A Description of NUEXS, 
an Upgrade of the Code FCUP Used to Compute 
Proton Recoil Current from CH₂ Foils

Marion L. Stelts
Bernard E. Wood
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, make 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.
A DESCRIPTION OF NUEXS, AN UPGRADE OF THE CODE FCUP USED TO COMPUTE PROTON RECOIL CURRENT FROM CH₂ FOILS

by

Marion L. Stelts and Bernard E. Wood

ABSTRACT

A computer code, FCUP, developed by A. Craft computes currents of recoil protons from a time- and energy-dependent neutron flux striking a CH₂ foil. Three problem areas need to be addressed to extend the code's usefulness. First, FCUP computes a response that is not time dependent; that is, only the input time bin is broadened to account for the finite time distribution of protons from a single neutron energy; second, the time coordinate of the signal predicted is translated arbitrarily rather than absolutely relative to the time of maximum neutron production in the source; and third, the code does not account for electron pickup by protons at low proton energies in the target and absorber foils. This report describes the changes in calculational method used to overcome these problems.

INTRODUCTION

We first briefly describe the method used to compute the response. Reference 1 should be read for a more detailed description. A neutron beam is assumed to uniformly illuminate a CH₂ target of radius RT. A detector with a circular defining aperture of radius RD at a distance ZTAO and angle θ collects the current of recoil protons. The detector may be a simple Faraday cup (spectrum detector) viewing the target, it may have an aluminum filter to reduce response to low-energy protons (thresholded detector), or it may consist of thin aluminum foil preceded by an aluminum filter (ΔE detector) so that it is sensitive only to protons from a finite range of neutron energies.

The problem of computing the time-dependent response (TDR) is assumed to be separable into components involving geometry (that is, proton flight paths, weighting factors, and angles relative to the central angle of scatter as a function of scattering angle) and those components involving the velocity distribution of protons emerging from different depths in the target foil. The TDR distributes neutrons from input time bins into output time bins of identical structure. We discuss these components in separate sections.
GEOMETRICAL PARAMETERS

Reference 1 describes the use of the routine GEOM. However, two changes have now been made. Previously we computed the geometrical quantities by superimposing evenly spaced points on both target and detector aperture and computing the parameters as a function of scattering angle averaged over all point pairs. We now use a Monte Carlo method to generate the point pairs. A random number generator, RAN, is used to select a radius

\[ r_{t,d} = R_{t,d} \sqrt{\text{RAN}} \]  \hspace{1cm} (1)

and an angle

\[ \phi_{t,d} = 2\pi \times \text{RAN} \]  \hspace{1cm} (2)

on both target and detector. This procedure gives a uniform distribution over both surfaces. These \((r,\phi)\) pairs are transformed to \((X,Y)\) pairs. Symmetry is invoked to place points at \((X_{t},Y_{t}), (-X_{t},Y_{t}), (X_{d},Y_{d}), \) and \((-X_{d},Y_{d})\) on the target and at \((X_{o},Y_{o})\) and \((-X_{o},Y_{o})\) on the detector aperture. Computational time is reduced without sacrificing statistical accuracy. The number of point pairs is chosen to be 1000 times the number of scattering angle bins. We choose a different starting point for the random number sequence each time the problem is run, so that statistical effects can be determined. This number of point pairs generally gives uncertainties of less than 1% in any individual point in the output. The results are in excellent agreement with previous results except where field-of-view (FOV) constraints are involved. The former method starts points at the edge of both target and detector aperture and forces the last point to be on the edge for each line across target or detector. This procedure gave an artificially large weighting to the edges; hence, we observed disagreement of up to 4.5% for the computed solid angle where FOV constraints apply. Without FOV constraints, agreement is generally within 0.01%.

A second, rather trivial change was made to the algorithm that sorts the computed quantities into bins as a function of scattering angle. Previously, the computed angle of scatter was compared with each angular bin boundary in ascending order for each bin boundary until a match was found. Currently, we use a successive approximation method, starting at the middle bin, which converges in fewer steps to the proper match.

Both versions of the code contain a parameter DTAU, which is proportional to the bin size in the scattering angle mesh. For DTAU = 0.3 ns, the old and new versions require similar computational time. For DTAU = 0.15 ns (which implies twice the number of angular bins) the new version is several times faster than the old.

RESPONSE FUNCTION: SINGLE SCATTERING ANGLE

For a single scattering angle and neutron energy, the TDR is computed by the subroutine ARESPT as follows. First, ARESPT computes the effective target thickness, which is either the entire CH\(_{2}\) target or that portion producing recoil protons that can be detected. This portion is determined by the relation

\[ E_{p} = E_{n} \cos^{2} \theta \]  \hspace{1cm} (3)
the range-energy relationship for protons in CH$_2$ and aluminum, and the average proton angle relative to the normals to the CH$_2$ target and aluminum filter (and $\Delta E$ foil if appropriate). The effective target is then divided into NTARG slices. NTARG is chosen either to give sufficient time resolution per response target slice (based on the minimum time resolution required, $\Delta$TAU) for the target half nearest the detector or to have a minimum of 4 or maximum of 40 steps. This algorithm appears to give an adequate representation of the response. The time scale is determined by computing the proton time of flight relative to the detector from each bin boundary. The velocities are determined from Eq. (3), from the appropriate range-energy relationships for the materials in each portion of the proton flight path, and from the average flight path segments according to parameters generated by the subroutine GEOM.

The amount of response (Coulombs/neutron) in each time bin for neutron energy $E$ and scattering angle $\theta_s$ is

$$\Delta R(E,X,\theta_s) = \frac{zeN_A}{1000A} \frac{d\sigma}{dE} \Delta \Omega W(J) Q(E_p) \Delta X(E,\theta_s),$$

where

- $z$ = number of protons/molecule of target ($z = 2$ for CH$_2$),
- $e$ = electronic charge (Coulombs),
- $N_A$ = Avogadro's number,
- $1000$ = number of mg/g,
- $A$ = molecular wt ($A = 14$ for CH$_2$),
- $\Delta X(E,\theta_s)$ = thickness of each target slice in mg/cm$^2$ for neutrons,
- $W(\theta_s)$ = geometrical weight, computed by GEOM,
- $\Delta \Omega$ = solid angle of detector, computed by GEOM,
- $d\sigma/d\Omega (E,\theta_s)$ = $(n,p)$ differential scattering cross section (cm$^2$/sr), and
- $Q(E_p)$ = effective charge of protons being detected (a function of the proton energy $E_p$, with which protons leave the last material they pass through before being detected).

The $(n,p)$ scattering cross sections are calculated from the parameterization of Gammel$^2$ as described in Ref. 1. The effective charge of the proton in CH$_2$ and aluminum as a function of $E_p$ is derived from the evaluation of Janni$^3$ and the information on the negative charge state in Ref. 4. Figures 1 and 2 show $Q(E_p)$ for CH$_2$ and aluminum, respectively. These curves were fit for program use by a sixth-order polynomial in $\ln Q$ vs $\ln E_p$. Reference 3 contains the latest information on range-energy relationships for CH$_2$ and aluminum from 1-KeV to10-GeV proton energy. These data were fit with 10th-order polynomials in $\ln R$ vs $\ln E$ and $\ln E$ vs $\ln R$ in the energy ranges 0.001 to 0.100 MeV and 0.100 to 40.0 MeV. These functions agree with Janni's tabulations to within 0.1% over the entire range.
Fig. 1. Detected charge fraction for CH$_4$ vs proton energy.

Fig. 2. Detected charge fraction for aluminum vs proton energy.
The calculations of the spectrum and thresholded TDR are rather straightforward as described. The effective charge $Q(E_p)$ is computed by assuming a uniform energy distribution of protons within the bin. The dotted lines in Fig. 3(a) and (b) show the lines used for computing $\langle Q \rangle$ between the energies $E_1$ and $E_n$ of the protons at the bin boundaries.

The situation is somewhat more complicated for a $\Delta E$ detector. A proton that passes through that detector can still deposit an effective charge

$$Q_{\text{EFF}} = 1 - Z_{\text{EFF}}(E_p),$$

where $E_p$ is the energy of the proton as it leaves the $\Delta E$ foil. Hence, protons of up to 0.4-MeV energy leaving the $\Delta E$ foil will contribute to this signal. The dotted lines in Fig. 3 (c) and (d) show how $Q_{\text{EFF}}$ is computed in this case. If the protons do not penetrate the $\Delta E$ foil, then $\langle Q \rangle$ is computed as for a thresholded detector, using the energies with which the protons leave the filter.

We have chosen to represent the response as an integral rather than a differential. This response, $SNI(I)$, for the $ith$ time entry is

$$SNI(I) = \sum_{i=1}^{I} R_l,$$

For each index $I$, a time of proton detection $T(I)$ is computed. This procedure allows the response between any two time values to be determined by the difference between them that is obtained by interpolation in the arrays $SNI(I)$ and $T(I)$.

**RESPONSE FUNCTION: ANGLE INTEGRATED**

The total TDR for a given neutron energy is determined by summing the responses for all possible angles of scatter $\theta$. A problem arises because the proton energy and hence proton velocity are strong functions of scattering angle. Moreover, if the neutron energy is so low that the recoil protons cannot fully penetrate the target foil, the effective target thickness becomes a strong function of $\theta$. Therefore, we need a master set of time bins, encompassing the extreme range of possible proton times of flight, in which to accumulate the total response.

Because the highest energy protons (and hence the fastest) are produced at the minimum possible scattering angle that gives a response, the smallest time bin is determined at this angle. Similarly, the largest time of flight can be determined as the end of the last time bin at the largest angle to give a response. Note that since the proton has zero effective charge below 0.004 MeV, no proton detected will have a velocity smaller than that from 0.004 MeV.

