*RINTP
0048      Z=Z+CZ*RINTP
0049      E=EP
0050      IMAT=2
0051      GO TO 1
0052  C*****ELECTRON ORIGINATED IN THE CAN
0053      2 IF(INOUT.EQ. 2) GO TO 4
0054      ELT=0.
0055      IF(PFLAG.LT. 0.) GO TO 1000
0056      IF(INOUT.LT. 1) GO TO 6
0057      IF(INOUT.LT. 2) GO TO 5
0058      GO TO 3

```

```

0059 C*****ELECTRON LOST PART OF ITS ENERGY IN THE CAN AND THEN ENTERED THE C
0060 4 RANGEP=(RANGE-RINT)*RHO(IMAT)
0061 IF(RANGEP.LT.EPS)RANGEP=EPS
0062 POWER=(1.265-SQRT(1.60022+0.38160*(-0.886732-ALOG(RANGEP))))/
0063 * 0.19080
0064 E=EXP(POWER)
0065 X=X+CX*(RINT+EPS)
0066 Y=Y+CY*(RINT+EPS)
0067 Z=Z+CZ*(RINT+EPS)
0068 IMAT=1
0069 GO TO 1
0070 1000 CONTINUE
0071 C*****RESTORE GAMMA COORDINATES AND DIRECTION COSINES
0072 X=XOLD
0073 Y=YOLD
0074 Z=ZOLD
0075 CX=CXOLD
0076 CY=CYOLD
0077 CZ=CZOLD
0078 IMAT=IMATOL
0079 RETURN
0080 5 X=X+CX*RANGE
0081 Y=Y+CY*RANGE
0082 Z=Z+CZ*RANGE
0083 RETURN
0084 6 PFLAG=2.
0085 RETURN
0086 END

```


LSCATR T=00004 IS ON CR00026 USING 00007 BLKS R=0000

```

0001 FTH4,L
0002 SUBROUTINE SCATR(COSTH,PHI)
0003 C
0004 C GIVEN A PARTICLE MOVING IN A DIRECTION SPECIFIED BY DIRECTION COSINES
0005 C C(I), I=1 TO 3 ,WHICH SCATTERS BY ANGLES ARCCOS(COSTH)=TH AND PHI WITH
0006 C RESPECT TO THE INITIAL DIRECTION. SCATR FINDS THE NEW DIRECTION
0007 C COSINES CNEW(I) AS FOLLOWS --
0008 C
0009 C THE OLD DIRECTION OF THE PARTICLE MAY BE DESCRIBED BY THE ANGLES
0010 C T AND P ,MEASURED FROM THE POSITIVE Z- AND X-AXES RESPECTIVELY.
0011 C THEN CT=COS(T)=C(3), SP=SIN(P)=C(2)/SIN(T),CP=COS(P)=C(1)/SIN(T).
0012 C NOW CONSIDER A COORDINATE SYSTEM WHICH HAS ITS POSITIVE Z-AXIS ALONG
0013 C THE ORIGINAL DIRECTION OF THE PARTICLE AND ITS X-AXIS ALONG THE
0014 C X-AXIS OF THE MAIN COORDINATE SYSTEM. IN THIS FRAME THE NEW DIRECTION
0015 C COSINES AFTER THE SCATTER ARE CXP=SIN(TH)*COS(PHI),
0016 C CYP=SIN(TH)*SIN(PHI) , CZP=COS(TH). ROTATING BY THE ANGLES T AND P
0017 C WILL PUT THE NEW DIRECTION COSINES BACK INTO THE MAIN COORDINATE
0018 C SYSTEM.
0019 C
0020 COMMON/CORD/X,Y,Z,CX,CY,CZ
0021 C PROTECT THE ANGLES
0022 CT=CZ
0023 IF (CT-1.)>6,4,2
0024 2 CT=1.
0025 4 ST=0.
0026 SP=0.
0027 CP=1.
0028 GO TO 30
0029 6 IF (-CT-1.)>20,4,8
0030 8 CT=-1.
0031 GO TO 4
0032 20 ST=SQRT(1.-CT*CT)
0033 CP=CX/ST
0034 IF (CP .GT. 1.) CP=1.
0035 IF (CP .LT. -1.) CP=-1.
0036 SP=CX/ST
0037 IF (SP .GT. 1.) SP=1.
0038 IF (SP .LT. -1.) SP=-1.
0039 30 IF (COSTH .GT. 1.) COSTH=1.
0040 IF (COSTH .LT. -1.) COSTH=-1.
0041 S=SQRT(1.-COSTH*COSTH)
0042 CXP= S*COS(PHI)
0043 CYP= S*SIN(PHI)
0044 CZP=COSTH
0045 CX=CXP*CT*CP-CYP*SP+CZP*ST*CP
0046 CY=CXP*CT*SP+CYP*CP+CZP*ST*SP
0047 CZ=-CXP*ST+CZP*CT
0048 RETURN
0049 END

```

&FOLGM T=00004 IS ON CR00026 USING 00017 BLKS R=0000

```

0001 FTH4,L
0002 SUBROUTINE FOLGM(E,EMIN,MS)
0003 C
0004 C MONTE CARLO PHOTON TRACER
0005 C
0006 C GIVEN A PHOTON OF ENERGY E0 AT POSITION X(I), I=1,3,
0007 C AND MOVING ALONG DIRECTION COSINE C(I), I=1 TO 3,
0008 C FOLGM FOLLOWS THE PHOTON THROUGH SUCCESSIVE INTERACTIONS
0009 C UNTIL IT EITHER ESCAPES THE INTERACTION VOLUME, OR IS ABSORBED,
0010 C OR UNTIL ITS ENERGY DROPS BELOW EMIN, OR MORE THAN 10 SCATTERS OCCUR.
0011 C
0012 C
0013 C FOLGM RETURNS EL(L), L=1,7, AN ARRAY CONTAINING THE LIGHT OUTPUT
0014 C IN EACH OF THE SIX REACTION CHANNELS (SEE BELOW), AND THE TOTAL
0015 C LIGHT OUTPUT (L=7). LIGHT OUTPUT IS EXPRESSED IN MEV ELECTRON EQUIVAL.
0016 C FOLGM ALSO RETURNS MS, THE NUMBER OF INTERACTIONS DURING THE HISTORY.
0017 C
0018 C
0019 C DIMENSION S(8)
0020 C COMMON/SCIN/ATDENS(2),RH0(2),IMAT
0021 C COMMON/CONST/PI,TWOPI
0022 C COMMON/BNDS/XB,YB,ZB1,ZB2,XBB,YBB,ZBB2
0023 C COMMON/GEOM/IGEO,NCT(50),NEG
0024 C COMMON/COORD/X,Y,Z,CX,CY,CZ
0025 C COMMON/DUMP/NDUMP,MDUMP,EL(9)
0026 C COMMON/BORDR/EPS
0027 C COMMON/ANIHL/XANN,YANN,ZANN,CXANN,CYANN,CZANN,IMATAN
0028 C LOGICAL GFLAG
0029 C REAL MU
0030 C GFLAG=.FALSE.
0031 C DATA MU/0.511006/
0032 C
0033 C INITIALIZE
0034 C MS=0
0035 C NS=0
0036 C ELT=0.
0037 C DO 4 I=1,4
0038 C 4 EL(I)=0.
0039 C
0040 C PREPARE TO PRINT INDIVIDUAL HISTORIES, IF DESIRED.
0041 C IF(NDUMP.GT.MDUMP)GO TO 2001
0042 C NDUMP=NDUMP+1
0043 C IF(MDUMP.EQ.0)GO TO 2001
0044 C IF(NDUMP.LT.2) WRITE(6,9)
0045 C 9 FORMAT(1H0,2X,'DUMP OF THE FIRST MDUMP EVENT HISTORIES',//,4X,
0046 C A 'E',3X,'EL(4)',1X,'MS',3X,'ELT',6X,'X',7X,'Y',7X,'Z',7X,
0047 C B 'CX',6X,'CY',6X,'CZ',6X,'COS',5X,'PHI',4X,'RINT',6X,'D',3X,
0048 C C 'INOUT',2X,'IMAT',2X,'TYPE')
0049 C IF(NDUMP.GT.MDUMP)GO TO 2001
0050 C WRITE(6,11)NDUMP,E
0051 C 11 FORMAT(7H0EVENT ,15,5X,14HINITIAL ENERGY ,F8.3,4H MEV)
0052 C
0053 C FIND TOTAL CROSS SECTIONS.
0054 C
0055 C
0056 C THE ARRAY S(I) RETURNED BY SIGTOT CONTAINS THE TOTAL CORSS-SECTIONS
0057 C AT ENERGY E FOR EACH REACTION CHANNEL.
0058 C

```

```

0059 C
0060 2001 IMAT=2
0061 26 CALL SIGTT(E,S,IMAT)
0062 PATH=1./(<ATDENS(IMAT)*(<S(1)+S(2)+S(3)>>)
0063 D=-PATH*ALOG(RAN(DMMY))
0064 IF(D.LT.EPS)D=EPS
0065 CALL GEOMT(INOUT,RINT,D)
0066 IF(NDUMP.GT.NDUMP)GO TO 101
0067 WRITE(6,1)E,EL(4),MS,ELT,X,Y,Z,CX,CY,CZ,COSGAM,PHI,RINT,D,INOUT,
0068 * IMAT
0069 1 FORMAT(F6.3,1X,F6.2,I3,F6.2,1X,3F8.4,1X,3F8.4,1X,2F7.4,2F9.5,
0070 * 2I5,4X,1MG)
0071 101 CONTINUE
0072 IF(RINT.LT.0.) NEG=NEG+1
0073 IF(INOUT.LT.1) GO TO 1002
0074 IF(INOUT.LT.2) GO TO 21
0075 IF(INOUT.GT.2) GO TO 24
0076 GO TO (22,23) IMAT
0077 22 IMAT=2
0078 GO TO 24
0079 23 IMAT=1
0080 24 X=X+CX*(RINT+EPS)
0081 Y=Y+CY*(RINT+EPS)
0082 Z=Z+CZ*(RINT+EPS)
0083 GO TO 26
0084 21 MS=MS+1
0085 X=X+CX*D
0086 Y=Y+CY*D
0087 Z=Z+CZ*D
0088 C.....SET ELT=0.0 BEFORE GOING TO SUBROUTINES
0089 ELT=0.0
0090 C
0091 C*****DECIDE WHICH INTERACTION OCCURRED
0092 C
0093 SCTRND=RAN(DMMY)
0094 PROBCP=PATH*ATDENS(IMAT)*S(1)
0095 IF(SCTRND.GT.PROBCP) GO TO 200
0096 MS=1
0097 CALL CMPST(1,E,SIGT,ELT,EGAM,COSGAM,PHI)
0098 GO TO 1000
0099 200 PROBPE=PATH*ATDENS(IMAT)*S(2)+PROBCP
0100 IF(SCTRND.GT.PROBPE) GO TO 300
0101 MS=2
0102 CALL PHOTO(E,ELT,EGAM)
0103 GO TO 1000
0104 300 MS=3
0105 IF(E.LT.1.555)GO TO 26
0106 CALL PAIR(E,ELT,EGAM,GFLAG,COSTH1,PHI1,COSTH2,PHI2)
0107 COSGAM=COSTH1
0108 PHI=PHI1
0109 C
0110 C INCREMENT THE ELEMENT MS=1 TO 6 OF THE LIGHT OUTPUT ARRAY ELT
0111 C CORRESPONDING TO THE CHANNEL OF INTERACTION, AND ELT(7),THE TOTAL
0112 C LIGHT OUTPUT.
0113 C
0114 1000 EL(MS)=EL(MS)+ELT
0115 EL(4)=EL(4)+ELT
0116 E=EGAM
0117 2000 CONTINUE
0118 IF(E.GT.EMIN) GO TO 26

```

```

0119      IF(E.LT.0.0001) GO TO 1001
0120      PH=E
0121      EL(2)=EL(2)+PH
0122      EL(4)=EL(4)+PH
0123 1001 IF(NDUMP.GE.NDUMP) GO TO 1002
0124      WRITE(6,1)E,EL(4),MS,ELT,X,Y,Z,CX,CY,CZ,COSGAM,PHI,RINT,D,INOUT,
0125      * IMAT
0126 1002 IF(GFLAG)GO TO 1003
0127      RETURN
0128 1003 GFLAG=.FALSE.
0129      E=MU
0130      X=XANH
0131      Y=YANH
0132      Z=ZANH
0133      CX=CXANH
0134      CY=CYANH
0135      CZ=CZANH
0136      IMAT=IMATAN
0137      CALL SCATB(COSTH2,PHI2)
0138      COSGAM=COSTH2
0139      PHI=PHI2
0140      GO TO 26
0141      END

```

GEOMT 1=0004 IS ON CR00026 USING 00027 BLKS R=0000

```

0001 FTN4,L
SUBROUTINE GEOMT(INOUT,RINT,D)
COMMON/BNDS/XB,YB,ZB1,ZB2,XB8,YB8,ZB82
COMMON/SQUAR/EXBSQ,EXBSQ
COMMON/CORD/X,Y,Z,CX,CY,CZ
COMMON/GEOM/IGEO,NCT(50),NEG
COMMON/SCIN/ATDENS(2),RHO(2),IMAT
COMMON/DUMP/NDUMP,MDUMP,EL(9)
COMMON/BORDR/EPS
DATA EPS/0.00001/
IF(CX.EQ.0.)CX=0.001
IF(CY.EQ.0.)CY=0.001
F=CX+CX+Y+CY
G=X+CX+Y+CY
IF(IGEO.NE.2) GO TO 701
XBSQ=EXBSQ
YBSQ=EXBSQ
ZBSQ=EXBSQ
GO TO 705
701 TPI=CX/CX
XMAX=XB
IF(CX.LT.0.)XMAX=-XMAX
YINT=Y+(XMAX-X)*TPI
IF(ABS(YINT).GT.YB)GO TO 702
XBSQ=EXBSQ+YINT*YINT
GO TO 703
702 YMAX=YB
IF(CY.LT.0.)YMAX=-YMAX
XINT=X+(YMAX-Y)/TPI
XBSQ=YB+YB+XINT*XINT
YBSQ=YB+YB+XINT*XINT
XMAX=XB
IF(CX.LT.0.)XMAX=-XMAX
YINT=Y+(XMAX-X)*TPI
IF(ABS(YINT).GT.YB)GO TO 702
XBSQ=EXBSQ+YINT*YINT
GO TO 703
703 YMAX=YB
IF(CY.LT.0.)YMAX=-YMAX
XINT=X+(YMAX-Y)/TPI
XBSQ=YB+YB+XINT*XINT
YBSQ=YB+YB+XINT*XINT
XMAX=XB
IF(CX.LT.0.)XMAX=-XMAX
YINT=Y+(XMAX-X)*TPI
IF(ABS(YINT).GT.YB)GO TO 704
XBSQ=EXBSQ+YINT*YINT
GO TO 705
704 YMAX=YB8
IF(CY.LT.0.)YMAX=-YMAX
XINT=X+(YMAX-Y)/TPI
XBSQ=YB8+YB8+XINT*XINT
YBSQ=YB8+YB8+XINT*XINT
XHB=X+X+Y+Y-XBSQ
XBB=X+X+Y+Y-XBSQ
DELXB=G-G-F+HXB
DELYB=G-G-F+HXB
IF(Z.GT.0.) GO TO 2
C *****EVENT ORIGINATED OUTSIDE THE SYSTEM
C
0045 C
0046 C *****EVENT ORIGINATED OUTSIDE THE SYSTEM
0047 C
0048 C
0049 C 1 WRITE(6,100) MDUMP
0050 C 100 FORMAT(2X,'EVENT',17,' ORIGINATED OUTSIDE THE SYSTEM')
0051 GO TO 1000
0052 RADSQ=X+X+Y+Y
2 RADSQ=X+X+Y+Y
IF(IGEO.NE.2)GO TO 7731
IF(RADSQ.LT.XBSQ) GO TO 3
GO TO 1
7731 IF((X*X).GT.(XB8*XB8))GO TO 1
IF((Y*Y).GT.(YB8*YB8))GO TO 1
3 IF(Z.LT.ZB82) GO TO 4
0056
0057 IF((Y*Y).GT.(YB8*YB8))GO TO 1
0058

```

```

0059      GO TO 1
0060      4 IF<Z .GT. ZB1> GO TO 5
0061      C
0062      C*****EVENT ORIGINATED IN REGION 1
0063      C
0064          IF<F .GT. 0.> GO TO 6
0065          IF<CZ .GT. 0.> GO TO 7
0066      11 RINT=-Z/CZ
0067          IF<D .LT. RINT> GO TO 8
0068      C*****EVENT ESCAPED THROUGH SURFACE 1
0069          NCT(11)=NCT(11)+1
0070          GO TO 1000
0071      C*****EVENT REMAINED IN REGION 1
0072          8 NCT(10)=NCT(10)+1
0073          GO TO 1001
0074      7 RINT=(ZB1-Z)/CZ
0075          IF<D .LT. RINT> GO TO 8
0076      C*****EVENT ESCAPED THROUGH SURFACE 2 OR 3
0077          RADSQ=X*X+Y*Y
0078          IF<RADSQ .GT. XBSQ> GO TO 9
0079      C*****EVENT ESCAPED THROUGH SURFACE 2
0080      14 NCT(12)=NCT(12)+1
0081          GO TO 1002
0082      C*****EVENT ESCAPED THROUGH SURFACE 3
0083          9 NCT(13)=NCT(13)+1
0084          GO TO 1003
0085          6 IF<CZ .GT. 0.> GO TO 10
0086          RINT=-Z/CZ
0087          XINT=X+CX*RINT
0088          YINT=Y+CY*RINT
0089          RADSQ=XINT*XINT+YINT*YINT
0090          IF<RADSQ .LT. XBSQ> GO TO 11
0091      13 RINT=(-G+SQRT(DELXBB))/F
0092          IF<D .LT. RINT> GO TO 8
0093      C*****EVENT ESCAPED THROUGH SURFACE 6
0094          NCT(16)=NCT(16)+1
0095          GO TO 1000
0096      10 RINT=(ZB1-Z)/CZ
0097          XINT=X+CX*RINT
0098          YINT=Y+CY*RINT
0099          RADSQ=XINT*XINT+YINT*YINT
0100          IF<RADSQ .LT. XBSQ> GO TO 12
0101          GO TO 13
0102      12 IF<D .LT. RINT> GO TO 8
0103          IF<RADSQ .GT. XBSQ> GO TO 9
0104          GO TO 14
0105      5 IF<Z .LT. ZB2> GO TO 15
0106      C
0107      C*****EVENT ORIGINATED IN REGION 2
0108      C
0109          IF<F .GT. 0.> GO TO 16
0110          IF<CZ .LT. 0.> GO TO 17
0111      21 RINT=(ZB2-Z)/CZ
0112          IF<D .LT. RINT> GO TO 18
0113      C*****EVENT ESCAPED THROUGH SURFACE 10
0114          NCT(50)=NCT(50)+1
0115          GO TO 1000
0116      C*****EVENT REMAINED IN REGION 2
0117          18 NCT(20)=NCT(20)+1
0118          GO TO 1001

```

```

0119      17 RINT=(ZB2-Z)/CZ
0120      IF(D .LT. RINT) GO TO 18
0121 C*****EVENT ESCAPED THROUGH SURFACE 8 OR 9
0122      RADSQ=X*X+Y*Y
0123      IF(RADSQ .GT. XBSQ) GO TO 19
0124 C*****EVENT ESCAPED THROUGH SURFACE 8
0125      24 NCT(28)=NCT(28)+1
0126      GO TO 1002
0127 C*****EVENT ESCAPED THROUGH SURFACE 9
0128      19 NCT(29)=NCT(29)+1
0129      GO TO 1003
0130      16 IF(CZ .LT. 0.) GO TO 20
0131      RINT=(ZBB2-Z)/CZ
0132      XINT=X+CX*RINT
0133      YINT=Y+CY*RINT
0134      RADSQ=XINT*XINT+YINT*YINT
0135      IF(RADSQ .LT. XBBSQ) GO TO 21
0136      23 RINT=(-G+SQRT(DELXBB))/F
0137      IF(D .LT. RINT) GO TO 18
0138 C*****EVENT ESCAPED THROUGH SURFACE 7
0139      NCT(27)=NCT(27)+1
0140      GO TO 1000
0141      20 RINT=(ZB2-Z)/CZ
0142      XINT=X+CX*RINT
0143      YINT=Y+CY*RINT
0144      RADSQ=XINT*XINT+YINT*YINT
0145      IF(RADSQ .LT. XBBSQ) GO TO 22
0146      GO TO 23
0147      22 IF(D .LT. RINT) GO TO 18
0148      IF(RADSQ .GT. XBSQ) GO TO 19
0149      GO TO 24
0150      15 RADSQ=X*X+Y*Y
0151      IF(RADSQ .GT. XBSQ) GO TO 25
0152 C
0153 C*****EVENT ORIGINATED IN REGION 3
0154 C
0155      IF(F .GT. 0.) GO TO 26
0156      IF(CZ .GT. 0.) GO TO 27
0157      RINT=(ZB1-Z)/CZ
0158      IF(D .LT. RINT) GO TO 28
0159 C*****EVENT ESCAPED THROUGH SURFACE 2
0160      30 NCT(32)=NCT(32)+1
0161      GO TO 1002
0162 C*****EVENT REMAINED IN REGION 3
0163      28 NCT(30)=NCT(30)+1
0164      GO TO 1001
0165      27 RINT=(ZB2-Z)/CZ
0166      IF(D .LT. RINT) GO TO 28
0167 C*****EVENT ESCAPED THROUGH SURFACE 8
0168      32 NCT(38)=NCT(38)+1
0169      GO TO 1002
0170      26 RINT=(-G+SQRT(DELXB))/F
0171      ZINT=Z+CZ*RINT
0172      IF(ZINT .GT. ZB1) GO TO 29
0173      RINT=(ZB1-Z)/CZ
0174      IF(D .LT. RINT) GO TO 28
0175      GO TO 30
0176      29 IF(ZINT .LT. ZB2) GO TO 31
0177      RINT=(ZB2-Z)/CZ
0178      IF(D .LT. RINT) GO TO 28

```

```

0179      GO TO 32
0180      31 IF(D .LT. RINT) GO TO 28
0181 C*****EVENT ESCAPED THROUGH SURFACE 4
0182      NCT(34)=NCT(34)+1
0183      GO TO 1002
0184 C
0185 C*****EVENT ORIGINATED IN REGION 4
0186 C
0187      25 IF(G .LT. 0.) GO TO 33
0188      38 RINT=(-G+SQRT(DELXBB))/F
0189      ZINT=Z+CZ+RINT
0190      IF(ZINT .GT. ZB1) GO TO 34
0191      RINT=(ZB1-Z)/CZ
0192      IF(D .LT. RINT) GO TO 35
0193 C*****EVENT ESCAPED THROUGH SURFACE 3
0194      40 NCT(43)=NCT(43)+1
0195      GO TO 1003
0196 C*****EVENT REMAINED IN REGION 4
0197      35 NCT(40)=NCT(40)+1
0198      GO TO 1001
0199      34 IF(ZINT .LT. ZB2) GO TO 36
0200      RINT=(ZB2-Z)/CZ
0201      IF(D .LT. RINT) GO TO 35
0202 C*****EVENT ESCAPED THROUGH SURFACE 9
0203      42 NCT(49)=NCT(49)+1
0204      GO TO 1003
0205      36 IF(D .LT. RINT) GO TO 35
0206 C*****EVENT ESCAPED THROUGH SURFACE 5
0207      NCT(45)=NCT(45)+1
0208      GO TO 1000
0209      33 IF(DELXB .GT. 0.) GO TO 37
0210      GO TO 38
0211      37 RINT=(-G-SQRT(DELXB))/F
0212      ZINT=Z+CZ+RINT
0213      IF(ZINT .GT. ZB1) GO TO 39
0214      RINT=(ZB1-Z)/CZ
0215      IF(D .LT. RINT) GO TO 35
0216      GO TO 40
0217      39 IF(ZINT .LT. ZB2) GO TO 41
0218      RINT=(ZB2-Z)/CZ
0219      IF(D .LT. RINT) GO TO 35
0220      GO TO 42
0221      41 IF(D .LT. RINT) GO TO 35
0222 C*****EVENT ESCAPED THROUGH SURFACE 4
0223      NCT(44)=NCT(44)+1
0224      GO TO 1002
0225      1000 INOUT=0
0226      RETURN
0227      1001 INOUT=1
0228      RETURN
0229      1002 INOUT=2
0230      RETURN
0231      1003 INOUT=3
0232      RETURN
0233      END

```


4.2 LSQVM LISTING

&CRVAA T=00004 IS ON CR00026 USING 00031 BLKS R=0000

```

0001 FTM4,L
0002     PROGRAM LSQVM
0003 C
0004 C     JTB - 01/14/77 - FIX DECLARATIVE STATEMENT ORDER FOR FTM4
0005 C
0006     COMMON /VARMT/  F      ,  FP      ,  FB      ,  SCALEF  ,  FO      ,
0007     1 S(40),  X(40),  XP(40),  T(40),  SCALEX(40),IFCALL,
0008     2 GS,  G(40),  GP(40),  GB(40),  GSP,  GSB,
0009     3 HK(40,40),  C(40,40),  DELTA,  GSS,  E,  IMOVE,
0010     4 EL      ,  DOUBLE ,  Z      ,  Q      ,  A      ,  ADAR  ,
0011     5 N      ,  IT      ,  ITT     ,  MS     ,  K      ,  ICHVRG ,
0012     6 IERR    ,  LOOPF   ,  IFIN   ,  ISHOT  ,  IMOVEL ,  INOVER ,
0013     7 IREAD   ,  IWRITE  ,  ICLOCK ,  IDENT  ,  ISTEP ,  ISCALE
0014     COMMON /TAPEO/ IOUTAP
0015     COMMON /BEST/ FBEST,FERR,XBEST(40),XERR(40),IBEST,ISAME
0016     COMMON /IUBLK/INPUT(144),NIN(3),IDUF(128)
0017     COMMON /FLBLK/IOUT(144),NOUT(3)
0018 C
0019     LOGICAL IFIN
0020     CALL CREAT(IOUT,IERR,NOUT,20,3,0,26,128)
0021     IF(IERR.LT.0)GO TO 901
0022     CALL OPEN(INPUT,IERR,NIN,0,0,26,128)
0023     IF(IERR.LT.0)GO TO 902
0024 C
0025 C     INITIALIZE
0026     1 IT=1
0027     ITT=0
0028     MS=0
0029     LOOPF=0
0030     IMOVEL=0
0031     INOVER=0
0032     IFIN=.FALSE.
0033     DOUBLE=1.0
0034     FBEST=-1.E37
0035     FFIRST=FBEST
0036     FSECND=FBEST
0037     FTHIRD=FBEST
0038     FO=FBEST
0039     CALL READN
0040 C START      CALCULATE S AND GS
0041     1000 CALL MATPY(N,N,H,G,S)
0042     DO 1010 I=1,N
0043     S(I)=-S(I)
0044     1010 CONTINUE
0045     L=1
0046     CALL MATPY (L,N,S,G,GS)
0047     IF (GS) 1100,5400,1030
0048 C     H NOT POSITIVE DEFINITE
0049     1030 L=1
0050     CALL ERROR(L)
0051     GO TO 1000
0052     1100 CONTINUE
0053 C ESTIMATE LOCATION OF MINIMUM IN DIRECTION OF S
0054     EL=AMIN1(2.0,-2.0*F/GS)
0055     1110 DO 1120 I=1,N
0056     XP(I)=X(I)+EL*S(I)
0057     1120 CONTINUE
0058     M1=2

```