The time scale is determined as follows. The routine ARESPT is called at the median angle for which there is a finite response. If ARESPT chose less than 20 time bins, the bins are subdivided so that their final number is between 32 and 40. A time $T_{\text{DMX}}$ is chosen to be the lesser of the average between $T_{\text{MIN}}$, the smallest possible response time, and $T_{\text{MAX}}$, the largest possible response time, or the time of flight of protons from the middle of the effective target at the largest scattering angle that gives a finite response. The median time entry in the output time bins is set to $T_{\text{DMX}}$. The time bins earlier than $T_{\text{DMX}}$ are scaled to cover the region from $T_{\text{MIN}}$ to $T_{\text{DMX}}$ and the later bins to span $T_{\text{DMX}}$ to $T_{\text{MAX}}$. 

5
Fig. 3. Schematic representation involved for computing the average effective charges in a bin bounded by proton energies $E_p$ and $E_u$. (a) and (b) refer to spectrum or thresholded detectors. (c) and (d) refer to penetration of protons through a $\Delta E$ detector.
We developed this algorithm by trial and error. There may be a better method, but this one gives a reasonable representation of all response functions thus far studied over a range of incident neutron energies (from 0.25 to 20 MeV) and detector types.

Once the master bin set has been determined, the routine ARESPT is called for each angle and the differential response is accumulated into these master time bins. After this is completed, the total cumulative response is computed. Figure 4 shows differential response functions for a thick spectrum detector for some neutron energies between 0.625 and 14.0 MeV. Figure 5 shows the cumulative response. The vertical bars on the 14.0- and 8.75-MeV responses indicate times beyond which there is no response.

FOLDING OF RESPONSE DATA AND NEUTRON FLUX

The main program reads in to memory the time- and energy-dependent flux computed by a Monte Carlo neutron transport code and then sets up the time bin array associated with this flux. It also tallies an energy array consisting of the midpoint of each energy bin. The object of the folding is to distribute the proton charge among output time bins as determined by the incident neutron flux and the TDR function.

One effect to be considered before proceeding is the distribution of the neutrons in the input time bin. The simplest assumption is uniform distribution. The mathematically correct method would subdivide each time bin into infinitesimally small intervals and apply the response function to each interval. Another approach would compute a new response function (a

![Graph showing differential TDR functions for a thick spectrum detector (127-μ CH₃ foil).](image)
convolution of the response function with a uniform distribution having the width of the input time bin) and then apply this new response to a delta function centered on the input time bin. Both of these methods were considered too time consuming. Instead we chose the following approximation. The original cumulative response function is “stretched” by $\Delta T = 0.5$ times the width of the input bin, in each direction, about the time of 50% of the cumulative response, as Fig. 6 shows. This folded response is then applied assuming that all neutrons in the input bin are located at the center. Although this method is not mathematically rigorous, it gives exact answers in the limiting cases of very small and very large input bins and is quite reasonable for intermediate widths.

The procedure for computing a signal is as follows. The parameters describing the neutron line of sight and the neutron flux are read in. The detector parameters are then entered, and the routine GEOM is called to define the geometrical parameters. For each neutron energy, the routine RESPT is called to get the cumulative response function for that energy. Then, for each input time bin, the response function is folded with the bin width, and the neutrons at that energy and time are distributed as proton charge appropriately among the output time bins:

$$Q(t_i) = \sum_j \sum_i F(L(E_nT_i) \left[ \text{SENSI}(TU_k - TO_i) - \text{SENSI}(TL_k - TO_i) \right] .$$  \hspace{1cm} (7)
where $E_n T_i$ are the energy and time bins for the neutron flux $F_L$, $\text{SENS}_I$ is the folded cumulative response as a function of time after arrival of the neutron, $T_{U_k}$ and $T_{L_k}$ are the upper and lower time limits of the $k$th bin, and $T_0$ is the center of the $i$th input time bin. The time-dependent current is computed by dividing the charge in each output time bin by the width of the bin. In addition, the average energy of the neutrons contributing to the current in each bin is computed.

Finally, the time scale is shifted by the time of maximum neutron production in the source, and the quantities of interest as described in Ref. 1 are printed out. If the current for another detector on the same line of sight is desired, more detector parameters may be read in and the procedure iterated. A listing of the program is included as the Appendix.

CONCLUSIONS

The procedures for calculating the response function in NUEXS, although somewhat more complicated than those in FCUP, allow a more physically rigorous description of the processes involved, that is, charge pickup and TDR. In addition, an absolute time scale is preserved, which will have application where cross timing between signals from different lines of sight and detectors is involved. It is difficult to make a detailed comparison between the two methods since one involves a TDR function and the other deals only with time-integrated quantities. In spite of this, we will compare the FCUP response functions (solid line) with the time-integrated NUEXS response (dashed line) in Figs. 7-9 for spectrum, 4-MeV thresholded, and 11-MeV
Fig. 7. Comparison of time-integrated NUEXS response (dotted line) with the old FCUP method (solid line) for a thick spectrum detector.

Fig. 8. Comparison of time-integrated NUEXS response (dotted line) with the old FCUP method (solid line) for a 4-MeV thresholded detector.

Fig. 9. Comparison of time-integrated NUEXS response (dotted line) with the old FCUP method (solid line) for an 11-MeV thresholded detector.
thresholded detectors. The responses differ only at low neutron (and hence proton) energy, where electron pickup by the proton becomes important. Figure 10 compares ΔE response functions. Here the differences are considerably larger because most protons have low energies as they are collected by the detector foil and hence an enhanced probability of electron pickup. For spectrum detectors there are some minor shape differences, but integrals over large regions of the spectrum generally agree to within ≤5% if the FCUP spectra are shifted in time to match the NUEXS calculations.

Thus NUEXS provides a more rigorous physical description of the response as well as an absolute time base. The currents previously calculated by FCUP are verified to agree to ≤5% (integrated) and to be reasonable in shape agreement. The ΔE calculations are now in better agreement with experiment. This added confidence in calculation of the response function allows us to look more critically at neutron transport calculations.

Fig. 10. Comparison of time-integrated NUEXS response (dotted line) with the old FCUP method (solid line) for a ΔE detector.
REFERENCES


APPENDIX

FORTRAN LISTING OF THE NUEXS PROGRAM
Los Alamos Identification No. LP-1437

FORTRAN IV-PLUS V3.0-3
NUEXS.FTN, 4

Page 1

FORTRAN IMAGE ON DM1:CLASS AT [223.202]NUEXS.FTN

THIS PROGRAM NEEDS MODULES IN NUEXSUB.OBJ

THIS PROGRAM COMPUTES THE RESPONSE OF FARADAY CUPS TO
PROTONS RECOLLING FROM NEUTRONS INCIDENT ON CH2 FOILS.
ALUMINUM FILTERS BEFORE THE FARADAY CUPS ARE ALLOWED
ALUMINUM DELTA-E DETECTORS ARE PROVIDED FOR

PROGRAM DEVELOPED BY DR. ALBA CRAFT, EG&G SUNSET
TRANSFERRED TO LANL JUNE 1981.
MODIFIED JULY 1981 FOR PDP-11 (B. E. WOOD)
ADDED DELTA - E CALCULATION
21-OCT-81 CHANGED GEOM IN RESPONSE TO COMM. FROM A. CRAFT
THIS CORRECTS ERRORS IN ORIGINAL TRANSMITTED FROM AL.
20 NOV 81, ADDED COMMENT CARDS IN ANTICIPATION OF DISCUSSIONS
WITH AL CRAFT. CHANGED CALCULATION OF DER(EN) RE TOF AT THEOR
STATION FROM INSIDE DO LOOP COMPUTING SIGNAL TO AN ARRAY
CALCULATED WHEN THE ENERGIES ARE READ.
7-DEC-81 CORRECTED SUBROUTINE RESP TO COMPUTE CORRECT DELTA-E
RESPONSE, SUBJECT TO STATED PHYSICS ASSUMPTIONS.
5-JAN-82 CHANGED COMPUTATION OF TIME SCALE TO MAKE TOF OF
NEUTRON WITH AVERAGE ENERGY AT MAXIMUM SIGNAL MATCH TIME SCALE
6-JAN-81 CHANGED TO ALLOW READING MC VERSION 2A OR 2B FILES.
THE DIFFERENCES IN FORMAT BETWEEN 2A AND 2B FILES ARE TAKEN
INTO ACCOUNT BY PARAMETER NENCOL, THE COLUMN IN WHICH THE WORD
ENERGY BEGINS.
12-JAN-81 CHANGED TO DO EITHER RELATIVISTIC OR NONREL.
CALCULATION. IT IS INTENDED THAT THIS CODE USE THE SAME KINEMATICS
AS WERE DONE IN MONTE CARLO.
26-JAN-82 MAJOR CHANGE IN STRUCTURE OF FCUP TO ALLOW
COMPUTATION OF A TIME DEPENDENT RESPONSE FUNCTION FOR
EACH ENERGY AND ANGLE. THIS RESPONSE IS DISTRIBUTED
OVER OUTPUT BINS WHICH ARE IDENTICAL TO THE INPUT BIN
STRUCTURE. THIS CHANGE ACCOUNTS PROPERLY FOR THE TIME
DEPENDENCE OF THE PROTON TIME DISTRIBUTION. ACCOUNTS
FOR THE LOSS IN RESPONSE AT LOW PROTON ENERGIES AS THE
PROTON PICKS UP ELECTRONS, AND PRESERVES THE PROPER
TIME SCALE.
AT THIS TIME THE PARAMETERIZATION OF THE RANGE-ENERGY
RELATIONS FOR AL AND CH2 HAVE BEEN CHANGED TO REFLECT
THE RECENT CALCULATIONS OF JOSEPH F. JANNI.
9-FEB-82
CHANGED TO COMPUTE TIME DEPENDENT RESPONSE FUNCTION, SUMMED
MODIFIED GEOM ROUTINE TO USE MONTE CARLO METHOD OF COMPUTING QUANTITIES WHICH ARE FUNCTION OF ANGULAR MESH.