```

0059      CALL CALCF (GP,FP,XP,M1)
0060      IF (M1 .GT. 0) GO TO 1200
0061      M1=2
0062 C      READY OVERFLOW
0063      L=2
0064      CALL ERROR(L)
0065      GO TO (1100,7000),L
0066 1200 IERR=0
0067      L=1
0068      CALL MATPY (L,N,S,GP,GSP)
0069      IF ((GSP .GE. 0.0) .OR. (FP .GE. F)) GO TO 2000
0070 C      UNDERSHOOT
0071      L=3
0072      CALL ERROR(L)
0073      GO TO (1000,1110,4000,5000),L
0074 C AIM
0075 2000 Z=GS+GSP+3.0*(F-FP)/EL
0076      ISHOT=0
0077      TO=Z*Z-GS*GSP
0078      IF (TO .GT. 0.) GO TO 2010
0079      WRITE (IOUTAP,2005) TO
0080 2005 FORMAT(5H TO =,E13.5)
0081      TO=0.
0082 2010 CONTINUE
0083      Q=SQRT(TO)
0084      A=(GSP+Q-Z)/(GSP-GS+Q+Q)
0085      FO=FP-EL*(GSP+Z+Q+Q)*A*A/3.0
0086      IF (OK(FO) .EQ. 0.) GO TO 2100
0087 C      AIM OVERFLOW
0088      L=4
0089      CALL ERROR(L)
0090 2100 IF ((A .GT. 0.0) .AND. (A .LT. 1.0)) GO TO 2200
0091 C      MINIMUM LIES AT AN END POINT
0092      L=5
0093      CALL ERROR(L)
0094      GO TO (1000,5000),L
0095 2200 ABAR=1.0-A
0096      DO 2210 I=1,N
0097      T(I)=A*X(I)+ABAR*XP(I)
0098 2210 CONTINUE
0099 C FIRE
0100 3000 M1=2
0101      CALL CALCF (GB,FB,T,M1)
0102      IF (M1 .GT. 0) GO TO 3100
0103      M1=2
0104 C      FIRE OVERFLOW
0105      L=6
0106      CALL ERROR(L)
0107      GO TO 2200
0108 3100 IF (AMIN(F,FP) .GE. FB) GO TO 3200
0109 C      MOVES
0110      L=7
0111      CALL ERROR(L)
0112      GO TO (1200,5000),L
0113 3200 LOOPF=0
0114      F=FB
0115      DO 3210 I=1,N
0116      X(I)=T(I)
0117      T(I)=G(I)
0118      G(I)=GB(I)

```

```

0119 3210 CONTINUE
0120 L=1
0121 CALL MATPY (L,N,S,GB,GSB)
0122 TP1=A/ABAR
0123 TP2=ABAR/A
0124 TO=GSB*(TP1-TP2)
0125 GSS=TO+Q+Q
0126 IF (GSS .GT. 0.0) GO TO 3300
0127 C 2ND DERIVATIVE NEGATIVE
0128 L=0
0129 CALL ERROR(L)
0130 GO TO 5000
0131 3300 DO 3310 I=1,N
0132 CB(I)=(GB(I)-T(I))*TP1+(GP(I)-GB(I))*TP2
0133 3310 CONTINUE
0134 C DRESS
0135 4000 CALL MATPY (N,N,H,GB,T)
0136 C CALCULATE LENGTH OF THE 2ND DERIVATIVE IN THE DIRECTION OF THE STEP
0137 L=1
0138 CALL MATPY (L,N,GB,T,TO)
0139 DO 4020 I=1,N
0140 DO 4010 J=1,N
0141 HK(I,J)=H(I,J)-T(I)*T(J)/TO+EL*S(I)*S(J)/GSS
0142 4010 CONTINUE
0143 4020 CONTINUE
0144 DELTA=DELTA*(EL*GSS/TO)
0145 IF (OK(DELTA) .EQ. 0.) GO TO 5000
0146 L=9
0147 CALL ERROR(L)
0148 L=2
0149 IFIN=.TRUE.
0150 C CONVERGENCE CRITERION
0151 5000 IF (F .EQ. FTHIRD) GO TO 5400
0152 FTHIRD=FSECND
0153 FSECND=FFIRST
0154 FFIRST=F
0155 C COMPUTE DIRECTION OF NEXT SEARCH
0156 5100 CALL MATPY(N,N,H,G,S)
0157 DO 5110 I=1,N
0158 S(I)=-S(I)
0159 5110 CONTINUE
0160 L=1
0161 CALL MATPY(L,N,G,S,GGS)
0162 IF (GGS) 5200,5400,5120
0163 5120 L=1
0164 CALL ERROR(L)
0165 GO TO 5100
0166 5200 IF (IFIN .AND. (ICNVRG .LT. 4)) GO TO 5400
0167 L=MOD(ICNVRG,4)+1
0168 GO TO (5210,5230,5240,5210),L
0169 5210 DO 5220 I=1,N
0170 IF (ABS(S(I)) .GE. E) GO TO 5300
0171 5220 CONTINUE
0172 IF (N .EQ. 1) GO TO 5400
0173 5230 IF (ABS(GGS) .LT. E) GO TO 5400
0174 GO TO 5300
0175 5240 IF (ABS(FFIRST-FSECND) .LT. E) GO TO 5400
0176 5300 L=1
0177 CALL RITEO(L)
0178 IT=IT+1

```

```

0179      GS=CCS
0180      GO TO (1100,5400,6000),L
0181      5400 M1=3
0182      CALL CALCF(G,F,X,M1)
0183      L=2
0184      CALL RITEO(L)
0185      C MINIMUM SURVAY
0186      6000 IF (ABS(SCALEF*F-FBEST)/SCALEF .GT. .1) GO TO 6050
0187      ISANE=1
0188      DO 6010 I=1,N
0189      XERR(I)=AMAX1(XERR(I),ABS(XBEST(I)-X(I)*SCALEX(I)))
0190      6010 CONTINUE
0191      FERR=AMAX1(FERR,ABS(FBEST-SCALEF*F))
0192      GO TO 6100
0193      6050 DO 6060 I=1,N
0194      XERR(I)=0.0
0195      6060 CONTINUE
0196      FERR=0.0
0197      ISANE=0
0198      6100 IF (SCALEF*F .GT. ABS(FBEST)) GO TO 7000
0199      FBEST=SCALEF*F
0200      IBEST=MS
0201      DO 6110 I=1,N
0202      XBEST(I)=X(I)*SCALEX(I)
0203      6110 CONTINUE
0204      C STUFF
0205      7000 IF (MS .LT. K) GO TO 7100
0206      L=3
0207      CALL RITEO(L)
0208      GO TO 1
0209      7100 MS=MS+1
0210      C IF (IWRITE .GT. 1) WRITE(IOUTAP,7110) MS
0211      C7110 FORMAT (19H1RANDOM STEP NUMBER,I4)
0212      MM=(MS+1)/2
0213      DO 7120 I=1,N
0214      M=MOD(I,MM)+MOD(MS,2)
0215      T(I)=2*MOD(M,2)-1
0216      T(I)=T(I)*SIGN(1.,SCALEX(I))
0217      7120 CONTINUE
0218      CALL MATPY (N,N,H,T,S)
0219      L=1
0220      CALL MATPY (L,N,S,T,TO)
0221      EL=(10.0**((ISTEP-1))/SQRT(TO))
0222      7130 DO 7140 I=1,N
0223      X(I)=X(I)+EL*S(I)
0224      7140 CONTINUE
0225      M1=2
0226      CALL CALCF (G,F,X,M1)
0227      IT=0
0228      L=1
0229      CALL RITEO(L)
0230      IT=1
0231      GO TO 1000
0232      901 WRITE(1,911)IERR
0233      911 FORMAT(' CREATE ERROR=',I4)
0234      GO TO 951
0235      902 WRITE(1,912)IERR
0236      912 FORMAT(' OPEN ERROR=',I4)
0237      951 STOP
0238      END

```

```

0239 C      SUBROUTINE TABLE(FIT,G,X,T,M1,INVAR)
0240          DIMENSION G(40),X(40)
0241          DIMENSION T(1)
0242          IF(M1.NE.1) GO TO 50
0243          RETURN
0244
0245 50 CONTINUE
0246          G(1)=1.0
0247          IF(INVAR.NE.2)GO TO 65
0248          SQ=T(2)*T(2)
0249          CU=SQ*T(2)
0250          QU=CU*T(2)
0251          G(2)=T(2)
0252          G(3)=0.5*(3.0+SQ-1.0)
0253          G(4)=0.5*(5.0+CU-3.0*T(2))
0254          G(5)=0.125*(35.0+QU-30.0*G(3.0))
0255          DO 52 I=6,25
0256              G(I)=G(I-5)*T(1)
0257 52 CONTINUE
0258          FIT=0.0
0259          DO 60 I=1,25
0260              FIT=FIT+X(I)*G(I)
0261 60 CONTINUE
0262          RETURN
0263
0264 65 G(2)=T(1)
0265          G(3)=G(2)*T(1)
0266          G(4)=G(3)*T(1)
0267          G(5)=G(4)*T(1)
0268          FIT=0.0
0269          DO 70 I=1,5
0270              FIT=FIT+X(I)*G(I)
0271 70 CONTINUE
0272          RETURN
0273          END
0274 C
0275          BLOCK DATA
0276          COMMON/VARMT/F,FP,FB,SCALEF,FO,SC(40),X(40),XP(40),T(40),SCALEX(40)
0277          1,IFCALL,GS,G(40),GP(40),GB(40),GSP,GSR,H(40,40),C(40,40),DELTA,GSS
0278          2,E,IMOVE,EL,DOUBLE,Z,Q,A,ABAR,N,IT,ITT,MS,K,ICNVRC,IERR,LOOPF,IFIN
0279          3,ISHOT,IMOVEL,IMOVER,IREAD,IWRITE,ICLOCK,IDENT,ISTEP,ISCALE
0280          COMMON /BEST/FBEST,FERR,XBEST(40),XERR(40),IBEST,ISAME
0281          COMMON /IOBLK/INPUT(144),NIN(3),IBUF(128)
0282          COMMON /FLBLK/IOUT(144),NOUT(3)
0283          COMMON /TAPEO/ IOUTAP
0284          DATA NIN/2HLS,2HQ1,2HN /
0285          DATA NOUT/2HOU,2HT0,2H2 /
0286          DATA IOUTAP/6/
0287          DATA IREAD,IWRITE,K,ICLOCK,ISTEP/5*0/
0288          DATA IMOVE,ICNVRC,IDENT,ISCALE/4*0/
0289          END

```

&CRYBB T=00004 IS ON CR00026 USING 00019 BLKS R=0000

```

0001 FTM4,L
0002 SUBROUTINE ERROR(L)
0003 COMMON /VART/ F , FP , FB , SCALEF , FO ,
0004 1 S(40), X(40), XP(40), T(40), SCALEX(40), IFCALL,
0005 2 GS, G(40), GP(40), GB(40), GSP, GSB,
0006 3 H(40,40), C(40,40), DELTA, GSS, E, IMOVE,
0007 4 EL , DOUBLE , Z , Q , A , ABAR ,
0008 5 N , IT , ITT , MS , K , ICNVRG ,
0009 6 IERR , LOOPF , IFIN , ISHOT , IMOVEL , IMOVER ,
0010 7 IREAD , IWRITE , ICLOCK , IDENT , ISTEP , ISCALE
0011 COMMON /TAPEO/ IOUTAP
0012 LOGICAL IFIN
0013 GO TO (100,200,300,400,500,600,700,800,900,1000 ),L
0014 100 CONTINUE
0015 C 100 IF (IWRITE .NE. 0) WRITE (IOUTAP,110) GS
0016 C 110 FORMAT(41H H IS NO LONGER POSITIVE DEFINITE FOR GS=,E13.5,70H SO D
0017 C *IAGONAL ELTS. OF H ARE SET POSITIVE AND OFF DIAGONAL ELTS. ZEROED)
0018 DELTA=1.0
0019 DO 130 I=1,N
0020 H(I,I)=ABS(H(I,I))
0021 DO 120 J=1,N
0022 IF (I .NE. J) H(I,J)=0.0
0023 120 CONTINUE
0024 IF (H(I,I) .NE. 0.0) DELTA=DELTA*H(I,I)
0025 130 CONTINUE
0026 L=1
0027 IF (IREAD .LE. -1) CALL CSTRN(L)
0028 RETURN
0029 200 CONTINUE
0030 C 200 IF (IWRITE .NE. 0) WRITE (IOUTAP,201)
0031 C 201 FORMAT (58H READY OVERFLOW LENGTH OF STEP IS HALVED FOR ANOTHER
0032 C *TRY)
0033 IF ((MS .NE. 0) .AND. (IT .EQ. 1)) .OR. (IERR .GE. 5) GO TO 210
0034 IERR=IERR+1
0035 L=1
0036 RETURN
0037 210 CONTINUE
0038 WRITE (IOUTAP,215)
0039 215 FORMAT (41H000 MANY READY OVERFLOWS CASE TERMINATED)
0040 MS=K
0041 L=2
0042 RETURN
0043 300 F=FP
0044 DO 310 I=1,N
0045 G(I)=GP(I)
0046 X(I)=XP(I)
0047 310 CONTINUE
0048 IF (MS .EQ. 0) GO TO 320
0049 EL=2.*EL
0050 L=2
0051 RETURN
0052 320 IF (EL .GE. 1.0) GO TO 350
0053 L=1
0054 RETURN
0055 350 DELTA=(DOUBLE+1.0)*DELTA
0056 DO 370 I=1,N
0057 DO 360 J=1,N
0058 H(I,J)=H(I,J)-DOUBLE*S(I)*S(J)/GS

```

```

0059      360 CONTINUE
0060      370 CONTINUE
0061      DOUBLE=DOUBLE+1.0
0062      ISHOT=2
0063      L=4
0064      RETURN
0065      400 CONTINUE
0066 C 400 IF (IWRITE .NE. 0) WRITE (IOUTAP,401)
0067 C 401 FORMAT (13H AIM OVERFLOW)
0068      RETURN
0069      500 IF (A .LE. 0.0) GO TO 550
0070 C      IF (IWRITE .NE. 0) WRITE (IOUTAP,510)
0071 C 510 FORMAT (64H MINIMUM IN READY INTERVAL APPEARS TO LIE AT THE LEFT
0072 C      *END POINT)
0073      IF (IDENT .LE. 0) GO TO 590
0074      DELTA=1.0
0075      DO 530 I=1,N
0076      H(I,I)=ABS(H(I,I))
0077      DO 520 J=1,N
0078      IF (I .NE. J) H(I,J)=0.0
0079      520 CONTINUE
0080      IF (H(I,I) .NE. 0.0) DELTA=DELTA+H(I,I)
0081      530 CONTINUE
0082 C      IF (IWRITE .NE. 0) WRITE (IOUTAP,535)
0083 C 535 FORMAT (67HODIAGONAL ELTS. OF H ARE SET POSITIVE AND OFF DIAGONAL
0084 C      *ELTS. ZEROED)
0085      L=1
0086      IF (IREAD .LT. 0) CALL CSTRN(L)
0087      L=1
0088      RETURN
0089      550 CONTINUE
0090 C 550 IF (IWRITE .NE. 0) WRITE (IOUTAP,560)
0091 C 560 FORMAT (64H MINIMUM IN READY INTERVAL APPEARS TO LIE AT THE RIGHT
0092 C      *END POINT)
0093      F=FP
0094      DO 570 I=1,N
0095      G(I)=CP(I)
0096      X(I)=XP(I)
0097      570 CONTINUE
0098      580 IF (IDENT .LE. 1) IFIN=.TRUE.
0099      L=2
0100      RETURN
0101      600 CONTINUE
0102 C 600 IF (IWRITE .NE. 0) WRITE (IOUTAP,601)
0103 C 601 FORMAT (83HOFIRE OVERFLOW TRY POINT HALFWAY BETWEEN INTERPOLATED
0104 C      * MINIMUM AND LOWER END POINT)
0105      IF (FP .GT. F) A=A-1.0
0106      A=(1.0+A)/2.0
0107      RETURN
0108      700 CONTINUE
0109      M=MOVE+3
0110      IF (LOOPF .LT. M) GO TO 710
0111 C      IF (IWRITE .NE. 0) WRITE (IOUTAP,705) M
0112 C 705 FORMAT (24H TRIED TO MOVE MORE THAN, I2, 6H TIMES)
0113      IFIN=.TRUE.
0114      LOOPF=0
0115      L=2
0116      RETURN
0117      710 LOOPF=LOOPF+1
0118      IF (F .GE. FP) GO TO 750

```



```

0119      IMOVEL=IMOVEL+1
0120      EL=ABAR*EL
0121      FP=FB
0122      DO 720 I=1,N
0123      XP(I)=T(I)
0124      GP(I)=GB(I)
0125      720 CONTINUE
0126 C    GOES TO AIM
0127      L=1
0128      RETURN
0129      750 IMOVER=IMOVER+1
0130      EL=EL*A
0131      F=FB
0132      LL=1
0133      CALL MATPY(LL,N,S,GB,GS)
0134      DO 760 I=1,N
0135      X(I)=T(I)
0136      G(I)=GB(I)
0137      760 CONTINUE
0138      L=1
0139      RETURN
0140      800 CONTINUE
0141 C 800 IF (IWRITE .NE. 0) WRITE (IOUTAP,810)
0142 C 810 FORMAT (57H APPROXIMATE 2ND DERIVATIVE NEGATIVE DONT MODIFY H MATR
0143 C      *IX)
0144      RETURN
0145      900 CONTINUE
0146 C 900 IF (IWRITE .NE. 0) WRITE (IOUTAP,901)
0147 C 901 FORMAT (15H0DRESS OVERFLOW)
0148      DELTA=1.
0149      1000 RETURN
0150      END

```

%CRVCC T=00004 IS ON CR00026 USING 00019 BLKS R=0000

```

0001 FTN4,L
0002 SUBROUTINE READN
0003 COMMON /VARM/ F , FP , FB , SCALEF , FO ,
0004 1 S(40), X(40), XP(40), T(40), SCALEX(40), IFCALL,
0005 2 GS, G(40), GP(40), GB(40), GSP, GSB,
0006 3 H(40,40), C(40,40), DELTA, GSS, E, IMOVE,
0007 4 EL , DOUBLE , Z , Q , A , ABAR ,
0008 5 N , IT , ITT , MS , K , ICHVRG ,
0009 6 IERR , LOOPF , IFIN , ISHOT , IMOVEL , IMOVER ,
0010 7 IREAD , IWRITE , ICLOCK , IDENT , ISTEP , ISCALE
0011 COMMON /TAPE/ IOUTAP
0012 COMMON /IOBLK/INPUT(144),NIN(3),IBUF(128)
0013 DATA ICON / 0 /
0014 C WRITE (IOUTAP,5)
0015 C 5 FORMAT ('*****NEW CASE ')
0016 IF (ICON .EQ. 1) GO TO 20
0017 ICON=1
0018 ISIZE=10
0019 K=5
0020 ICK=0
0021 C CALL READF(INPUT,IERR,IBUF,30,LEN)
0022 C CALL CODE(60)
0023 C READ (IBUF,10) IREAD,IWRITE,K,ISIZE,ISTEP,ICK,IMOVE,ICLOCK,ICON
0024 C 1,ICHVRG,IDENT,ISCALE
0025 C 10 FORMAT ('1615')
0026 C WRITE (IOUTAP,10) IREAD,IWRITE,K,ISIZE,ISTEP,ICK,IMOVE,ICLOCK,ICON
0027 C 1,ICHVRG,IDENT,ISCALE
0028 E=2.**((ISIZE-30))
0029 K=K+1
0030 20 M1=1
0031 CALL FCN(M,G,F,X,M1)
0032 IFCALL=1
0033 N=IABS(M)
0034 IF (M .LE. 0) GO TO 30
0035 NF=2*N
0036 CALL READF(INPUT,IERR,X,NF)
0037 M1=2
0038 M=N
0039 CALL FCN(M,G,F,X,M1)
0040 30 WRITE (IOUTAP,35) (G(I),I=1,N)
0041 35 FORMAT('21H0AT THE INITIAL GUESS /1H0,20X,1HG/(10E13.5))
0042 WRITE (IOUTAP,40) F,(X(I),I=1,N)
0043 40 FORMAT ('4H0F =,E13.5/1H0,20X,1HX/(10E13.5))
0044 IF (ICK .EQ. 1) CALL DIFCK
0045 IF (ISCALE .NE. 1) GO TO 60
0046 SCALEF=1.0
0047 DO 50 I=1,N
0048 SCALEX(I)=1.0
0049 50 CONTINUE
0050 GO TO 80
0051 60 SCALEF=1.0
0052 IF (F .NE. 0.0) SCALEF=F
0053 F=F/SCALEF
0054 DO 70 I=1,N
0055 SCALEX(I)=1.0
0056 IF (X(I) .NE. 0.) SCALEX(I)=X(I)
0057 X(I)=X(I)/SCALEX(I)
0058 G(I)=G(I)*SCALEX(I)/SCALEF

```

```

0059      70 CONTINUE
0060      80 IF (IREAD .LE. -1) GO TO 100
0061      M=IREAD+1
0062      GO TO (200,300,400,300),M
0063 C IREAD LT 0 H MATRIX IS CONSTRAINED
0064      100 DO 120 I=1,N
0065          DO 110 J=1,N
0066              C(I,J)=0.0
0067              H(I,J)=0.0
0068          110 CONTINUE
0069              H(I,I)=1.
0070      120 CONTINUE
0071          DELTA=1.0
0072 C      WRITE (IOUTAP,125)
0073 C 125 FORMAT (40H0THE H MATRIX IS LINEARLY CONSTRAINED BY )
0074          M=-IREAD
0075          DO 140 I=1,M
0076              CALL READF(INPUT,IERR,IBUF,40,LEN)
0077              LENP=2*LEN
0078              CALL CODE(LENP)
0079              READ (IBUF,25) (C(I,J),J=1,N)
0080          25 FORMAT(8F10.0)
0081 C      WRITE (IOUTAP,130) (C(I,J),J=1,N)
0082 C 130 FORMAT (10E13.5)
0083      140 CONTINUE
0084          M=0
0085          CALL CSTRN(M)
0086 C      WRITE (IOUTAP,145)
0087 C 145 FORMAT (28H0THE CONSTRAINED H MATRIX IS )
0088 C      DO 150 I=1,N
0089 C          WRITE (IOUTAP,130) (H(I,J),J=1,N)
0090 C 150 CONTINUE
0091          RETURN
0092 C H MATRIX STARTED AS IDENTITY
0093      200 DO 220 I=1,N
0094          DO 210 J=1,N
0095              H(I,J)=0.0
0096          210 CONTINUE
0097              H(I,I)=1.0
0098      220 CONTINUE
0099          DELTA=1.0
0100 C      WRITE (IOUTAP,225)
0101 C 225 FORMAT (39H0H SET INITIALLY TO THE IDENTITY MATRIX )
0102          RETURN
0103 C DIAGONAL ELEMENTS OF H MATRIX READ IN
0104      300 DO 320 I=1,N
0105          DO 310 J=1,N
0106              H(I,J)=0.0
0107          310 CONTINUE
0108      320 CONTINUE
0109          CALL READF(INPUT,IERR,IBUF,40,LEN)
0110          LENP=2*LEN
0111          CALL CODE(LENP)
0112          READ (IBUF,25) (H(I,J),I=1,N)
0113 C      WRITE (IOUTAP,325) (H(I,I),I=1,N)
0114 C 325 FORMAT (42H0THE DIAGONAL ELEMENTS OF THE H MATRIX ARE /(10E13.5))
0115          IF (M .NE. 4) GO TO 340
0116          DO 330 I=1,N
0117              H(I,I)=H(I,I)*SCALEF/SCALEX(I)**2
0118      330 CONTINUE

```

```

0119      340 DELTA=1.0
0120      DO 350 I=1,N
0121      IF (H(I,I) .NE. 0.0) DELTA=DELTA*H(I,I)
0122      350 CONTINUE
0123      RETURN
0124 C  WHOLE H MATRIX READ IN
0125      400 TO=1.0
0126 C  WRITE (IOUTAP,405)
0127 C  405 FORMAT (17H0INITIAL H MATRIX )
0128      DO 420 I=1,N
0129      CALL READF(INPUT,IERR,IBUF,40,LEN)
0130      LENP=2*LEN
0131      CALL CODE(LENP)
0132      READ (IBUF,25) (H(I,J),J=1,N)
0133 C  WRITE (IOUTAP,25) (H(I,J),J=1,N)
0134      GSS=SCALEF/SCALEX(I)
0135      TO=TO*GSS/SCALEX(I)
0136      DO 410 J=1,N
0137      H(I,J)=H(I,J)*GSS/SCALEX(J)
0138      410 CONTINUE
0139      420 CONTINUE
0140      CALL READF(INPUT,IERR,IBUF,2)
0141      DELTA=IBUF(1)
0142      DELTA=DELTA*TO
0143      RETURN
0144      END

```

&CRYDD T=00004 IS ON CR00026 USING 00020 BLKS R=0000

```

0001  FTH4,L
0002      SUBROUTINE RITEO(L)
0003      COMMON /VARMT/  F      ,  FP      ,  FB      ,  SCALEF      ,  FO      ,
0004      1  S(40),  X(40),  XP(40),  T(40),  SCALEX(40),  IFCALL,
0005      2  GS,  G(40),  GP(40),  GB(40),  GSP,GSB,
0006      3  H(40,40),  C(40,40),  DELTA,  GSS,  E,  IMOVE,
0007      4      EL      ,  DOUBLE ,  Z      ,  Q      ,  A      ,  ABAR      ,
0008      5      N      ,  IT      ,  ITT     ,  MS     ,  K      ,  ICHVRG ,
0009      6      IERR    ,  LOOPF   ,  IFIN    ,  ISHOT   ,  IMOVEL ,  IMOVER ,
0010      7      IREAD   ,  IWRITE  ,  ICLOCK  ,  IDENT   ,  ISTEP ,  ISCALE
0011      COMMON /TAPEO/ IOUTAP
0012      COMMON /BEST/ FBEST,FERR,XBEST(40),XERR(40),IBEST,ISAME
0013      LOGICAL IFIN,FIRST,SETUP
0014      DATA FIRST,SETUP,SAFE /.FALSE.,.FALSE.,.3/
0015      ITT=ITT+1
0016      GO TO (100,300,3000),L
0017      100 CONTINUE
0018      200 IF (SETUP) GO TO 250
0019      IF (FIRST) GO TO 210
0020      IT1=ITT
0021      FIRST=.TRUE.
0022      GO TO 1000
0023      210 IF (ITT-IT1 .LT. 10) GO TO 1000
0024      SETUP=.TRUE.
0025      GO TO 1000
0026      300 IFIN=.FALSE.
0027      IF (IWRITE .LE. 0) RETURN
0028      WRITE (IOUTAP,310) MS
0029      310 FORMAT (25H1/--/--/--/--/--/--/--/21H0FINAL VALUES OF STEP ,15)
0030      GO TO 1200
0031      1000 IF (IWRITE .EQ. 0) GO TO 1600
0032      C  IF (IWRITE .NE. 1) WRITE (IOUTAP,1010)
0033      C1010 FORMAT (130H - - - - -)
0034      C * - - - - -
0035      C * - - - - -
0036      1020 GO TO (1600,1500,1400,1300,1200,1100),IWRITE
0037      1100 CONTINUE
0038      FO=FO*SCALEF
0039      WRITE (IOUTAP,1110) EL,A,Q,FO,Z
0040      1110 FORMAT (4H0EL=,E13.5,3H A=,E13.5,3H Q=,E13.5,4H FO=,E13.5,3H Z=,
0041      *E13.5)
0042      1200 WRITE (IOUTAP,1205)
0043      1205 FORMAT (1H0,20X,1HH)
0044      TO=1.0
0045      DO 1220 I=1,N
0046      GSS=SCALEX(I)/SCALEF
0047      TO=TO*SCALEX(I)
0048      DO 1210 J=1,N
0049      T(J)=GSS*SCALEX(J)*H(I,J)
0050      1210 CONTINUE
0051      WRITE (IOUTAP,1215) (T(J),J=1,N)
0052      1215 FORMAT (10E13.5)
0053      1220 CONTINUE
0054      GSS=DELTA*TO*TO/(SCALEF**N)
0055      WRITE (IOUTAP,1230) GSS
0056      1230 FORMAT (7H0DELTA=,E13.5)
0057      1300 IF (ISHOT .NE. 0) WRITE (IOUTAP,1310)
0058      1310 FORMAT (10H0UNDERSHOT)

```



```

0119      WRITE (IOUTAP,1215) (Q(I),I=1,N)
0120      IF (IFIN) CALL FLCLS
0121      RETURN
0122      END
0123 C
0124      SUBROUTINE FLWRT(N,XBEST)
0125      COMMON /FLBLK/IOUT(144),NOUT(3)
0126      DIMENSION XBEST(1)
0127      NW=2*N
0128      CALL WRITF(IOUT,IERR,XBEST,NW)
0129      RETURN
0130      END
0131 C
0132      SUBROUTINE FLCLS
0133      COMMON /FLBLK/IOUT(144),NOUT(3)
0134      CALL LOCF(IOUT,IERR,IREF,IRB,IOFF,JSEC)
0135      ITRUN=JSEC/2-IRB-1
0136      CALL CLOSE(IOUT,IERR,ITRUN)
0137      STOP
0138      END

```

&CRVEE T=00004 IS ON CR00026 USING 00009 BLKS R=0000