11-MAR-82
MODIFIED CODE TO USE FINER BINNING FOR TARGET AND A MORE ACCURATE ALGORITHM FOR COMPUTING EFFECTIVE CHARGE IN A GIVEN BIN.

12-APR-82
CHANGED I/O TO REMOVE DEPENDENCE ON TECO PREPROCESSING

0002 COMMON/A/CS(75),WT(75),CST(75),CSQ(75),AVR(75),NA
0003 COMMON/B/E(60),SENS(60),TNL,TMN,THK,NE,TOF(60),DTAU,ZFV,RFV
0004 COMMON/C/RT,RD,DS,ZTAO,THETA,TTHICK,FTTHICK,ZGD,THIX,THMN,OMEGA
0005 COMMON/D/DTHICK,DERUN
0006 COMMON/H/MCOM/CTMC
0007 COMMON/RESP0/NTMAX,TIMIN,TIMAX,TA(41),SENSI(41),DIFT(41)
0008 COMMON/RESP1/SIGN,TOFP,NTARGM,ZSS,TND
0009 COMMON/RESP2/TIMINP,TIMAXP,TA(41),DIFT(41)
0010 COMMON/RESP3/NTMAXP,NTARGM,ZSS,TIMEP
0011 VIRTUAL FL(150,60),TT(300),SIG(300),DT(150),GT(150),TIMX(150)
0012 VIRTUAL GI(150),EBAR(150),GE(60),GN(60)
0013 DIMENSION F2D(60),FD(2),FSUM(2),ETEXT(3),I2D(3,5)
0014 LOGICAL DERUN,PRTRSP
0015 BYTE FILNAM(14),TITLE(50),STIL(80),ANS
0016 INITIALIZE 1D-2D FACTORS TO 1.
0017 DATA F2D/60*1./
0018 DATA F2D/60*1./
0019 FORMAT(10I5)
0020 FORMAT(BF10.0)
0021 FORMAT(/,4X,'AVERAGE(E:\B)='),F6.3,(/9X,'AVERAGE(E:\B)='),F6.3)
0022 FORMAT(5X,'APERTURE RADIUS=',F7.3,' CM',1X,'TS2, 'RDF*',/5X,'TARGET TO APERTURE DISTANCE=',F7.3,' CM',
0023 5X,'APERTURE TO DETECTOR DISTANCE=',F8.3,' CM',/5X,'TARGET THICKNESS=',F8.3,' MICRONS',)
0024 FORMAT(5X,'FOV DISTANCE (FROM TARGET)=',F9.3,' CM',
0025 5X,'FOV RADIUS (DETECTOR)=',F8.3,' CM',/5X,'TARGET SOLID ANGLE=',1PE10.3,
0026 5X,'APPROXIMATE TIME RESOLUTION=',OPF6.3,' NSEC',/5X,'NUMBER OF M.C. MESH POINT PAIRS=',F9 0.,/5X,
0027 5X,'CENTRAL PROTON SCATTERING ANGLE=',OPF7.3,' DEG',/5X,'TARGET PARAMETERS',/5X,'TARGET PARAMETERS',/5X,'TARGET PARAMETERS'
FIND OUT WHICH MC VERSION, SET UP REQUIRED PARAMETERS

C

OPEN PRINT FILE ON LUN 4 TO AVOID HOGGING LINE PRINTER

C

GET FILE NAMES AND OPEN

C
ASSIGN 60 TO NRET
WRITE(5,1010)
1010 FORMAT(1X,'enter name of MONTE file: ','5i
READ(5,7) NCH, (FILNAM(I), I=1, MINO(NCH,13))
IF((NCH.LE.0) OR. (NCH.GT.13)) GO TO 60
FILNAM(NCH+1) = 0
OPEN(UNIT=2, NAME=FILNAM, TYPE='OLD', READONLY, ERR=2100)

... PRINT TIME-DEPENDENT RESPONSE FUNCTIONS?

WRITE(5,1011)
1011 FORMAT(1X,'do you wish time-dependent response printed? ','5)
READ(5,47) ANS
47 FORMAT(A1)
PRTRSP = ((ANS.EQ.1HY). OR. (ANS.EQ.1Hy))

... READ PARAMETER FILE

READ(182) ZSS, ZCOL, RCOL, ZFOV, RFOV, DTAU, GCOR, TCOR
IF(TCOR.GT.0.0) GO TO 65
WRITE(5,1015)
1015 FORMAT(1X,'enter TCOR = TMAX or 0.0 to do Ebar calculation: ','8)
READ(5,2) TCOR
65 IF(GCOR.LE.0.0) GCUR = 1.0
IF(DTAU.LE.0.0) DTAU = 0.3
GCOL = RCOL/ZCOL

SLOPE OF PENUMBRA

RCF = (RCOL+RFOV)/(ZCOL-ZFOV)

 GEOMETRICAL FACTOR OF DEFINING COLL.

GF = GCOL*GCOL/4.0
GFAC = GCOR*GF*1.0E+9
OMEG = 3.1415927*GCOL*GCOL

IF ITPE .NE. 0 READ 1D TO 2D FACTORS

IF ITPE .NE. 0 WRITE SIGNAL VS TIME TO OUTPUT FILE
NOTE ITPE .NE. 0 IMPLIES ITPE .NE. 0
READ(1,1) I2Di ITPE, IPLOT
IF(IPLOT .NE. 0) ITPE = -1

GET OUTPUT FILE IF REQUIRED

IF(ITPE.EQ.0) GO TO 80
WRITE(5,1020)
1020 FORMAT(1X,'enter name of OUTPUT file: ','8)
READ(5,7) NCH, (FILNAM(I),I=1,MINO(13,NCH))
IF((NCH. LE. 0).OR.(NCH. GT. 13)) GO TO 70
FILNAM(NCH+1)=0
C ... READ SOURCE TITLE, THEN POSITION TO GET ENERGY AND TIME BINS
READ(2,7) NSTITL, (STITL(I),I=1,MINO(80,NSTITL))
IF(NSTITL .GT. 80) NSTITL = 80
C ... GET FROM MC OUTPUT TAPE-
C ... NE = NUMBER OF ENERGY BINS
C ... EN = AVERAGE ENERGY OF EACH BIN (MEV)
C ... NT = NUMBER OF TIME BINS
C ... TIMEX(I) = UPPER LIMIT OF TIME BIN
C ... DT(I) = WIDTH OF TIME BIN
CALL FINDIT(2, 'TIME: ', 5, NTCOL)
CALL FINDIT(2, 'ENERGY', 6, NENCOL)
ELOW=0.0
CONST = ZSS/(EON*29.979)
IEN=0
85 IF(IEN,GE,60) STOP 'TOO MANY ENERGIES'
READ(2,1022) ETEXT
1022 FORMAT(<NENCOL)x,3A4)
102 IF(NMCC.EQ.'2A').AND.(ETEXT(2).EQ.'TOT') GO TO 90
103 IF(NMCC.EQ. '28').AND.(ETEXT(2).EQ. 'TAL')) GO TO 90
104 DECODE(10,1025,ETEXT) EHIGH
105 1025 FORMAT(F10.0)
106 IEN=IEN+1
C ... COMPUTE MEAN ENERGY IN EACH BIN
E(IEN) = 0.5*(ELOW + EHIGH)
108 ELOW=EHIGH
109 GO TO 85
110 NE=IEN
C
REWIND 2
IT=0
113 TIMEX(1)=0.0
114 100 IF(IT.GE.150) STOP 'TOO MANY TIMES'
115 CALL FINDIT(2, 'TIME: ', 5, NTCOL)
116 BACKSPACE 2
117 READ(2,1027) TTEXT
1027 FORMAT(<NENCOL+14)x,5(3A4,9X))
119 DO 105 I=1,5
120 IF(TTEXT(1,I).EQ. 'TO')) GO TO 110
121 IT=IT+1
122 DECODE(10,1025,TTEXT(1,I)) TTEMP
123 TIMEX(IT+1) = 10.0*TTEMP
124 105 DT(IT) = TIMEX(IT+1)-TIMEX(IT)
125 GO TO 100
126 110 NT=IT
C ... READ FROM MC OUTPUT TAPE FL(I,J)
C ... FL(I,J) = ATTENUATED NUMBER OF NEUTRONS INTO 4PI IN ITH TIME BIN
C ... AND JTH ENERGY BIN.
REWIND 2

NFULL = NT/5

IF(NFULL .EQ. 0) GO TO 120

DO 115 IPAGE=1,NFULL

CALL FINDIT(2, 'TIME', 5, NTCOL)

CALL FINDIT(2, 'ENERGY', 6, NENCOL)

ITBASE = 1+5*(IPAGE-1)

DO 115 IEN=1,NE

READ(2,1033) (FL(IT, IEN), IT=ITBASE, ITBASE+4)

CONTINUE

115

NTLEFT = NT-5*NFULL

IF(NTLEFT .EQ. 0) GO TO 150

CALL FINDIT(2, 'ENERGY', 6, NENCOL)

ITBASE = 1+5*NFULL

DO 150 IEN=1,NE

READ(2,1033) (FL(IT, IEN), IT=ITBASE, ITBASE+NTLEFT-1)

CONTINUE

1033

FORMAT(<NENCOL>X,5(10X,F11.0))

POSITION MC FILE TO GET UNATTENUATED TOTAL COUNTS

CALL FINDIT(2, 'RAD', 3, 2)

IQTPOS = 22+21*NTLEFT + (NENCOL-5)

CALL FINDIT(2, 'TOTAL', 5, IQTPOS)

READ FROM MC OUTPUT TAPE QE(I)

QE(I) = UNATTENUATED NUMBER OF NEUTRONS INTO 4PI IN ITH ENERGY BIN

TOTALLED FOR ALL TIMES.