```

0001 FTH4,L
0002 SUBROUTINE CALCF (GG,FF,XX,M1)
0003 DIMENSION XX(40),YY(40),GG(40)
0004 COMMON /VARMT/ F , FP , FB , SCALEF , FO ,
0005 1 S(40), X(40), XP(40), T(40), SCALEX(40), IFCALL,
0006 2 GS, G(40), GP(40), GB(40), GSP, GSB,
0007 3 H(40,40), C(40,40), DELTA, GSS, E, IMOVE,
0008 4 EL , DOUBLE , Z , Q , A , ABAR ,
0009 5 N , IT , ITT , MS , K , ICNVRG ,
0010 6 IERR , LOOPF , IFIN , ISHOT , IMOVEL , IMOVER ,
0011 7 IREAD , IWRITE , ICLOCK , IDENT , ISTEP , ISCALE
0012 COMMON /TAPEO/ IOUTAP
0013 C UNSCALE X
0014 200 DO 210 I=1,N
0015 XX(I)=XX(I)*SCALEX(I)
0016 210 CONTINUE
0017 C CALL FCN AND CHECK F
0018 CALL FCN(H,GG,FF,XX,M1)
0019 IFCALL=IFCALL+1
0020 IF (OK(FF)) 310,330,320
0021 310 WRITE (IOUTAP,315)
0022 315 FORMAT (40H0AN INDEFINITE RESULT WAS RETURNED FOR F )
0023 STOP
0024 320 M1=-M1
0025 330 IF (FF .GE. 0.) GO TO 335
0026 C IF (IWRITE .NE. 0) WRITE (IOUTAP,331)
0027 C 331 FORMAT (25H0F WAS NEGATIVE F=ABS(F) )
0028 FF=-FF
0029 DO 332 I=1,N
0030 GG(I)=-GG(I)
0031 332 CONTINUE
0032 335 IF (ISCALE .EQ. 1) GO TO 600
0033 IF (IABS(M1) .NE. 3) GO TO 500
0034 C UNSCALE H MATRIX
0035 TO=1.
0036 DO 350 I=1,N
0037 GSS=SCALEX(I)/SCALEF
0038 TO=TO*GSS
0039 DO 340 J=1,N
0040 H(I,J)=H(I,J)*GSS*SCALEX(J)
0041 340 CONTINUE
0042 350 CONTINUE
0043 DELTA=DELTA*TO*TO*SCALEF**N
0044 C RESCALE F,X, AND SCALE H
0045 400 SCALEF=1.
0046 IF (FF .NE. 0.) SCALEF=FF
0047 DO 410 I=1,N
0048 SCALEX(I)=1.
0049 IF (XX(I) .NE. 0.) SCALEX(I)=XX(I)
0050 410 CONTINUE
0051 TO=1.
0052 DO 430 I=1,N
0053 GSS=SCALEF/SCALEX(I)
0054 TO=TO*SCALEX(I)
0055 DO 420 J=1,N
0056 H(I,J)=H(I,J)*GSS/SCALEX(J)
0057 420 CONTINUE
0058 430 CONTINUE

```



```

0059      DELTA=DELTA*SCALEF**N/TO**2
0060 C    SCALE F,X,G
0061      500 FF=FF/SCALEF
0062      DO 510 I=1,N
0063          XX(I)=XX(I)/SCALEX(I)
0064          GG(I)=GG(I)*SCALEX(I)/SCALEF
0065      510 CONTINUE
0066      600 RETURN
0067      END

```

&CRYFF T=00004 IS ON CR00026 USING 00010 BLKS R=0000

```

0001  FTN4,L
0002      SUBROUTINE DIFCK
0003      COMMON /VART/  F      , FP      , FB      , SCALEF      , FO      ,
0004      1  S<40> , X<40> , XP<40> , T<40> , SCALEX<40> , IFCALL ,
0005      2  GS , G<40> , GP<40> , GB<40> , GSP , GSB ,
0006      3  H<40,40> , C<40,40> , DELTA , GSS , E , IMOVE ,
0007      4  EL      , DOUBLE , Z      , Q      , A      , ABAR      ,
0008      5  N      , IT      , ITT      , MS      , K      , ICNVRG      ,
0009      6  IERR      , LOOPF      , IFIN      , ISHOT      , IMOVEL      , IMOVER      ,
0010      7  IREAD      , IWRITE      , ICLOCK      , IDENT      , ISTEP      , ISCALE
0011      COMMON /TAPE0/ IOUTAP
0012      IFAIL=0
0013      DO 10 I=1,N
0014      T(I)=X(I)
0015  10 CONTINUE
0016      DO 100 I=1,N
0017      DO 50 ITRY=1,3
0018      TRY=AMAX1(1.0,AMAX1(ABS(G(I)),1.0),ABS(X(I))+1.0E-4)/(10.0*(ITRY=
0019      *1))
0020      T(I)=X(I)-TRY
0021      TB=T(I)
0022      M1=2
0023      CALL FCH (N,GB,FB,T,M1)
0024      T(I)=X(I)+TRY
0025      TP=T(I)
0026      M1=2
0027      CALL FCH (N,GP,FP,T,M1)
0028      T(I)=X(I)
0029      FN=TRY*(GB(I)+4.0*G(I)+GP(I))/3.0
0030      FD=FP-FB
0031      GN=FD/(2.0*TRY)
0032      GE=ABS(GN-G(I))
0033      GT=ABS(GN+G(I))*0.5E-3
0034      FE=ABS(FD-FN)
0035      FT=ABS(FD+FN)*0.5E-4
0036      IF ((FE .LE. FT) .OR. (GE .LE. GT)) GO TO 98
0037  50 CONTINUE
0038      WRITE (IOUTAP,55) I
0039  55 FORMAT (1H0/4H0THE,14,83H-TH COMPONENT OF THE ANALYTICAL DERIVATIV
0040      IE DOES NOT AGREE TO WITHIN .1 OF 1 PERCENT )
0041      WRITE (IOUTAP,56) I,TRY,TB,FB,GB(I),X(I),F,G(I),TP,FP,GP(I)
0042      $,GN,G(I),GE,GT,FN,FD,FE,FT
0043  56 FORMAT(8H WHEN X<12,18H> IS CHANGED BY H=,E13.5
0044      $/'0X- = X-H =',E13.5,' F(X-) =',E13.5,' G(X-) =',E13.5
0045      $/' X      =',E13.5,' F(X ) =',E13.5,' G(X ) =',E13.5
0046      $/' X+ = X+H =',E13.5,' F(X+) =',E13.5,' G(X+) =',E13.5
0047      $/32H0      (F(X+)-F(X-))/(2.0*H)= ,E13.5,'      G=,E13.5
0048      $' ERR=,E13.5,' TOL=,E13.5
0049      $/32H0(H/3.0)*(G(X-)+4.0*G(X)+G(X+))= ,E13.5,' F(X+)-F(X-)=,E13.5
0050      $' ERR=,E13.5,' TOL=,E13.5
0051      $)
0052      IFAIL=1
0053      GO TO 100
0054  98 WRITE (IOUTAP,99) I,ITRY
0055  99 FORMAT (16H0SUCCESS FOR THE,14,20H-TH COMPONENT ON THE,14,4H TRY)
0056  100 CONTINUE
0057      IF (IFAIL .EQ. 1) STOP
0058      RETURN
0059      END

```

LCRYGG T=00004 IS ON CR00026 USING 00012 BLKS R=0000

```

0001 FTH4,L
0002 SUBROUTINE CSTRN (ITEST)
0003 COMMON /VART/ F , FP , FB , SCALEF , FD ,
0004 1 S<40>, X<40>, XP<40>, T<40>, SCALEX<40>, IFCALL,
0005 2 GS, G<40>, GP<40>, GB<40>, GSP, GSB,
0006 3 H<40,40>, C<40,40>, DELTA, GSS, E, IMOVE,
0007 4 EL , DOUBLE , Z , Q , A , ABAR ,
0008 5 N , IT , ITT , MS , K , ICNVRG ,
0009 6 IERR , LOOPF , IFIN , ISHOT , IMOVEL , IMOVER ,
0010 7 IREAD , IWRITE , ICLOCK , IDENT , ISTEP , ISCALE
0011 COMMON /TAPEO/ IOUTAP
0012 LOGICAL ZERO(1)
0013 EQUIVALENCE (ZERO,T)
0014 IF (ITEST .NE. 0) GO TO 70
0015 DO 10 I=1,N
0016 ZERO(I)=.FALSE.
0017 10 CONTINUE
0018 DO 60 J=1,N
0019 C FIND THE PIVOT FOR THE J-TH COL.
0020 PIVOT=0.
0021 DO 20 I=1,N
0022 IF (ZERO(I)) GO TO 20
0023 SAVE=ABS(C(I,J))
0024 IF (SAVE .LE. PIVOT) GO TO 20
0025 PIVOT=SAVE
0026 IBIG=I
0027 20 CONTINUE
0028 IF (PIVOT .EQ. 0.) GO TO 60
0029 C NON ZERO PIVOT SET FLAG AND SWITCH ROWS J AND IBIG
0030 ZERO(J)=.TRUE.
0031 PIVOT=C<IBIG,J>
0032 DO 30 I=J,N
0033 SAVE=C<J,I>
0034 C<J,I>=C<IBIG,I>/PIVOT
0035 C<IBIG,I>=SAVE
0036 30 CONTINUE
0037 C ELIMINATE OFF DIAGONAL ELEMENTS
0038 DO 50 I=1,N
0039 IF (I .EQ. J) GO TO 50
0040 DO 40 II=J,N
0041 C<I,II>=C<I,II>-C<I,J>*C<J,II>
0042 40 CONTINUE
0043 50 CONTINUE
0044 60 CONTINUE
0045 70 IF (ISCALE .EQ. 1) GO TO 100
0046 DO 90 I=1,N
0047 GSS=SCALEF/SCALEX(I)
0048 DO 80 J=1,N
0049 H<I,J>=H<I,J>*SCALEX(J)/GSS
0050 80 CONTINUE
0051 90 CONTINUE
0052 100 DELTA=1.
0053 DO 150 I=1,N
0054 II=N-I+1
0055 IF (C<II,II> .NE. 0.) GO TO 110
0056 DELTA=DELTA*H<II,II>
0057 GO TO 150
0058 110 DO 130 J=II,N

```

```

0059      JJ=N-J+II
0060      DO 120 K=II,N
0061      H<II,JJ>=H<II,JJ>-C<II,K>*H<K,JJ>
0062      120 CONTINUE
0063      H<JJ,II>=H<II,JJ>
0064      130 CONTINUE
0065      150 CONTINUE
0066      IF (ISCALE .NE. 0) RETURN
0067      DO 220 I=1,N
0068      GSS=SCALEF/SCALEX(I)
0069      DELTA=DELTA+GSS/SCALEX(I)
0070      DO 210 J=1,N
0071      H<I,J>=H<I,J>*GSS/SCALEX(J)
0072      210 CONTINUE
0073      220 CONTINUE
0074      RETURN
0075      END
0076      SUBROUTINE MATPY(M,N,H,G,S)
0077      DIMENSION H(40,40),G(40),S(40)
0078      IF (M-1) 705,705,702
0079      702 DO 703 I=1,M
0080      S(I)=0.0
0081      DO 703 J=1,N
0082      703 S(I)=H<I,J>*G(J)+S(I)
0083      RETURN
0084      705 S(1)=0.0
0085      DO 706 I=1,N
0086      706 S(1)=H<I,1>*G(I)+S(1)
0087      RETURN
0088      END
0089      FUNCTION OK(X)
0090      OK=ABS(X)
0091      IF(OK.GE.1.0E20)GO TO 12
0092      OK=0.0
0093      RETURN
0094      12 OK=1.0
0095      RETURN
0096      END

```

LCRVHH T=00004 IS ON CR00026 USING 00013 BLKS R=0000

```

0001 FTH4,L
0002 SUBROUTINE FCN (N,DER,CHISQ,X,M11)
0003 DIMENSION DER(40),GRADT(40),XDATA(2,200),YDATA( 500),WEIGHT(500),
0004 * YCALC( 500),X(40),BUF(128)
0005 COMMON /TAPE0/ IOUTAP
0006 COMMON /IOBLK/ INPUT(144),NIN(3),IBUF(128)
0007 M1=M11
0008 IF (M1 .NE. 1) GO TO 400
0009 333 CALL READF(INPUT,IERR,IBUF,30,LEN)
0010 N=IBUF(1)
0011 INVAR=IBUF(2)
0012 NUMDTA=IBUF(3)
0013 IWGT=IBUF(4)
0014 IDIFER=IBUF(5)
0015 IGRAPH=IBUF(6)
0016 IF(N.LE.0)CALL FLCLS
0017 C IF(IGRAPH.NE.2)GO TO 777
0018 C IGRAPH=0
0019 WRITE (IOUTAP,7) N,INVAR,NUMDTA
0020 7 FORMAT (35H1THE NUMBER OF FITTING PARAMETERS =,I4/38H THE NUMBER O
0021 *F INDEPENDENT VARIABLES =,I4/28H THE NUMBER OF DATA POINTS =,I4)
0022 IF (MOD(IWGT,2) .EQ. 1) GO TO 100
0023 DO 10 I=1,NUMDTA
0024 CALL READF(INPUT,IERR,BUF,40,LEN)
0025 YDATA(I)=BUF(1)
0026 DO 9 J=1,INVAR
0027 XDATA(J,I)=BUF(J+1)
0028 9 CONTINUE
0029 IF(YDATA(I).GT.0.)GO TO 8
0030 WEIGHT(I)=1.0
0031 GO TO 10
0032 8 WEIGHT(I)=1.0/YDATA(I)
0033 10 CONTINUE
0034 GO TO 200
0035 100 DO 110 I=1,NUMDTA
0036 CALL READF(INPUT,IERR,BUF,40,LEN)
0037 YDATA(I)=BUF(1)
0038 DO 105 J=1,INVAR
0039 XDATA(J,I)=BUF(J+1)
0040 105 CONTINUE
0041 WEIGHT(I)=BUF(INVAR+2)
0042 110 CONTINUE
0043 200 IF (IWGT .GE. 2) GO TO 300
0044 C WRITE (IOUTAP,201)
0045 C 201 FORMAT (12H0DATA POINTS)
0046 C DO 210 I=1,NUMDTA
0047 C WRITE (IOUTAP,205) YDATA(I),(XDATA(J,I),J=1,INVAR),WEIGHT(I)
0048 C 205 FORMAT (1X,10E13.5)
0049 C 210 CONTINUE
0050 300 CALL TABLE (FUNCT,GRADT,X,XDATA,M1,INVAR)
0051 RETURN
0052 C 777 WRITE(1,901)(IBUF(LL),LL=1,6)
0053 C 901 FORMAT(6I10)
0054 C DO 780 I=1,NUMDTA
0055 C CALL READF(INPUT,IERR,BUF,40,LEN)
0056 C 780 CONTINUE
0057 C CALL READF(INPUT,IERR,X,50,LEN)
0058 C GO TO 333

```

```

0059      JJ=M-J+11
0060      DO 120 K=11,N
0061      H<11,JJ>=H<11,JJ>-C<11,K>*H<K,JJ>
0062 120 CONTINUE
0063      H<JJ,11>=H<11,JJ>
0064 130 CONTINUE
0065 150 CONTINUE
0066      IF (ISCALE.NE. 0) RETURN
0067      DO 220 I=1,N
0068      GSS=SCALEF/SCALEX<I>
0069      DELTA=DELTA+GSS/SCALEX<I>
0070      DO 210 J=1,N
0071      H<I,J>=H<I,J>+GSS/SCALEX<J>
0072 210 CONTINUE
0073 220 CONTINUE
0074      RETURN
0075      END
0076      SUBROUTINE MATPY<M,N,H,G,S>
0077      DIMENSION H<40,40>,G<40>,S<40>
0078      IF (M-1) 705,705,702
0079 702 DO 703 I=1,M
0080      S<I>=0.0
0081      DO 703 J=1,N
0082 703 S<I>=H<I,J>*G<J>+S<I>
0083      RETURN
0084 705 S<1>=0.0
0085      DO 706 I=1,N
0086 706 S<1>=H<1,I>*G<I>+S<1>
0087      RETURN
0088      END
0089      FUNCTION OK<X>
0090      OK=ABS<X>
0091      IF<OK.GE.1.0E20>GO TO 12
0092      OK=0.0
0093      RETURN
0094 12 OK=1.0
0095      RETURN
0096      END

```

4.3 UTLTY LISTING

UTLTY T=00004 IS ON CR00026 USING 00040 BLKS R=0000

```

0001 FTM4,L
0002     PROGRAM UTLTY
0003 C   PROGRAM TO CONVERT A SET OF 22 CALCULATED ENERGY DISTRIBUTION
0004 C   RESPONSES TO AN INPUT FILE FOR THE 3-DIMENSIONAL SMOOTHING
0005 C   ROUTINE LSQVM.
0006     DIMENSION CONT1(3,15,5),CONT2(3,10,5),CONT3(3,15,7),CONT4(3,10,7),
0007     A CONT5(3,15,10),CONT6(3,10,10),CONT7(3,5,10),CONT8(3,5,10),
0008     B PEAK1(2,12),PEAK2(2,10),PEAK3(2,10),PEAK4(2,10)
0009     DIMENSION X(3),DATA(100),DATA(20),TITL(6),INDX(6),XXI(25)
0010     COMMON/IOBLK/INPUT(144),IOUT(144),NIN(3),NOUT(3),ITL(144),NI2(3)
0011     DATA EM/0.311/
0012     NLE=0
0013     NL2=0
0014     NHE=0
0015     CALL OPEN(INPUT,IERR,NIN,0,0,26,128)
0016     IF(IERR.LT.0)GO TO 701
0017     GO TO 51
0018 701 WRITE(1,721)
0019 721 FORMAT(' CANNOT OPEN FILE....OUT01')
0020     GO TO 500
0021 51 DO 400 KT=1,22
0022 120 CALL READF(INPUT,IERR,TITL,6)
0023     IF(IERR.LT.0)GO TO 500
0024     CALL READF(INPUT,IERR,DATA,200)
0025     WRITE(1,101)KT,(TITL(K),K=1,3)
0026 101 FORMAT(15,3E15.5)
0027     E=TITL(1)
0028     IF(E.LT.0.2)GO TO 120
0029     X(2)=E
0030     SLPX=E/100.
0031     EZOX=-SLPX/2.0
0032     ECPT=E/(1.0+EM/(2.0+E))
0033     IF(E.GT.1.2)GO TO 200
0034     NLE=NLE+1
0035     SLP=ECPT/15.0
0036     EZO=-SLP/2.0
0037     CR=ECPT
0038     SUB=0.
0039     NR=15
0040     NX=3
0041     NZ=5
0042     IF(E.GT.0.6)GO TO 150
0043     CALL MANIP(NR,NX,NZ,NLE,DATA,SLPX,EZOX,DATA,SLP,EZO,X,CR,
0044     A SUB,CONT1)
0045 155 SLP=(E-ECPT)/11.0
0046     EZO=ECPT-SLP/2.0
0047     CR=E-ECPT
0048     SUB=ECPT
0049     NR=10
0050     IF(E.GT.0.6)GO TO 160
0051     CALL MANIP(NR,NX,NZ,NLE,DATA,SLPX,EZOX,DATA,SLP,EZO,X,CR,
0052     A SUB,CONT2)
0053 165 PEAK1(1,NLE)=DATA(100)
0054     PEAK1(2,NLE)=E
0055     GO TO 400
0056 150 NZ=7
0057     NL2=NL2+1
0058     CALL MANIP(NR,NX,NZ,NL2,DATA,SLPX,EZOX,DATA,SLP,EZO,X,CR,

```



```

0059      A SUB,CONT3)
0060      GO TO 155
0061 160 CALL MANIP(NR,NX,NZ,HL2,DATA,X,SLPX,EZO,X,CR,
0062      A SUB,CONT4)
0063      GO TO 165
0064 200 NHE=NHE+1
0065      SLP=(E-2.0*EM)/15.5
0066      EZO=-SLP/2.0
0067      CR=E-2.0*EM
0068      SUB=0.
0069      NZ=10
0070      NR=15
0071      CALL MANIP(NR,NX,NZ,NHE,DATA,X,SLPX,EZO,X,CR,
0072      A SUB,CONT5)
0073      SLP=EM/11.5
0074      EZO=E-2.0*EM
0075      CR=EM
0076      SUB=EZO
0077      NR=10
0078      CALL MANIP(NR,NX,NZ,NHE,DATA,X,SLPX,EZO,X,CR,
0079      A SUB,CONT6)
0080      SLP=(ECPT-E+EM)/5.5
0081      EZO=(E-EM)
0082      CR=ECPT-E+EM
0083      SUB=EZO
0084      NR=5
0085      CALL MANIP(NR,NX,NZ,NHE,DATA,X,SLPX,EZO,X,CR,
0086      A SUB,CONT7)
0087      SLP=(0.99*E-ECPT)/5.0
0088      EZO=ECPT-SLP/2.0
0089      CR=E-ECPT
0090      SUB=ECPT
0091      CALL MANIP(NR,NX,NZ,NHE,DATA,X,SLPX,EZO,X,CR,
0092      A SUB,CONT8)
0093      N1=100.0*(E-2.0*EM)/E-2.0
0094      N2=100.0*(E-EM)/E-2.0
0095      DMAX=DATA(N1)
0096      DO 250 I=N1+1,N1+4
0097      IF(DMAX.LT.DATA(I))DMAX=DATA(I)
0098 250 CONTINUE
0099      PEAK2(1,NHE)=DMAX
0100      PEAK2(2,NHE)=E
0101      DMAX=DATA(N2)
0102      DO 252 I=N2+1,N2+4
0103      IF(DMAX.LT.DATA(I))DMAX=DATA(I)
0104 252 CONTINUE
0105      PEAK3(1,NHE)=DMAX
0106      PEAK3(2,NHE)=E
0107      PEAK4(1,NHE)=DATA(100)
0108      PEAK4(2,NHE)=E
0109 400 CONTINUE
0110      CALL CREAT(IOUT,IERR,NOUT,100,3,0,26,128)
0111      IF (IERR.LT.0) GO TO 702
0112      GO TO 421
0113 702 WRITE(1,106)IERR
0114 106 FORMAT(' CREAT ERROR CODE=' I6)
0115      GO TO 500
0116 421 CALL OPEN(ITL,IERR,N12,0,0,26,128)
0117      IF (IERR.LT.0) GO TO 703
0118      GO TO 425

```

```

0119      703 WRITE(1,722)
0120      722 FORMAT(' CANNOT OPEN FILE....INPT02')
0121      GO TO 500
0122      425 INDX(1)=25
0123          INDX(2)=2
0124          INDX(4)=0
0125          INDX(5)=1
0126          INDX(6)=0
0127          NI=5
0128          NJ=15
0129          INDX(3)=NI*NJ
0130          WRITE(1,102)(INDX(I),I=1,3)
0131      102 FORMAT(3I5)
0132          CALL WRITE(IOUT,IERR,INDX,6)
0133          CALL RTOUT(NI,NJ,CONT1)
0134          CALL READF(ITL,IERR,XXI,50,NW)
0135          CALL WRITE(IOUT,IERR,XXI,NW)
0136          NJ=10
0137          INDX(3)=NI*NJ
0138          WRITE(1,102)(INDX(I),I=1,3)
0139          CALL WRITE(IOUT,IERR,INDX,6)
0140          CALL RTOUT(NI,NJ,CONT2)
0141          CALL READF(ITL,IERR,XXI,50,NW)
0142          CALL WRITE(IOUT,IERR,XXI,NW)
0143          NI=7
0144          NJ=15
0145          INDX(3)=NI*NJ
0146          WRITE(1,102)(INDX(I),I=1,3)
0147          CALL WRITE(IOUT,IERR,INDX,6)
0148          CALL RTOUT(NI,NJ,CONT3)
0149          CALL READF(ITL,IERR,XXI,50,NW)
0150          CALL WRITE(IOUT,IERR,XXI,NW)
0151          NJ=10
0152          INDX(3)=NI*NJ
0153          WRITE(1,102)(INDX(I),I=1,3)
0154          CALL WRITE(IOUT,IERR,INDX,6)
0155          CALL RTOUT(NI,NJ,CONT4)
0156          CALL READF(ITL,IERR,XXI,50,NW)
0157          CALL WRITE(IOUT,IERR,XXI,NW)
0158          NI=10
0159          NJ=15
0160          INDX(3)=NI*NJ
0161          WRITE(1,102)(INDX(I),I=1,3)
0162          CALL WRITE(IOUT,IERR,INDX,6)
0163          CALL RTOUT(NI,NJ,CONT5)
0164          CALL READF(ITL,IERR,XXI,50,NW)
0165          CALL WRITE(IOUT,IERR,XXI,NW)
0166          NJ=10
0167          INDX(3)=NI*NJ
0168          WRITE(1,102)(INDX(I),I=1,3)
0169          CALL WRITE(IOUT,IERR,INDX,6)
0170          CALL RTOUT(NI,NJ,CONT6)
0171          CALL READF(ITL,IERR,XXI,50,NW)
0172          CALL WRITE(IOUT,IERR,XXI,NW)
0173          NJ=5
0174          INDX(3)=NI*NJ
0175          WRITE(1,102)(INDX(I),I=1,3)
0176          CALL WRITE(IOUT,IERR,INDX,6)
0177          CALL RTOUT(NI,NJ,CONT7)
0178          CALL READF(ITL,IERR,XXI,50,NW)

```

```

0179      CALL WRITF(IOUT,IERR,XXI,NW)
0180      WRITE(1,102)(INDX(I),I=1,3)
0181      CALL WRITF(IOUT,IERR,INDX,6)
0182      CALL RTOUT(NI,NJ,CONT8)
0183      CALL READF(ITL,IERR,XXI,50,NW)
0184      CALL WRITF(IOUT,IERR,XXI,NW)
0185      NI=12
0186      INDX(1)=5
0187      INDX(2)=1
0188      INDX(3)=NI
0189      WRITE(1,102)(INDX(I),I=1,3)
0190      CALL WRITF(IOUT,IERR,INDX,6)
0191      CALL RTTWO(NI,PEAK1)
0192      CALL READF(ITL,IERR,XXI,10,NW)
0193      CALL WRITF(IOUT,IERR,XXI,NW)
0194      NI=10
0195      INDX(3)=NI
0196      WRITE(1,102)(INDX(I),I=1,3)
0197      CALL WRITF(IOUT,IERR,INDX,6)
0198      CALL RTTWO(NI,PEAK2)
0199      CALL READF(ITL,IERR,XXI,10,NW)
0200      CALL WRITF(IOUT,IERR,XXI,NW)
0201      WRITE(1,102)(INDX(I),I=1,3)
0202      CALL WRITF(IOUT,IERR,INDX,6)
0203      CALL RTTWO(NI,PEAK3)
0204      CALL READF(ITL,IERR,XXI,10,NW)
0205      CALL WRITF(IOUT,IERR,XXI,NW)
0206      WRITE(1,102)(INDX(I),I=1,3)
0207      CALL WRITF(IOUT,IERR,INDX,6)
0208      CALL RTTWO(NI,PEAK4)
0209      CALL READF(ITL,IERR,XXI,10,NW)
0210      CALL WRITF(IOUT,IERR,XXI,NW)
0211      INDX(1)=0
0212      WRITE(1,102)(INDX(I),I=1,3)
0213      CALL WRITF(IOUT,IERR,INDX,6)
0214      CALL LOCF(IOUT,IERR,IREF,IRB,IOFF,JSEC)
0215      ITRUN=JSEC/2-IRB-1
0216      WRITE(1,102)ITRUN
0217      CALL CLOSE(IOUT,IERR,ITRUN)
0218      WRITE(1,103)
0219      103 FORMAT(' FINIS ')
0220      500 CALL CLOSE(INPUT)
0221      CALL CLOSE(ITL)
0222      STOP
0223      END
0224 C
0225      SUBROUTINE RTOUT(NI,NJ,ARRAY)
0226      COMMON/IOBLK/INPUT(144),IOUT(144),NIN(3),NOUT(3),ITL(144),NI2(3)
0227      DIMENSION ARRAY(3,NJ,NI),BUF(5)
0228      DO 10 I=1,NI
0229      DO 10 J=1,NJ
0230      DO 5 K=1,3
0231      BUF(K)=ARRAY(K,J,I)
0232      5 CONTINUE
0233      CALL WRITF(IOUT,IERR,BUF,6)
0234      10 CONTINUE
0235      RETURN
0236      END
0237 C
0238      SUBROUTINE RTTWO(NI,ARRAY)

```