CALL FINDIT(2, 'ENERGY', 6, NENCOL)

NSKIP = IQTPOS-7

DO 180 IEN=1,NE

READ(2, 1035) QE(IEN)

CONTINUE

1035

FORMAT(<NSKIP>X,F11.0)

CLOSE(UNIT=2)

CALL FINDIT(2, 'ENERGY', 6, NENCOL)

FD(1) = 0.0

FD(2) = 0.0

IF(I2D .EQ. 0) GO TO 290

READ IN ENERGY DEPENDENT 1D TO 2D CORRECTION FACTORS

READ(1,2) (F2D(I), I=1,NE)

COMPUTE AVERAGE 1D TO 2D CORR FACTORS FOR FISSION, FUSION.

J = 1

FD(1) = QE(1)*F2D(1)

FD(2) = 0.0

FSUM(1) = QE(1)

FSUM(2) = 0.0
DO 220 K=1,NE
IF(F2D(K) .LE. 0.0) F2D(K) = F2D(K-1)
IF(E(K) .GT. 8.0) J = 2
FSUM(J) = FSUM(J)+GE(K)
FD(J) = FD(J)+F2D(K)*GE(K)
220 CONTINUE
FD(1) = FD(1)/FSUM(1)
FD(2) = FD(2)/FSUM(2)

C ... ADJUST FL(T,E) BY ENERGY DEPENDENT 1D TO 2D F2D(E)
C
DO 230 I=1,NT
DO 230 J=1,NE
FL(I,J) = FL(I,J)*F2D(J)
230 CONTINUE
C ... PUT DATE INTO FCUP RUN TITLE
290 CALL DATE(TITLE(32))
C ... PUT TIME INTO FCUP RUN TITLE AND READ IT
300 TITLE(41)="40
301 TITLE(42)="40
302 TITLE(43)="40
303 READ(1,7,END=999) NTITLE, (TITLE(I), I=1, MINO(30, NTITLE))
C ... FILL UNUSED TITLE SPACE WITH BLANKS
304 DO 305 I=NTITLE+1,31
305 TITLE(I)="40
C ...
ZS = EXPERIMENTAL TARGET STATION (CM)
ZTAO = TARGET TO APERTURE (CM)
ZAD = APERTURE TO DETECTOR (CM)
RD = APERTURE RADIUS (CM)
ZFV = DISTANCE FROM TARGET OF FOV APERTURE (CM)
RFV = RADIUS (CM) OF FOV APERTURE
THETA = DETECTOR ANGLE (DEGREES) (DROPOUT = 30.)
C
READ(1,2) ZS, ZTAO, ZAD, RD, ZFV, RFV, THETA
C ...
TTHICK = TARGET THICKNESS (MG/CM**2)
FTHICK = FILTER THICKNESS (MG/CM**2) (DROPOUT = 0.)
DTHICK = DELT E THICKNESS (MG/CM**2) (DROPOUT = 0.)
C
READ(1,2) TTHICK, FTHICK, DTHICK
IF(RD .LE. 0.0) RD = 1.27
IF(THETA .LE. 0.0) THETA = 30.0
IF(FTHICK .LT. 0.0) FTHICK = 0.0
IF(DTHICK .LT. 0.0) DTHICK = 0.0
C ...
COMPUTE THICKNESSES IN MICRONS. RHO(CH2) = 0.92, (AL) = 2.70
C
TTHIKM = TTHICK/0.092
FTHIKM = FTHICK/0.27
DTHIKM = DTHICK/0.27
C ...
DERUN = (DTHICK, NE. 0.)
C ...
COMPUTE RADIUS OF PENUMBRA AT TARGET
RT = (ZS-ZFOV)*RCF-RFOV
C ... INITIALIZE OUTPUT ARRAYS

C
C
C
C
C

0194 GDOT = 0.0
0195 DO 310 I=1,300
0196 310 SIG(I) = 0.0
0197 DO 311 I=1,150
0198 GT(I) = 0.0
0199 EBAR(I) = 0.0
0200 311 GI(I) = 0.0
0201 DO 312 I=1,60
0202 312 QN(I) = 0.0
0203 SENS(I) = 0.
0204 TOF(I) = 0.
0205 QE(I) = 0.0

C ... SET UP ANGULAR ARRAY AND PERTINENT PARAMETERS FOR RESPT
C

0206 CALL GEOM
C

0207 DO 320 J=1,NE
C

C ... PARAMETERIZE H(N,N)H CROSS SECTION
C ... SIGH = (TOTAL INTEGRATED XSECT IN BARNS)/PI
C ... DIFF XSECT = SIGH*COS(THETA)
C

0208 EN = E(J)
0209 FA = (0.09415+0.0001306*EN)*EN-1.86
0210 FB = 0.42230.13*EN
0211 FC = 1.206*EN
0212 SIGH = (3.0/(FC+FA*FA)+(1.0/(FC+FB*FB)))

C ... REFERENCE: J. L. GAMMEL IN 'FAST NEUTRON PHYSICS', PART II, P. 2185
C ... ED. BY MARION AND FOWLER.

C

0213 CALL RESPT(EN, IPOS)
C

C ... CHECK FOR ZERO RESPONSE
0214 IF(IPOS.EQ.0) GO TO 320
0215 SENS(J) = SENS(I+1)
0216 TOF(J) = TOFP

C ... PRINT TIME-DEPENDENT RESPONSE FUNCTIONS IF NECESSARY
C

0217 IF(.NOT. PRTRSP) GO TO 313
0218 WRITE(4,7000) EN,SENS(J),NTMAX
0219 7000 FORMAT(/,' RESPONSE FOR E = ',1PE12.3,' SENS = ',1PE12.3,
1    2X,13,' POINTS',/,' TIME INTERVAL',3X,
2    'SENS(COL/NSEC/NEUT)',' INTEGRAL TO T2',/
0220 DO 7313 I=1,NTMAX-1
0221 TI = TI(I)
0222 T2 = TI(I+1)
0223 XX = (SENS(I+1) - SENS(I))/(T2 - TI)
WRITE(4,7001) TI, T2, XX, SENSI(I+1)

7001 FORMAT(1X,1PE12.3, ' TO',3E12.3)

C ...
C ... FOLD RESPONSE FROM ALL INPUT TIME BINS ...........
C ...
C ... NOTE: POSSIBLE TROUBLE IF ZSS .GT. ZS
C ...
C ... TL = LOWER TIME LIMIT FOR INPUT TIME BIN
C ...
C ... TU = UPPER TIME LIMIT FOR INPUT TIME BIN

313 TU = 0.
DO 316 KK=1, NT
TL = TU
TU = TIMEX(KK+1)
XNEUT = GFAC*FL(KK, J)
IF(XNEUT .EQ. 0.) GO TO 316
TO = 0.5*(TU + TL)
C ...
C .. BROADEN TIME BASE OF INTEGRAL RESPONSE FUNCTION TO ACCOUNT
C ...
C ... FOR THE WIDTH OF THE INPUT TIME BIN. BROADENING IS RELATIVE
C ...
C ... TO THE CENTER OF THE TIME BIN AND MAXIMUM OF RESPONSE.
C

DELTP = TO - TL
TIMINP = TIMIN - DELTP
TIMAXP = TIMAX + DELTP
C ...
C ... FIND INDEX OF AVERAGE TIME OF RESPONSE
C
TEST = 0.5*SENS(J)
DO 3000 IM=1, NTMAX
IF(SENSI(IM) .GE. TEST) GO TO 3001
CONTINUE
3001 IL = IM - 1
IU = NTMAX - IM
C ...
C ... ADJUST LOWER HALF TIME SCALE
DTTL = DELTP/IL
DTT = 0.
DO 3002 I=IL,1,-1
DTT = DTT + DTTL
TIP(I) = TI(I) - DTT
CONTINUE
3002
C ...
C ... ADJUST UPPER HALF TIME SCALE
DTTU = DELTP/IU
DTT = -DTTU
DO 3003 I=IM,NTMAX
DTT = DTT +DTTU
TIP(I) = TI(I) + DTT
CONTINUE
3003
C ...
C ... INTO ALL OUTPUT TIME BINS.
C ...
C ... TM = LOWER LIMIT OF OUTPUT TIME BIN RE TO.
C ...
C ... TP = UPPER LIMIT OF OUTPUT TIME BIN RE TO.
C
TP = -TO
RINTP = PRESP(TP)
DO 314 KKK=1,NT
TM = TP
RINTM = RINTP
TP = TIMEX(KKK+1) - TO
C ... IS TP BEFORE FINITE RESPONSE TIME?
IF(TP .LE. TIMINP) GO TO 314
C ... IS TM AFTER FINITE RESPONSE TIME?
IF(TM .GE. TIMAXP) GO TO 316
C . COMPUTE CONTRIBUTION IN KKK TH TIME BIN
RINTP = PRESP(TP)
CONT = XNEUT*(RINTP - RINTM)
QT(1) = GT(KKK) = GT(KKK) + CONT
C ... ACCUMULATE CHARGE IN KKK
C ... ACCUMULATE SIGNAL WEIGHTED AVERAGE NEUTRON ENERGY IN KKK
EBAR(KKK) = EBAR(KKK) + CONT*EN
314 CONTINUE
C ... END OF LOOP ON OUTPUT TIME BINS
316 CONTINUE
C ... END OF LOOP ON INPUT TIME BINS
320 CONTINUE
C . END OF LOOP ON NEUTRON ENERGY
C . NORMALIZE EBAR, COMPUTE TIME INTEGRAL OF CHARGE, COMPUTE CURRENT
K = 0
DO 322 I=1,NT
IF(EBAR(I) .LE. 0.0) EBAR(I) = -1.0
IF(EBAR(I) .GT. 0.0) EBAR(I) = EBAR(I)/GT(I)
QTOT = QTOT + QT(I)
QI(I) = QTOT
K = K+2
C ... COMPUTE CURRENT
SIG(K-1) = GT(I)/DT(I)
SIG(K) = SIG(K-1)
322 CONTINUE
C . ACCUMULATE CHARGE AND NUMBER OF NEUTRONS IN EACH ENERGY BIN
DO 326 J=1,NE
DO 324 I=1,NT
XNEUT = FL(I, J)
C ... CHARGE IN EACH ENERGY BIN
GE(J) = GE(J) + SENS(J)*XNEUT*GFAC
C ... NUMBER OF NEUTRONS IN EACH ENERGY BIN
GN(J) = GN(J) + XNEUT
324 CONTINUE
326 CONTINUE
C . COMPUTE TOTAL TOF FOR EACH ENERGY BIN
C . TOF = TIME-OF-FLIGHT TO ZSS + <T-O-F FROM ZSS TO DETECTOR>
DO 330 I=1,NE
TERM = 0.
IF(SENS(I) .GT. 0.) TERM = TOF(I)/SENS(I)
TOF(I) = TOFF(E(I),ZSS,EON) + TERM