```

0239      COMMON/IOBLK/INPUT(144),IOUT(144),NIN(3),NOUT(3),ITL(144),NI2(3)
0240      DIMENSION ARRY(2,1),BUF(5)
0241      DO 10 I=1,NI
0242      DO 5 J=1,2
0243      5 BUF(J)=ARRY(J,I)
0244      CALL WRITF(IOUT,IERR,BUF,4)
0245      10 CONTINUE
0246      RETURN
0247      END
0248  C
0249      BLOCK DATA
0250      COMMON/IOBLK/INPUT(144),IOUT(144),NIN(3),NOUT(3),ITL(144),NI2(3)
0251      DATA NIN/2H0U,2HT0,2H1 /
0252      DATA NI2/2HIN,2HPT,2H02/
0253      DATA NOUT/2HLS,2HQ1,2HN /
0254      END
0255  C
0256      SUBROUTINE MANIP(NR,NX,NZ,NDX,DATA,X,CR,SUB,OUT)
0257      1 X,CR,SUB,OUT)
0258      DIMENSION DATA(1),DAT(1),X(1),EDAT(20),OUT(NX,NR,NZ)
0259      CALL BINDT(NR,EDAT,SLP,EZO)
0260      DO 10 I=1,NR
0261      EDAT(I)=(EDAT(I)-SUB)/CR
0262      10 CONTINUE
0263      CALL REBIN(NR,DATA,SLPX,EZOX,DATA,SLP,EZO)
0264      CALL SAVIT(NR,NX,NZ,NDX,X,DATA,EDAT,OUT)
0265      RETURN
0266      END
0267  C
0268      SUBROUTINE SAVIT(NR,NX,NZ,NDX,X,DATA,EDAT,OUT)
0269      DIMENSION X(1),DATA(1),EDAT(1),OUT(NX,NR,NZ)
0270      DO 10 I=1,NR
0271      X(1)=DATA(I)
0272      X(3)=EDAT(I)
0273      DO 5 J=1,3
0274      OUT(J,I,NDX)=X(J)
0275      5 CONTINUE
0276      10 CONTINUE
0277      RETURN
0278      END
0279  C
0280      SUBROUTINE REBIN(NR,DATA,SLPX,EZOX,DATA,SLP,EZO)
0281  C TRANSFORM 100-DIMENSIONAL DATA(SLPX,EZOX)
0282  C INTO NR-DIMENSIONAL DATA(SLP,EZO)
0283      DIMENSION DATA(1),DATA(1)
0284      RTIO=CLPX/6LP
0285      CHNL=1.5
0286      CALL ERG(EL0,CHNL-1.0,SLP,EZO)
0287      CALL ERG(EH1,CHNL,SLP,EZO)
0288      DO 100 I=1,NR
0289      DATA(I)=0.0
0290      CHNLX=1.5
0291      DO 101 IX=1,100
0292      CALL ERG(EXHI,CHNLX,SLPX,EZOX)
0293      IF(EXHI.GT.EL0)GO TO 102
0294      CHNLX=CHNLX+1.0
0295      101 CONTINUE
0296      GO TO 500
0297  C
0298      102 CALL ERG(EXL0,CHNLX-1.0,SLPX,EZOX)

```

```

0299      DO 103 I=1,NR
0300      IMIN=I
0301      IF(EHI.GT.EXLO)GO TO 104
0302      ELO=EHI
0303      CHNL=CHNL+1.0
0304      103 CALL ERG(EHI,CHNL,SLP,EZO)
0305      GO TO 500
0306  C
0307      104 DO 400 I=IMIN,NR
0308      105 IF(EXLO.GT.ELO) GO TO 201
0309      200 IF(EXHI.GT.EHI) GO TO 210
0310      W=(EXHI-ELO)/(EXHI-EXLO)
0311      DATA(I)=DATA(I)+W*DATA(XIX)
0312      GO TO 300
0313      201 IF(EXHI.GT.EHI)GO TO 211
0314      DATA(I)=DATA(I)+DATA(XIX)
0315      GO TO 300
0316      210 W=(EHI-ELO)/(EXHI-EXLO)
0317      DATA(I)=DATA(I)+W*DATA(XIX)
0318      GO TO 301
0319      211 W=(EHI-EXLO)/(EXHI-EXLO)
0320      DATA(I)=DATA(I)+W*DATA(XIX)
0321      GO TO 301
0322      300 IX=IX+1
0323      IF(IX.GT.100)GO TO 500
0324      EXLO=EXHI
0325      CHNLX=CHNLX+1.0
0326      CALL ERG(EXHI,CHNLX,SLPX,EZOX)
0327      GO TO 105
0328      301 ELO=EHI
0329      CHNL=CHNL+1.0
0330      CALL ERG(EHI,CHNL,SLP,EZO)
0331      400 CONTINUE
0332      500 DO 550 I=1,NR
0333      550 DATA(I)=DATA(I)*RTIO
0334      RETURN
0335      END
0336  C
0337      SUBROUTINE ERG(E,CHNL,SLOPE,EZERO)
0338  C  COMPUTE CHANNEL CENTER ENERGY
0339      E=SLOPE*CHNL+EZERO
0340      RETURN
0341      END
0342  C
0343      SUBROUTINE BINDT(NR,EDAT,SLOPE,EZERO)
0344  C  CONSTRUCT CHANNEL CENTER ENERGY VECTOR EDAT
0345      DIMENSION EDAT(1)
0346      DO 2 I=1,NR
0347      CHNL=I
0348      CALL ERG(EDAT(I),CHNL,SLOPE,EZERO)
0349      2 CONTINUE
0350      RETURN
0351      END

```


4.4 RESPN LISTING

&RESPN T=00004 IS ON CR00026 USING 00014 BLKS R=0000

```

0001 FTH4,L
0002 PROGRAM RESPN
0003 C THIS PROGRAM WILL WRITE AN OUTPUT FILE CONTAINING THE RESPONSE.
0004 C IT EXPECTS TWO INPUT FILES...<1>OUT02, <2>INPT03.
0005 C FILE OUT02 MUST CONTAIN THE BEST FIT PARAMETERS.
0006 C FILE INPT03 MUST CONTAIN PHOTON ENERGY, SLOPE AND EZERO.
0007 C CHOOSE SLOPE AND EZERO SUCH THAT THE RESPONSE OUTPUT HAS
0008 C LESS THAN 256 CHANNELS.
0009 C THE LAST ENTRY IN INPT03 INPUT FILE HAS TO HAVE ENERGY = 0.0
0010 DIMENSION X1(25),X2(25),X3(25),X4(25),X5(25),X6(25),X7(25),X8(25)
0011 DIMENSION P1(5),P2(5),P3(5),P4(5)
0012 DIMENSION IOUT(144),NOUT(3),INP1(144),NP1(3),INP2(144),NP2(3)
0013 DIMENSION BUF(6),RESP(256)
0014 DATA NOUT/2HOU,2HTP,2HT /
0015 DATA NP1/2HOU,2HT0,2H2 /
0016 DATA NP2/2HIN,2HPT,2H03/
0017 DATA EM/0.511/
0018 CALL CREAT(IOUT,IERR,NOUT,30,3,0,26,128)
0019 CALL OPEN(INP1,IERR,NP1,0,0,26,128)
0020 CALL OPEN(INP2,IERR,NP2,0,0,26,128)
0021 CALL READF(INP1,IERR,X1,50,LEN)
0022 CALL READF(INP1,IERR,X2,50,LEN)
0023 CALL READF(INP1,IERR,X3,50,LEN)
0024 CALL READF(INP1,IERR,X4,50,LEN)
0025 CALL READF(INP1,IERR,X5,50,LEN)
0026 CALL READF(INP1,IERR,X6,50,LEN)
0027 CALL READF(INP1,IERR,X7,50,LEN)
0028 CALL READF(INP1,IERR,X8,50,LEN)
0029 CALL READF(INP1,IERR,P1,10,LEN)
0030 CALL READF(INP1,IERR,P2,10,LEN)
0031 CALL READF(INP1,IERR,P3,10,LEN)
0032 CALL READF(INP1,IERR,P4,10,LEN)
0033 1001 CALL READF(INP2,IERR,BUF,6,LEN)
0034 E=BUF(1)
0035 IF(E.LE.0.)GO TO 2001
0036 DE=BUF(2)
0037 EZ0=BUF(3)
0038 NR=(E-EZ0)/DE+0.5
0039 IF(NR.GT.256)GO TO 701
0040 ECMPT=E/(1.0+EM/(2.0+E))
0041 DO 10 I=1,256
0042 10 RESP(I)=0.0
0043 IF(E.GT.1.2)GO TO 50
0044 DO 40 I=1,NR-1
0045 CI=I
0046 ECH=(CI-0.5)*DE+EZ0
0047 IF(ECH.LE.0.)GO TO 40
0048 IF(ECH.GE.ECMPT)GO TO 35
0049 ECH=ECH/ECMPT
0050 IF(E.GT.0.6)GO TO 36
0051 CALL EDEP(E,ECH,RESP(I),X1)
0052 GO TO 40
0053 36 CALL EDEP(E,ECH,RESP(I),X3)
0054 GO TO 40
0055 35 ECH=(ECH-ECMPT)/(E-ECMPT)
0056 IF(E.GT.0.6)GO TO 38
0057 CALL EDEP(E,ECH,RESP(I),X2)
0058 GO TO 40

```



```

0059      38 CALL EDEP(E,ECH,RESP(1),X4)
0060      40 CONTINUE
0061      CALL PEAKE(E,RESP(NR),P1)
0062      GO TO 1101
0063      50 E1=E-2.0*EM
0064      E2=E-EM
0065      DO 100 I=1,NR-1
0066      CI=I
0067      ECH=(CI-0.5)*DE+E20
0068      IF(ECH.LE.0.)GO TO 100
0069      IF(ECH.GT.E1)GO TO 60
0070      ECH=ECH/E1
0071      CALL EDEP(E,ECH,RESP(1),X5)
0072      GO TO 100
0073      60 IF(ECH.GT.E2)GO TO 70
0074      ECH=(ECH-E1)/EM
0075      CALL EDEP(E,ECH,RESP(1),X6)
0076      GO TO 100
0077      70 IF(ECH.GT.ECMPT)GO TO 80
0078      ECH=(ECH-E2)/(ECMPT-E2)
0079      CALL EDEP(E,ECH,RESP(1),X7)
0080      GO TO 100
0081      80 ECH=(ECH-ECMPT)/(E-ECMPT)
0082      CALL EDEP(E,ECH,RESP(1),X8)
0083      100 CONTINUE
0084      NP1=E1/DE+0.5
0085      NP2=E2/DE+0.5
0086      CALL PEAKE(E,RESP(NP1),P2)
0087      CALL PEAKE(E,RESP(NP2),P3)
0088      CALL PEAKE(E,RESP(NR),P4)
0089      1101 CALL WRITF(IOUT,IERR,BUF,6)
0090      WRITE(6,201)(BUF(I),I=1,3)
0091      201 FORMAT(3E12.5)
0092      NWRD=2*NR
0093      CALL WRITF(IOUT,IERR,RESP,NWRD)
0094      WRITE(6,202)(RESP(I),I=1,NR)
0095      202 FORMAT(10F7.0)
0096      GO TO 1001
0097      701 WRITE(1,721)
0098      721 FORMAT(' OUTPUT RESPONSE HAS MORE THAN 256 CHANNELS' /
0099      A' CHECK SLOPE AND EZERO VALUES IN FILE INPT03')
0100      2001 CALL LOCF(IOUT,IERR,IERR,IERR,IERR,IERR,IERR,IERR,IERR,IERR)
0101      ITRUN=JSEC/2-IRB-1
0102      CALL CLOSE(IOUT,IERR,ITRUN)
0103      CALL CLOSE(INP1)
0104      CALL CLOSE(INP2)
0105      STOP
0106      END
0107      C
0108      C
0109      SUBROUTINE EDEP(E,ECH,RESP,X)
0110      DIMENSION G(25),X(1)
0111      ECHS=ECH*ECH
0112      ECHC=ECHS*ECH
0113      ECHQ=ECHC*ECH
0114      G(1)=1.0
0115      G(2)=ECH
0116      G(3)=0.5*(3.0*ECHS-1.0)
0117      G(4)=0.5*(5.0*ECHC-3.0*ECH)
0118      G(5)=0.125*(35.0*ECHQ-30.0*ECHS+3.0)

```

```

0119      DO 10 I=6,25
0120      G(I)=G(I-5)*E
0121      10 CONTINUE
0122      RESP=0.0
0123      DO 20 I=1,25
0124      RESP=RESP+X(I)*G(I)
0125      20 CONTINUE
0126      RETURN
0127      END
0128      C
0129      C
0130      SUBROUTINE PEAKE(E,RESP,P)
0131      DIMENSION P(1)
0132      ES=E*E
0133      EC=ES+E
0134      EQ=EC+E
0135      RESP=P(1)+P(2)*E+P(3)*ES+P(4)*EC+P(5)*EQ
0136      RETURN
0137      END

```

4.5 PLTTR LISTING

&PLTTR T=00003 IS ON CR00026 USING 0021 BLKS R=0000

```

0001 FTH4,L
0002 $EMA(DATT,0)
0003 PROGRAM PLTTR
0004 COMMON/DATT/DATA(10,280
0005 COMMON SLOPE,EZERO,RESO POWER
0006 C THIS PROGRAM WILL PLOT RES ONSE FUNCTIONS CALCULATED BY ARES,
0007 C AFTER FOLDING IN THE DETEC OR RESOLUTION.
0008 C PROGRAM WILL ASK FOR DETEC OR RESOLUTION AND POWER.
0009 C IT EXPECTS AN OUTPUT FILE REATED BY ARES MODULE RESPH NAMED 'OUTPT'.
0010 DIMENSION IGCB(192),ATA(280),NIN(3),INPT(144),BUF(128),NCH(10)
0011 DATA NIN/2HOU,2HTP,2HT
0012 LU=60
0013 ID=4
0014 CALL OPEN(INPT,IERR,NIN 0,0,26,128)
0015 IF(IERR.LT.0)GO TO 951
0016 CALL PLOT(IIGCB,ID,1,LU
0017 WRITE(1,900)
0018 900 FORMAT(' ENTER NUMBER 0 RESPONSES TO PLOT')
0019 READ(1,*)NPL
0020 DO 10 KK=1,NPL
0021 CALL READF(INPT,IERR,BU ,6)
0022 IF(IERR.LT.0)GO TO 20
0023 WRITE(1,901)(BUF(I),I=1 3)
0024 901 FORMAT(3E12.5)
0025 CALL READF(INPT,IERR,AT ,512,LEN)
0026 NN=LEN/2
0027 NCH(KK)=NN
0028 DO 5 J=1,NN
0029 DATA(KK,J)=ATA(J)
0030 5 CONTINUE
0031 10 CONTINUE
0032 20 WRITE(1,902)
0033 902 FORMAT(' ENTER SLOPE,AN EZERO')
0034 READ(1,*)SLOPE,EZERO
0035 WRITE(1,903)
0036 903 FORMAT(' ENTER PERCENT ESOLUTION, AND POWER')
0037 READ(1,*)RESO,POWER
0038 DO 30 IS=1,NPL
0039 NR=NCH(IS)
0040 DO 25 I=1,NR
0041 ATA(I)=DATA(IS,I)
0042 25 CONTINUE
0043 CALL SMEAR(ATA,NR)
0044 NCH(IS)=NR
0045 DO 28 I=1,NR
0046 DATA(IS,I)=ATA(I)
0047 28 CONTINUE
0048 30 CONTINUE
0049 XMIN=12.4
0050 XMAX=185.2
0051 CALL LIMIT(IGCB,XMIN,XM X,0.0,254.0)
0052 CALL SETAR(IGCB,0.68)
0053 CALL WINDU(IGCB,0.,100. ,0.,147.0)
0054 CALL MOVE(IGCB,5.0,0.0)
0055 CALL DRAW(IGCB,100.,0.0
0056 CALL DRAW(IGCB,100.,147 0)
0057 CALL DRAW(IGCB,5.0,147. )
0058 CALL DRAW(IGCB,5.0,0.0)

```