330 CONTINUE

DEFINE INTEGRATION LIMITS—NOMINALLY 1—, 8—, 14—MEV.

ENL = 1.
IF(ENL .LT. E(1)) ENL = E(1)
ENM = 8.
IF(ENM .LT. E(1)) ENM = E(1)
ENH = 14.

FIND TIMES-OF-FLIGHT FOR THESE ENERGIES

CALL TINTCENLi, TNLi, TOFi, NE)
CALL TINT(ENL, TNM, E, TOF, NE)
CALL TINTCENH, TNH, E, TOFi, NE)

DT1 = TNL - TNM
DT2 = TNM - TNH

IF TCOR = 0, COMPUTE TORP AS TOF(EBAR(MAX SIGNAL))
TCOR = TCOR
IF(TCOR .NE. 0.) GO TO 356
WRITE(4F1400)
1400 FORMAT(//1X, '**** NOTE: Time shift was calc’ed via Ebar, /)

FIND TIME BIN WITH MAXIMUM SIGNAL
IMX = 1
K = 0
TEST = 0.
DO 351 I=1,NT
K = K+2
IF(ISIG(K) .LT. TEST) GO TO 351
TEST = SIG(K)
IMX = I
351 CONTINUE

FIND AVERAGE TOF FOR EBAR(IMX)
EBARP = EBAR(IMX)
DO 352 I=1,NE-1
IF(EBARP .LE. E(I)) GO TO 353
352 CONTINUE
WRITE(5,1500)
1500 FORMAT(//1X, '** WARNING — FELL THROUGH FINDING EBAR BIN')

SLP = (TOF(I)-TOF(I-1))/(E(I)-E(I-1))
TCORP = TOF(I-1) + SLP*(EBARP-E(I-1))

ADJUST TIME SCALE SUCH THAT MAXIMUM SIGNAL HAS TOF OF
AVERAGE NEUTRON ENERGY IN THAT TIME BIN
TCORP = TIMEX(IMX) - 0.5*DT(IMX) - TCORP

SHIFT TO PROPER TIME-OF-FLIGHT SCALE

TT(1) = -TCORP
NTS = 2*NT
J = 0
DO 360 I=2,NTS,2
J = J+1
TT(I) = TT(I-1) + DT(J)
TT(I+1) = TT(I)
360 CONTINUE
IF(TT(NTS).LT.100000.) GO TO 361
TT(NTS) = 99999.99
TT(NTS+1) = 99999.99
361 CONTINUE

IF NOT A DELTA-E, COMPUTE FWHM OF 14 MEV PEAK

FWHM = 0.0
IF(DERUN) GO TO 365
JMX = 2
FM = 0.
DO 362 I=2,NTS,2
IF(TT(I).GE. TNM) GO TO 363
IF(SIG(I).LT. FM) GO TO 362
FM = SIG(I)
362 CONTINUE
JMX = 1
FM = SIG(I)
363 CONTINUE
FHM = FM/2.0
DO 20 I=1,JMX,2
IF(SIG(I).GT. FHM) GO TO 21
20 CONTINUE
TA = (TT(I)+TT(I-2))/2.0
TB = (TT(I)+TT(I+2))/2.0
TFA = TA+(TB-TA)*(FHM-SIG(I-2))/(SIG(I)-SIG(I-2))
DO 22 I=JMX,NTS,2
IF(SIG(I).GE. FHM) GO TO 22
IF(SIG(I+2).LT. FHM) GO TO 23
22 CONTINUE
J = I-1
TA = 0.5*(TT(J)+TT(J-2))
TB = 0.5*(TT(J)+TT(J+2))
FHW = (TA+(TB-TA)*(SIG(I-2)-FHM)/(SIG(I-2)-SIG(I)))-TFA

... COMPUTE CHARGE INTEGRALS E = (0,8) AND (8,Infinity)

NOTE TERP1 ARGUMENTS TT AND GI ARE VIRTUAL ARRAYS

GSUM1 = TERP1(TNM,NT,TT,GI)
GSUM2 = TERP1(TNL,NT,TT,GI)-GSUM1
ZTDO = ZTAD+ZAD
WRITE(4,33) TITLE
WRITE(4,33)
FORMAT( ' VERSION NUEXS- 11-MAR-82, TIME-DEPENDENT RESPONSE. ')
C TTHICK = CH2 TARGET THICKNESS (MG/CM**2)
C TTHIKM = CH2 TARGET THICKNESS (MICRONS)
C RT = TARGET RADIUS (CM)
C ZS = EXPERIMENTAL STATION (CM)
C ZSS = THEORETICAL STATION (CM)
C ZCOL = DEFINING COLLIMATOR STATION (CM)
C RCOL = RADIUS OF DEFINING COLLIMATOR (CM)
C OMEG = SOURCE TO TARGET SOLID ANGLE.
C GF = SOURCE GEOMETRICAL FACTOR.
C ZFOV = LOCATION OF FOV COLLIMATOR RE WP (CM)
C RFOV = RADIUS OF FOV COLLIMATOR
C
WRITE(4,8) TTHICK, TTHIKM, RT, ZS, ZSS
WRITE(4,9) ZCOL, RCOL, OMEG, GF, ZFOV, RFOV
C
FTHICK = AL FILTER THICKNESS (MG/CM**2)
FTHIKM = AL FILTER THICKNESS (MICRONS)
RD = DETECTOR APERTURE RADIUS (CM)
ZTAO = TARGET TO APERTURE DISTANCE (CM)
ZTDO = TARGET TO DETECTOR DISTANCE (CM)
ZAD = APERTURE TO DETECTOR (CM)
ZFV = TARGET TO TARGET FOV APERTURE (CM)
RFV = RADIUS OF TARGET FOV APERTURE (CM)
OMEGA = TARGET TO DETECTOR SOLID ANGLE
CTMC = NUMBER OF M.C. MESH POINT PAIRS USED IN GEOM.
DTAU = APPROX. TIME RESPONSE OF DETECTOR (NS)
THETA = CENTRAL PROTON SCATTERING ANGLE
C
WRITE(4,4) FTHICK, FTHIKM, DTHICK, DTHIKM
WRITE(4,5) RD, ZTAO, ZTDO, ZAD
WRITE(4,6) ZFV, RFV, OMEGA, DTAU, CTMC, THETA
C
THMX = MAXIMUM SCATTERING ANGLE
THMN = MINIMUM SCATTERING ANGLE
NA = NUMBER OF ANGLES
TCOR = GENERAL CORRECTION FACTOR
TCOR = TIME SHIFT (NS)
GSUM1 = INTEGRAL CHARGE FOR FUSION
GSUM2 = INTEGRAL CHARGE FOR FISSION
FWHM = WIDTH OF 14-MEV PEAK IN NSEC.
C
WRITE(4,11) THMX, THMN, NA, GCOR, TCOR, ENM,
1 GSUM1, GSUM2, FWHM
C
ENL = ENERGY OF LOWER ENERGY BOUND FOR INTEGRALS
TNL = TIME OF FLIGHT FOR LOWEST ENERGY.
ENM = ENERGY OF UPPER BOUND FOR FISSION INTEGRAL.
(TLOWER BOUND FOR FUSION INTEGRAL)
TNM = TIME OF FLIGHT FOR THAT ENERGY.
TNH = TIME OF FLIGHT FOR 14 MEV NEUTRON.
DT1 = TNL - TNM
DT2 = TNM - TNH
FORTRAN IV-PLUS V3.0-3  16:22:23  20-May-82  NUEXS.FTN: 4

0379 WRITE(4,16) ENL, TNL, ENM, TNM, TNH, ENL, DT1, ENM, DT2
0380 WRITE(4,10) TITLE
0381 WRITE(4,26) (STITL(I), I=1, NSTITL)
0382 WRITE(4,12)
0383 N = NT
0384 IF(NT.GT.50) N = 50
0385 J = 1
C ... TT = TIME BIN BOUNDARY (NSEC)
C ... SIG = SIGNAL IN TIME BIN (AMPS)
C ... QT = CHARGE IN TIME BIN
C ... GI = INTEGRATED CHARGE FROM TIME ZERO THRU BIN.
C
0386 DO 370 I=1, N
0387 WRITE(4,13) I, TT(J), TT(J+1), SIG(J), QT(I), GI(I), EBAR(I)
0388 J = J+2
0389 CONTINUE
0390 IF(N.EQ.NT) GO TO 390
0391 WRITE(4,10) TITLE
0392 WRITE(4,26) (STITL(I), I=1, NSTITL)
0393 WRITE(4,12)
0394 DO 375 I=1, NT
0395 WRITE(4,13) I, TT(J), TT(J+1), SIG(J), QT(I), GI(I), EBAR(I)
0396 J = J+2
0397 CONTINUE
0398 CONTINUE

C ... WRITE SIGNAL TO OUTPUT FILE
C
0399 IF(ITPE.EQ.0) GO TO 390
0400 WRITE(3,25) TITLE, (STITL(I), I=1, NSTITL), FD(1), FD(2)
0401 WRITE(3,11) NTS
0402 WRITE(3,14) (TT(I), SIG(I), I=1, NTS)
0403 WRITE(3,11) NE
0404 WRITE(3,14) (E(I), SENS(I), I=1, NE)
0405 CONTINUE

C ... PRINT ENERGY DEPENDENT PARAMETERS
C
0406 WRITE(4,10) TITLE
0407 WRITE(4,26) (STITL(I), I=1, NSTITL)
0408 WRITE(4,15)
0409 IF(. NOT. DERUN) WRITE(4,17)
0410 IF(DERUN) WRITE(4,18)

C ... E = AVERAGE NEUTRON ENERGY IN BIN (MEV)
C ... TOF = TIME-OF-FLIGHT (NSEC)
C ... GN = TOTAL NUMBER OF NEUTRONS IN ENERGY BIN INTO 4PI
C ... GE = CHARGE DETECTED DUE TO NEUTRON IN ENERGY BIN
C ... SENS = DETECTOR SENSITIVITY AT ENERGY, E (COUL/NEUT)
C ... F2D = ENERGY DEPENDENT 1D TO 2D CORRECTION FACTOR.
C
0411 DO 400 I=1, NE
0412 WRITE(4,19) E(I), TOF(I), GN(I), GE(I), SENS(I), F2D(I)
0413 CONTINUE
C ... FD(1) = AVERAGE 1D TO 2D CORR FACTOR FOR FISSION
C ... FD(2) = AVERAGE 1D TO 2D CORR FACTOR FOR FUSION
C
0414 IF(I2D.NE.0) WRITE(4,3) FD(1),FD(2)
0415 GO TO 300
C
C ... ALL DONE
0416 999 CLOSE(UNIT=1)
0417  CLOSE(UNIT=2)
0418  IF(ITPE.NE.0) CLOSE(UNIT=3)
0419  CLOSE(UNIT=4)
0420  CALL EXIT
C
C ... COME HERE ON FILE - OPEN ERROR
0421 2100 WRITE(5,2500) (FILNAM(I),I=1,NCH)
0422 2500 FORMAT(1X,'can't open file ',14A1)
0423  GO TO NRET
0424  END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>#CODE1</td>
<td>016136</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>#PDATA</td>
<td>005522</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>#IDATA</td>
<td>000160</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>#VARS</td>
<td>001414</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>#TEMP5</td>
<td>000024</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>A</td>
<td>0002736</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>B</td>
<td>0001352</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>C</td>
<td>0000054</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>D</td>
<td>000006</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>MCCOM</td>
<td>000004</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>RESP0</td>
<td>000766</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>RESP1</td>
<td>000022</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>RESP2</td>
<td>000520</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>RESP3</td>
<td>000766</td>
<td>RW, D, OVR, GBL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 034614  7366
Total Virtual Array Storage = 659
SUBROUTINE RESPT(EN, IPOS)

COMPUTES TIME DEPENDENT RESPONSE OF NEUTRONS OF ENERGY, EN.
IPOS = 0 IMPLIES ZERO RESPONSE.
ROUTINE GEOM MUST BE CALLED BEFORE CALLING RESPT TO SET UP
ANGULAR ARRAYS AND COMPUTE OMEGA.
CALLED BY - MAIN
CALLS - ARESPT DELRES
8-FEB-82

NTMAX = NUMBER OF ENTRIES IN CUMULATIVE RESPONSE.
TIMIN = MINIMUM TIME OF RESPONSE.
TIMAX = MAXIMUM TIME OF RESPONSE.
TI = TIME ENTRIES RELATIVE TO NEUTRONS AT ZSS.
SENSI = CUMULATIVE RESPONSE AT TI
DIFT = TI(I) - TI(I-1)

/RESPO/
TIMAX = NUMBER OF ENTRIES IN CUMULATIVE RESPONSE.
TIMIN = MINIMUM TIME OF RESPONSE.
TIMAX = MAXIMUM TIME OF RESPONSE.
TI = TIME ENTRIES RELATIVE TO NEUTRONS AT ZSS.
SENSI = CUMULATIVE RESPONSE AT TI
DIFT = TI(I) - TI(I-1)

/RESP1/ SEE SUBROUTINE ARESPT
NTARGM = MAXIMUM NUMBER OF DIVISIONS OF CH2 FOIL.

/RESP3/ SAME AS /RESPO/ EXCEPT INTERMEDIATE STORAGE
WHILE SUMMING OVER ANGLE.

COMMON/A/CS(75),WT(75),CST(75),CSQ(75),AVR(75),NA
COMMON/B/E(60),SENS(60),TNL,TNM,TNH,NE,TOF(60),DTAU,ZFV,RFV
COMMON/D/ DTHICK,DERUN
COMMON /RESPO/ NTMAX,TIMIN,TIMAX,TI(41),SENSI(41),DIFT(41)
COMMON /RESP1/ SIGH,TOFP,NTARGM,ZSS,TMID
COMMON /RESP3/ NTMX,TMINT,TMAXT,TIT(41),SENS(41),DIFTT(41)
DIMENSION TRP(41)
LOGICAL DERUN

SEE IF EN GREAT ENOUGH TO GIVE FINITE RESPONSE
IF(EN .GT. 0.006) GO TO 1
IPOS = 0
RETURN

FIND TIMIN
IF(DERUN) GO TO 10
FOR F.C. OR THRESH
CALL ARESPT(0, EN, 1, IPOS, 2)
NDMN = 1
IF(IPOS .EQ. 0) RETURN
GO TO 12
FOR DELTA-E
C
0018  10   DO 11 I=1,NA
0019   CALL ARESPT(0, EN, I, IPOS, 2)
0020   NDMIN = I
0021   IF(IPOS .NE. 0) GO TO 12
0022   11   CONTINUE
0023   RETURN
0024   12   TIMIN = TMINT
  C ... FIND TIMAX, TMDMX
0025   IF(DERUN) GO TO 14
C ... FOR F.C. OR THRESH
0026   K = NA
0027   13   CALL ARESPT(0, EN, K, IPOS, 2)
0028   NDMX = K
0029   IF(IPOS .NE. 0) GO TO 16
0030   K = K-1
0031   GO TO 13
C ... FOR DELTA-E
C
0032   14   DO 15 I=NA,1,-1
0033   CALL ARESPT(0, EN, I, IPOS, 2)
0034   NDMIN = I
0035   IF(IPOS .NE. 0) GO TO 16
0036   15   CONTINUE
0037   RETURN
0038   16   TIMAX = TMAXT
0039   TMDMX = TMID
  C ... SET MID TIME FOR DIVISION OF TIME SCALE TO LESSER OF
  C ... (MIDDLE OF TOTAL TIME SPAN) OR (TOF OF PROTON FROM MIDDLE
  C ... OF TARGET AT LARGEST ANGLE PARTICIPATING).
0040   TEST = 0.5*(TIMIN+TIMAX)
0041   IF(TMDMX .GT. TEST) TMDMX = TEST
C ... DEFINE TIME SCALE. GET TIT FOR ANGLE NDMX/2 AND SCALE TO
C ... REGION (TIMIN,TIMAX).
C ... MAKE MASTER SET OF TIME BINS WITH APPROX 2X RESOLUTION
C ... OF BINNING FOR EACH ANGLE.
C
0042   NM = NDMIN/2
0043   IF(NM .LT. 1) NM = 1
0044   DTAUT = DTAU
0045   DTAU = 0.5*DTAU
0046   CALL ARESPT(0, EN, NM, IPOS, 2*NTARGM)
0047   DTAU = DTAUT
0048   IF(IPOS .EQ. 0) RETURN
0049   IF(NTMAX .GE. 2) GO TO 160
0050   IPOS = 0
0051   RETURN
C
c  .  IF NECESSARY SUBDIVIDE TIME BINS TO MAINTAIN NTMAX BETWEEN C  .  33 AND 41.