```

0059      DMIN=1.0E+7
0060      DMAX=0.0
0061      DO 170 I=1,NPL
0062      DO 170 J=1,NCH(I)
0063      IF(DATA(I,J).LE.0.)DATA(I,J)=1.0
0064      IF(DMIN.GT.DATA(I,J))DMIN=DATA(I,J)
0065      IF(DMAX.LT.DATA(I,J))DMAX=DATA(I,J)
0066 170 CONTINUE
0067      MIN=0
0068 181 IF(10.0**MIN.LT.DMIN)GO TO 182
0069      MIN=MIN+1
0070      GO TO 181
0071 182 MIN=MIN+1
0072      IF(10.0**MIN.GT.DMIN)GO TO 183
0073      GO TO 182
0074 183 MIN=MIN-1
0075      DMIN=MIN
0076      MAX=MIN
0077 185 IF(10.0**MAX.GT.DMAX)GO TO 186
0078      MAX=MAX+1
0079      GO TO 185
0080 186 DMAX=MAX
0081 187 NX=MAX-MIN
0082      NR=NCH(NPL)
0083      NY=NR/20+1
0084      DO 190 I=1,NPL
0085      DO 190 J=1,NCH(I)
0086      DATA(I,J)=ALOGT(DATA(I,J))
0087 190 CONTINUE
0088      CALL YTICS(IGCB,20.0,5.0,137.0,NY)
0089      CALL XTICS(IGCB,20.0,90.0,137.0,NX)
0090      CALL VIEWP(IGCB,20.0,90.0,5.0,137.0)
0091      AR=NY*20
0092      CALL WINDW(IGCB,DMIN,DMAX,0.,AR)
0093      CALL CSIZE(IGCB,2.0,0.7,0.)
0094      CALL LDIR(IGCB,-1.57)
0095      DO 200 I=1,NPL
0096      CALL MOVE(IGCB,(DATA(I,1)),AR)
0097      DO 200 J=2,NCH(I)
0098      CH=J-1
0099      BR=AR-CH
0100 200 CALL DRAW(IGCB,(DATA(I,J)),BR)
0101      WRITE(1,917)
0102 917 FORMAT('PLOT FINIS')
0103 959 CALL PLOTB(IGCB,ID,0)
0104      CALL CLOSE(INPT)
0105      STOP
0106 951 WRITE(1,911)IERR
0107 911 FORMAT(' OPEN ERROR =',I4)
0108      GO TO 959
0109      END
0110 C
0111      SUBROUTINE XTICS(IGCB,XMIN,XMAX,YMAX,NX)
0112      DIMENSION IGCB(1)
0113      DX=(XMAX-XMIN)/NX
0114      XTIC=YMAX-1.0
0115      X=XMIN
0116      DO 10 I=1,NX
0117      X=X+DX
0118      CALL DRAW(IGCB,X,YMAX)

```

```

0119      CALL DRAW(IGCB,X,XTIC)
0120      CALL MOVE(IGCB,X,YMAX)
0121      10 CONTINUE
0122      RETURN
0123      END
0124      C
0125      SUBROUTINE YTICS(IGCB,XMIN,YMIN,YMAX,NY)
0126      DIMENSION IGCB(1)
0127      DY=(YMAX-YMIN)/NY
0128      YTIC=XMIN+1.0
0129      Y=YMIN
0130      DO 10 I=1,NY
0131      CALL MOVE(IGCB,YTIC,Y)
0132      CALL DRAW(IGCB,XMIN,Y)
0133      Y=Y+DY
0134      CALL DRAW(IGCB,XMIN,Y)
0135      10 CONTINUE
0136      RETURN
0137      END
0138      C
0139      C
0140      SUBROUTINE SMEAR(PUTIN,NR)
0141      DIMENSION PUTIN(280),OUT(280),E(280)
0142      COMMON SLOPE,EZERO,RESO,POWER
0143      DO 10 I=1,280
0144      10 OUT(I)=0.
0145      TOTIN=0.0
0146      DO 25 I=1,NR
0147      TOTIN=TOTIN+PUTIN(I)
0148      25 CONTINUE
0149      CALL EDAT(E)
0150      DO 50 K=1,NR
0151      SIG=SIGMA(E(K))
0152      Y=1.0/SIG
0153      ND=5.0*SIG/SLOPE
0154      NI=K-ND
0155      IF(NI.LT.1)NI=1
0156      NF=K+ND
0157      IF(NF.GT.280)NF=280
0158      DO 30 I=NI,NF
0159      X=(E(K)-E(I))*Y
0160      GAUSS=Y*EXP(-0.5*X*X)
0161      OUT(I)=OUT(I)+GAUSS*PUTIN(K)
0162      30 CONTINUE
0163      50 CONTINUE
0164      TOTOT=0.0
0165      DO 60 I=1,NF
0166      TOTOT=TOTOT+OUT(I)
0167      60 CONTINUE
0168      RTIO=TOTIN/TOTOT
0169      DO 70 I=1,NF
0170      PUTIN(I)=OUT(I)*RTIO
0171      70 CONTINUE
0172      NR=NF
0173      RETURN
0174      END
0175      C
0176      C
0177      FUNCTION SIGMA(E)
0178      COMMON SLOPE,EZERO,RESO,POWER

```

```

0179      SIGMA=0.424661E-02*RES0*E*(0.6616/E)**POWER
0180      RETURN
0181      END
0182  C
0183  C
0184      SUBROUTINE EDAT(E)
0185      COMMON SLOPE,EZERO,RES0,POWER
0186      DIMENSION E(1)
0187      DO 10 I=1,280
0188      CI=I
0189      E(I)=(CI-0.5)*SLOPE+EZERO
0190  10 CONTINUE
0191      RETURN
0192      END

```

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

4.6 File Manipulating Routines

4.6.1 MERGE LISTING

&MERGE T=00004 IS ON CR00026 USING 00006 BLKS R=0000

```
0001 FTH4,L
0002 PROGRAM MERGE
0003 C PROGRAM FOR MERGING OUTPUT FILES OBTAINED FROM GAMRES.
0004 C THIS PROGRAM SHOULD BE USED TO CREATE THE COMBINED ENERGY
0005 C DEPOSITION FILE FOR THE 22 PHOTON ENERGIES.
0006 DIMENSION INPUT(144),NIN(3),IBUF(128),IOUT(144),MOUT(3)
0007 DATA YES/1HY/
0008 WRITE(1,101)
0009 101 FORMAT('ENTER OUTPUT FILE NAME')
0010 READ(1,102)MOUT
0011 102 FORMAT(3A2)
0012 CALL CREAT(IOUT,IERR,MOUT,40,3,0,26,128)
0013 IF(IERR.LT.0) GO TO 901
0014 10 WRITE(1,103)
0015 103 FORMAT('ENTER INPUT FILE NAME')
0016 READ(1,102)NIN
0017 CALL OPEN(INPUT,IERR,NIN,0,0,26,128)
0018 IF(IERR.LT.0)GO TO 902
0019 15 CALL READF(INPUT,IERR,IBUF,6,LEN)
0020 IF(LEN.LT.0)GO TO 20
0021 CALL WRITF(IOUT,IERR,IBUF,6)
0022 CALL READF(INPUT,IERR,IBUF,256,LEN)
0023 CALL WRITF(IOUT,IERR,IBUF,LEN)
0024 GO TO 15
0025 20 CONTINUE
0026 WRITE(1,104)
0027 104 FORMAT('ANY MORE FILES TO MERGE?')
0028 READ(1,105)YEP
0029 105 FORMAT(A1)
0030 IF(YEP.EQ.YES) GO TO 10
0031 GO TO 903
0032 901 WRITE(1,106)IERR
0033 106 FORMAT('CREAT ERROR =',I3)
0034 GO TO 903
0035 902 WRITE(1,107)IERR,NIN
0036 107 FORMAT('OPEN ERROR =',I3,' FILE NAME...',3A2)
0037 903 CALL LOCF(IOUT,IERR,IREF,IRB,IOFF,JSEC)
0038 ITRUN=JSEC/2-IRB-1
0039 CALL CLOSE(IOUT,IERR,ITRUN)
0040 CALL CLOSE(INPUT)
0041 STOP
0042 END
```

4.6.2 MKFL2 LISTING

&MKFL2 T=00004 IS ON CR00026 USING 00010 BLKS R=0000

```

0001  FTN4,L
0002      PROGRAM MKFL2
0003  C   THIS PROGRAM IS USED TO MAKE OR CHANGE THE BINARY INPUT FILE
0004  C   'INPT02' FOR INITIAL VALUES OF X'S
0005  C
0006  C
0007  C   IT WILL ALLOW TO CREATE A NEW FILE OR CHANGE VALUES IN AN
0008  C   EXISTING FILE.
0009      DIMENSION CONT(8,25),PEAK(4,5),INPT(144),NIN(3),BUF(128)
0010      DATA YS/1HY/
0011      DATA NIN/2HIN,2HPT,2H02/
0012      CALL OPEN(INPT,IERR,NIN,0,0,26,128)
0013      IF(IERR.GT.0)GO TO 5
0014      CALL CREAT(INPT,IERR,NIN,40,3,0,26,128)
0015      DO 10 I=1,8
0016      DO 10 J=1,25
0017      CONT(I,J)=0.0
0018  10 CONTINUE
0019      DO 15 I=1,4
0020      DO 15 J=1,5
0021      PEAK(I,J)=0.0
0022  15 CONTINUE
0023      GO TO 55
0024      5 DO 20 I=1,8
0025      CALL READF(INPT,IERR,BUF,50)
0026      DO 20 J=1,25
0027      CONT(I,J)=BUF(J)
0028  20 CONTINUE
0029      DO 25 I=1,4
0030      CALL READF(INPT,IERR,BUF,10)
0031      DO 25 J=1,5
0032      PEAK(I,J)=BUF(J)
0033  25 CONTINUE
0034      CALL RWNDF(INPT,IERR)
0035      55 DO 60 I=1,8
0036      WRITE(1,101)I
0037  101 FORMAT(' ANY INPUT VALUE FOR CONTINUUM',I3,' ?')
0038      GO TO 66
0039      65 WRITE(1,104)I
0040  104 FORMAT(' ANYMORE INPUT VALUE FOR CONTINUUM',I3,' ?')
0041      66 READ(1,102)YSNO
0042  102 FORMAT(A1)
0043      IF(YSNO.NE.YS)GO TO 60
0044      WRITE(1,103)
0045  103 FORMAT(' ENTER COMPONENT NO. AND VALUE')
0046      READ(1,*)J,CONT(I,J)
0047      GO TO 65
0048      60 CONTINUE
0049      DO 70 I=1,4
0050      WRITE(1,106)I
0051  106 FORMAT(' ANY INPUT VALUE FOR PEAKS',I3,' ?')
0052      GO TO 76
0053      75 WRITE(1,107)I
0054  107 FORMAT(' ANYMORE INPUT VALUE FOR PEAKS',I3,' ?')
0055      76 READ(1,102)YSNO
0056      IF(YSNO.NE.YS)GO TO 70
0057      WRITE(1,103)
0058      READ(1,*)J,PEAK(I,J)

```

```

0059      GO TO 75
0060      70 CONTINUE
0061      DO 80 I=1,8
0062      DO 85 J=1,25
0063      BUF(J)=CONT(I,J)
0064      85 CONTINUE
0065      CALL WRITF(INPT,IERR,BUF,50)
0066      WRITE(6,108)(BUF(K),K=1,25)
0067      108 FORMAT(5E12.5)
0068      80 CONTINUE
0069      DO 90 I=1,4
0070      DO 95 J=1,5
0071      BUF(J)=PEAK(I,J)
0072      95 CONTINUE
0073      CALL WRITF(INPT,IERR,BUF,10)
0074      WRITE(6,108)(BUF(K),K=1,5)
0075      90 CONTINUE
0076      CALL LOCF(INPT,IERR,IREF,IRB,IOFF,JSEC)
0077      ITR=JSEC/2-IRB-1
0078      CALL CLOSE(INPT,IERR,ITR)
0079      STOP
0080      END

```

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

4.6.3 MKFL3 LISTING

LNKFL3 T=00004 IS ON CR00026 USING 00005 BLKS R=0000

```
0001 FTH4,L
0002 PROGRAM MKFL3
0003 C PROGRAM FOR MAKING INPUT FILE INPT03
0004 C THIS PROGRAM WILL ALLOW TO TYPE IN THE ENERGY, SLOPE, AND
0005 C EZERO VALUES. IT WILL CREATE THE BINARY INPUT FILE.
0006 DIMENSION EDT(3),NOUT(144),NT(3)
0007 DATA NT/2HIN,2HPT,2H03/
0008 CALL CREAT(NOUT,IERR,NT,20,3,0,26,128)
0009 IF(IERR.LT.0)GO TO 701
0010 WRITE(1,101)
0011 101 FORMAT(' ENTER NUMBER OF PHOTON ENERGIES FOR'/
0012 A ' COMPUTING RESPONSE')
0013 READ(1,*)INP
0014 WRITE(1,102)
0015 102 FORMAT(' ENTER..ENERGY, SLOPE, AND EZERO FOR EACH RESPONSE')
0016 WRITE(1,103)
0017 103 FORMAT(' ENTER ONCE FOR EACH PHOTON ENERGY')
0018 DO 10 I=1,INP
0019 READ(1,*)XEDT(J),J=1,3)
0020 CALL WRITF(NOUT,IERR,EDT,6)
0021 10 CONTINUE
0022 EDT(1)=0.0
0023 CALL WRITF(NOUT,IERR,EDT,6)
0024 CALL LOCF(NOUT,IERR,IREC,IRB,IOFF,JSEC)
0025 IT=JSEC/2-IRB-1
0026 CALL CLOSE(NOUT,IERR,IT)
0027 700 STOP
0028 701 WRITE(1,711)IERR
0029 711 FORMAT(' CREATE ERROR=',I4)
0030 GO TO 700
0031 END
```