C

0052  160  NTMAX = NTMX
0053  161  NPR = NTMX - 1
0054  162  NFC = (2*NTARGM)/NPR
0055  163  IF(NFC .EQ. 1) GO TO 20
0056  164  NTMAX = NFC*NPR + 1
0057  165  NTT = NTMX + 1
0058  166  NT = NTMAX + 1
0059  17  NTT = NTT - 1
0060  18  DO 18 I=1,NFC
0061  19  NT = NT - 1
0062  20  DIFTT(NT) = DIFTT(NTT)/NFC
0063  21  CONTINUE
0064  22  IF(NTT .GT. 1) GO TO 17
0065  23  TIT(1) = TMINT
0066  24  DO 19 I=2,NTMAX
0067  25  TIT(I) = TIT(I-1) + DIFT(I)
0068  26  CONTINUE

C  .  SCALE TI FROM 1 TO NM=NTMAX/2 TO COVER INTERVAL (TIMIN,TMDMX)
C

C

0069  27  NM = NTMAX/2
0070  28  FACT = (TMDMX-TIMIN)/(TIT(NM)-TMINT)
0071  29  TIT(1) = TIMIN
0072  30  DIFT(1) = 1.
0073  31  DO 21 I=2,NM
0074  32  DIFT(I) = FACT*DIFT(I)
0075  33  TIT(I) = TIT(I-1) + DIFT(I)
0076  34  CONTINUE

C  .  SCALE TI FROM NM+1 TO NTMAX TO COVER INTERVAL (TMDMX,TMAX)
C

C

0077  35  FACT = (TMAX-TMDMX)/(TMAX-TIT(NM))
0078  36  DO 22 I=NM+1,NTMAX
0079  37  DIFT(I) = FACT*DIFT(I)
0080  38  TIT(I) = TIT(I-1) + DIFT(I)
0081  39  CONTINUE

C  .  FIND RESPONSE FOR EACH ANGLE AND ACCUMULATE IN TRP
C

C

0082  40  NTGBNS = 2*NTARGM
0083  41  DO 30 I=1,NTMAX
0084  42  TRP(I) = 0.
0085  43  CONTINUE
0086  44  TQFPP = 0.
0087  45  DO 40 K=NDMN,NDMx
0088  46  CALL ARESPIT(1,EN,K,IPOS,NTGBNS)
0089  47  IF(IPOS .EQ. 0) GO TO 40
0090  48  TQFPP = TQFPP + TOFP
0091  49  TP = TIT(1)
0092  50  RP = 0.
0093  51  DO 35 J=2,NTMAX
0094  52  TM = TP
0095  53  RM = RP
TP = TI(J)
RP = DELRES(TP)
TRP(J-1) = TRP(J-1) + (RP - RM)
CONTINUE
CONTINUE
C
C ... FORM CUMULATIVE RESPONSE
C
TOT = 0.
SENSI(1) = 0.
DO 50 I=2,NTMAX
TOT = TOT + TRP(I-1)
SENSI(I) = TOT
50 CONTINUE
TOFP = TOFPP
IPOS = 1
RETURN
END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE1</td>
<td>581</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>IDATA</td>
<td>10</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>VARS</td>
<td>113</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>TEMPS</td>
<td>3</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>A</td>
<td>751</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>B</td>
<td>373</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>RESP0</td>
<td>251</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>RESP1</td>
<td>9</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>RESP3</td>
<td>251</td>
<td>RW, D, OVR, GBL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 011216 2375
SUBROUTINE ARESPT(INCTRL, EN, J, IPPOS, NTTRT)

C... Computes time dependent response of neutrons of energy, EN,
C... at Jth angular mesh point. IPPOS = 0 implies zero response.
C... IF(INCTRL .EQ. 0) only time bins are computed.
C... NTTRT is maximum number of CH2 foil divisions.
C... Geometry is assumed to be: target perpendicular to
C... neutron beam, circular apertures.
C... Called by - RESPT
C... Calls - ENERGY
C... RANGE
C... TOFF
C... ZEFF
C... 
C... Routine GEOM must be called before calling ARESPT to set up
C... angular arrays and compute OMEGA.
C... 8-FEB-82
C... 
C... /RESP1/
C... SIGH = (DIFF.(N,P) XSECT)/(COS(THETA))
C... TOFP = response weighted T-O-F (N+P) from ZSS to detector.
C... NTARGM = maximum number of target thickness steps.
C... ZSS = theor. foil station.
C... 
C... /RESP3/
C... NTMX = number of entries in time-dependent response.
C... TMINT = minimum time of response.
C... TMAXT = maximum time of response.
C... TIT = time entries.
C... SNSI = cumulative response to time TIT.
C... DIFT = TIT(I) - TIT(I-1).

COMMON/A/CS(75), WT(75), CST(75), CSQ(75), AVR(75), NA
COMMON/B/E(60), SENS(60), TNL, TNH, NE, TOF(60), DTIT, ZFV, RFV
COMMON/C/RD, ZS, ZTAQ, THETA, TTHICK, ZAD, THMX, THMN, OMEGA
COMMON/D/DTHICK, DERUN
COMMON /RESP1/ SIGH, TOFP, NTARGM, ZSS, TMID
COMMON /RESP3/ NTMX, TMINT, TMAXT, TIT(41), SNSI(41), DIFTT(41)
DIMENSION EP(41), ZEFF(41), ZSS, TMID
DIMENSION C(25), D(25), CA(25), DA(25), ZPOLY(7), ZAL(7)
LOGICAL DERUN

DATA C/ 0.001, 0.00291, 0.10173, 0.821486778E+1, 0.97903919E-1, 0.10173,
1 8.2677984E-1, 1.5429533E-1, 1.1416341E-1, 3.713083E-2, 6.8796021E-3,
2 7.520038E-4, 4.5281766E-5, 1.606486E-6,
3 7.4128067E-1, 1.6310188E+0, 4.9908146E-2, -1.4756338E-2,
4 1.6253492E-2, -4.354309E-3, -2.103061E-3, 1.0201402E-3,
5 3.7088954E-5, -6.9876078E-5, 8.4381196E-6/

DATA D/ 0.00291, 0.001, 0.10173,
1 7.0718777E-1, 1.0397812E+0, -1.4575638E0, -6.9523209E-1,
2 8.3148688E-1, 9.7903919E-1, 4.3677825E-1, 1.0537574E-1,
THE ARRAYS C, D, CA, DA CONTAIN THE COEFFICIENTS WHICH FIT
THE RANGE-ENERGY TABLES OF JOSEPH F. JANNI,
(PRIVATE COMMUNICATION JAN'82, TO BE PUBLISHED NDT)
A 10TH ORDER POLYNOMIAL FIT IN LOG(R) VS LOG(E) WAS USED FOR
THE REGIONS (0.001-0.100-MEV) AND (0.100-40.0-MEV).
A LINEAR INTERPOLATION TO (0.001-MEV) IS USED BELO.
THE COEFFICIENTS ARE DEFINED AS FOLLO.
COEF(1) = (ENERGY OR RANGE) AT 0.001
COEF(2) = (RANGE OR ENERGY) AT 0.001
COEF(3) = (ENERGY OR RANGE) AT 0.100
REMAINING 22 COEFFICIENTS ARE FOR FIT IN REGIONS (0.001-0.100) AND
(0.100-40.0)
C
C GIVES RANGE OF PROTONS IN POLYETHYLENE (MG/CM**2)
D GIVES PROTON ENERGY FOR A GIVEN RANGE IN POLY.
CA, DA GIVE CORRESPONDING VALUES FOR ALUMINUM.

C ZPOLY AND ZAL CONTAIN THE COEFFICIENTS OF A 6TH ORDER FIT OF A
POLYNOMIAL TO LOG(Z) VS LOG(E) TO DATA DERIVED FROM CALCULATIONS
OF JOSEPH F. JANNI (P.C. JAN'82, TO BE PUBLISHED IN NDT). THE
DATA DESCRIBE THE DETECTED CHARGE AS A FUNCTION OF PROTON ENERGY.
ZEFF = 0. IF E < 0.004 MEV, ZEFF = 1.0 IF E .GE. 0.400 MEV.

C FOR EACH ANGLE IN THE MESH THE FOLLOWING APPLY
CS = COSINE OF ANGLE OF SCATTERING
CSQ = COSINE**2
CST = COSINE OF ANGLE RE NORMAL TO FILTER
WT = GEOMETRICAL WEIGHTING FACTOR
AVR = AVERAGE PROTON FLT. PTH. TO DEFINING COLLIMATOR
CONST = (2*ELECT. CHARGE*AVOGADROS #)/(1000MG/CM*MOL. WT.)*OMEG
IWEXS.

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C
C ... WT(J) = WEIGHTING FACTOR (FROM GEOM) FOR ANGLE J.
C ... CONST = EARLIER DEFINITION IN THIS ROUTINE.

FACTOR = CONST*SIGH*CSG(J)*WT(J)
TOT = 0.
PRES(1) = 0.
SNSI(1) = 0.

C ... **********...
C
C ... COMPUTE RESPONSE FOR UNFILTERED FARADAY CUP.
C
IF(FTHICK .GT. 0.) GO TO 110
C
C ... SET UP TARGET THICKNESS BINS. TIME RESOLUTION TO EQUAL DTAU, UNLESS
C ... MORE THAN NTRT TOTAL MESH POINTS, FOR HALF OF EFFECTIVE TARGET
C ... NEAREST DETECTOR.
C
Z = AVR(J) + ZAD/CST(J)
TNEUT = TOFF(EN, ZS-ZSS, EON)
THIN = TOFF(EPMX, Z, EOP) + TNEUT
EMID = ENERGY(RPMX-0.5*TEFF, D, 25)
TMID = TOFF(EMID, Z, EOP) + TNEUT
NTARG = 2*(TMID-TMIN)/DTAU
IF(NTARG .LT. 4) NTARG = 4
IF(NTARG .GT. NTRT) NTARG = NTRT
DELT = TEFF/NTARG
NTMX = NTARG + 1

C ... COMPUTE PROTON ENERGIES, GEFF, AND TOF FROM THEOR. STA. AT EACH MESH
C
TPOL = -DELT
DO 100 I=1,NTMX
TPOL = TPOL + DELT
EP(I) = ENERGY(RPMX-TPOL, D, 25)
EPP = EP(I)
IF(EPP .LT. 0.004) EPP = 0.004
TIT(I) = TOFF(EPP, Z, EOP) - TNEUT
GEFF(I) = ZEFF(ZPOLY, EPP(I))
CONTINUE
IF(ICNTRL .EQ. 0) GO TO 400
C
C ... COMPUTE PARTIAL RESPONSE, INTEGRAL RESPONSE, AND RESPONSE WEIGHTED
C ... (PROTON + NEUTRON) TOF FROM THEOR. STA.
C ... DRESP IS 0.5*RESPONSE PER THICKNESS STEP, LESS EFFECT OF GEFF.
C
DRESP = 0.5*FACTOR*DELT
DO 102 I=2,NTMX
QFACT = 2.
IF(EP(I) .GE. 0.4) GO TO 101
QFACT = GEFF(I-1) + GEFF(I)
IF(EP(I-1) .LE. 0.4) GO TO 101
DIFEF = 2./(EP(I-1) - EP(I))
QFACT = DIFEF*(0.4-EP(I))*GEFF(I) + (EP(I-1)-0.4)*GEFF(I-1)
CONTINUE
DO 101 I=2,NTMX
PRES(I) = DRESP*QFACT
TOT = TOT + PRES(I)
SNB(I) = TOT
TOFP = TOFP + 0.5*(TIT(I) + TIT(I-1))*PRES(I)
CONTINUE
GO TO 400
C... *****************************************************
C... COMPUTE RESPONSE FOR AL-FILTERED FARADAY CUP
C
110 IF(DTHICK .GT. 0.) GO TO 210
C
C... SET UP TARGET THICKNESS BINS. SAME CRITERIA USED AS FOR
C... UNFILTERED F.C.
C
Z1 = AVR(J)
Z2 = IAD/CST(J)
TNEUT = TOFF(EN, ZS-ZSS, EON)
EPFMX = ENERGY(RPMX-FTHKP, D.25)
C
C... SEE IF MAX ENERGY TOO SMALL FOR DETECTION
C
IF(EPFMX .LT. 0.006) RETURN
TMIN = TOFF(EPMX, Z1, EOP) + TOFF(EPMX, Z2, EOP) + TNEUT
EMID = ENERGY(RPMX-0.5*TEFF, D.25)
EFMID = ENERGY(RPMX-0.5*TEFF-FTHKP, D.25)
TMID = TOFF(EMID, Z1, EOP) + TOFF(EFMID, Z2, EOP) + TNEUT
NTARG = 2*(TMID-TMIN)/DTAU
IF(NTARG .LT. 4) NTARG = 4
IF(NTARG .GT. NTRT) NTARG = NTRT
DELT = TEFF/NTARG
NTMX = NTARG + 1
C
C... COMPUTE PROTON ENERGIES AFTER FILTER, GEFF FOR AL, AND
C... (NEUTRON + PROTON) TOF FROM ZSS AT EACH MESH POINT.
C
TPOL = -DELT
DO 200 I=1,NTMX
TPOL = TPOL + DELT
R = RPMX - TPOL
EP1 = ENERGY(R, D.25)
EP(I) = ENERGY(R-FTHKP, D.25)
EPP = EP(I)
IF(EPP .LT. 0.004) EPP = 0.004
TIT(I) = TOFF(EPP, Z1, EOP) + TOFF(EPP, Z2, EOP) + TNEUT
GEFF(I) = ZEFF(ZAL, EP(I))
CONTINUE
DO 202 1=2,NTMX
CONTINUE
GO TO 400
C
C... COMPUTE PARTIAL RESPONSE, INTEGRAL RESPONSE AT EACH MESH POINT
C
DRESP = 0.5*FACTOR*DELT
DO 202 I=2,NTMX
QFACT = 2.
IF(EPP .GE. 0.4) GO TO 201
QFACT = GEFF(I-1) + GEFF(I)
IF(EPP .LE. 0.4) GO TO 201
DIFEF = 2./(EP(I-1) - EP(I))
QFACT = DIFE*((0.4-EP(I))*QEFF(I) + (EP(I-1)-0.4)*QEFF(I-1))

PRES(I) = DRESP*QFACT

TOT = TOT + PRES(I)

SNSI(I) = TOT

TOFP = TOFP + 0.5*(TIT(I)+TIT(I-1))*PRES(I)

CONTINUE

GO TO 400

C ... ***********...

C ... COMPUTE RESPONSE FOR DELTA-E DETECTOR.

C ... USE SAME CRITERIA TO SET UP TARGET MESH AS FOR UNFILTERED F.C.

C

Z1 = AVR(J)

Z2 = ZAD/CST(J)

TNEUT = TOFF(EN, ZS-ZSS, EON)

RTEMP = RPMX - TINIT

EPMXP = ENERGY(RTEMP, D, 25)

EPFMX = ENERGY(RTEMP-FTHKP, D, 25)

C ... SEE IF MAX ENERGY TOO SMALL FOR DETECTION

C

IF(EPFMX . LT. 0.006) RETURN

TMIN = TOFF(EPMXP, Z1, EOP) + TOFF(EPFMX, Z2, EOP) + TNEUT

TTEMP = RTEMP - 0.5*TEFF

EMID = ENERGY(TTEMP, D, 25)

EFMID = ENERGY(TTEMP-FTHKP, D, 25)

TMID = TOFF(EMID, Z1, EOP) + TOFF(EFMID, Z2, EOP) + TNEUT

NTARG = 2*(TMID-TMIN)/DTAU

IF(NTARG . LT. 4) NTARG = 4

IF(NTARG . GT. NTRT) NTARG = NTRT

DELT = TEFF/NTARG

NTMX = NTARG + 1

C ... COMPUTE PROTON ENERGIES AFTER FILTER, QEFF, AND (NEUTRON+PROTON)

C ... TOF FROM ZSS AT EACH MESH. GEFF IS (1+ZEFF(ZAL, PROTON ENERGY

C ... AFTER LEAVING DELTA-E) IF PROTON PASSES THRU DELTA-E.

C ... OTHERWISE GEFF = ZEFF(ZAL, PROTON ENERGY LEAVING FILTER).

C

ETEST = ENERGY(DTHKP, D, 25)

TPOL = TINIT - DELT

IGT = 0

DO 302 I=1, NTMX

TPOL = TPOL + DELT

RTEMP = RPMX - TPOL

EP(I) = ENERGY(RTEMP, D, 25)

EP = EP(I)

IF(EPP .LT. 0.004) EPP = 0.004

IF(EP(I) . LT. ETTEST) GO TO 300

PROTON DOES NOT PASS THRU DELTA-E

GEFF(I) = ZEFF(ZAL, EP(I))

GO TO 302

C ... PROTON PASSES THRU DELTA-E


GEFF(I) = 1. - ZEFF(ZAL, EP(I))
I QT = I

IF(1 + CNTRL .EQ. 0) GO TO 400

C
C .. COMPUTE PARTIAL RESPONSE, INTEGRAL RESPONSE, AND RESPONSE WEIGHTED
C .. (NEUTRON+PROTON) TOF AT EACH MESH POINT.
C
DRESP = 0.5*FACTOR*DELT
DO 304 I = 2, NTMX
QFACT = QEFF(I) - QEFF(I-1)
IF(I .LT. (IQT+1)) GO TO 1302
IF(I .GT. (IQT+1)) GO TO 1301
C .. IF PROTON HAS JUST PENETRATED DELTA E AT BEGINNING OF BIN
QFACT = 0.5*(QFACT+2.)
GO TO 1302

C .. PROTON DOES NOT PENETRATE THRU DELTA E IN THIS BIN
IF((EP(I) .GE. 0.4) .OR. (EP(I-1) .LE. 0.4)) GO TO 1302
DIFEF = 2./(CEP(I-1) - EP(I))
QFACT = DIFEF*((0.4-EP(I))*QEFF(I) + (EP(I-1)-0.4)*QEFF(I-1))

1301 IF((EP(I) .GE. 0.4) .OR. (EP(I-1) .LE. 0.4)) GO TO 1302
DO 1302 I = 2, NTMX
QFACT = DIFEF*(QEFF(I) - QEFF(I-1))
TOT = TIT + PRES(I)
SNST(I) = TOT
TOFP = TOFP + 0.5*(TIT(I) + TIT(I-1))*PRES(I)
CONTINUE

400 DIFFT(I) = 1.
DO 402 I = 2, NTMX
IF(DIFFT(I) .LE. 0.) GO TO 403
DIF = TIT(I) - TIT(I-1)
CONTINUE
GO TO 404

C .. TRUNCATE DISTRIBUTION IF PROTONS OF LESS THAN 4KEV ARE INVOLVED
C
NTMX = I - 1
TMINT = TIT(I)
TMA X = TIT(NTMX)
IPDS = 1
RETURN
END
Total Space Allocated = 016270  3676
FUNCTION PRESP(T)
C
C ROUTINE TO INTERPOLATE AT TIME, T, IN CUMULATIVE RESPONSE FUNCTION
C SENSI(I) WHICH IS ENTERED AS TABLE VS TIP(I).
C
C NTMAX = NUMBER OF ENTRIES.
C TIMINP = TIP(1) = TIME BEFORE WHICH PRESP(T) = 0.
C TIMAXP = TIP(NTMAX) = TIME BEYOND WHICH PRESP(T) = SENSI(NTMAX).
C TIP = TIME ENTRIES RE TIME NEUTRON ARRIVES AT FSS.
C SENSI = CUMULATIVE PROBABILITY OF RESPONSE BY TIME, TI.
C DIFTP = DIFFERENCE BETWEEN ITH AND (I-1)TH TIME ENTRIES.
C
COMMON /RESP2/ TIMINP, TIMAXP, TIP(NTMAX), DIFTP(NMAX)
COMMON /RESP0/ NTMAX, TIMINP, TIMAXP, TIP(NTMAX), DIFTP(NMAX)
COMMON /RESP0/ NTMAX, TIMINP, TIMAXP, TIP(NTMAX), DIFTP(NMAX)
COMMON /RESP2/ NTMAX, TIMINP, TIMAXP, TIP(NTMAX), DIFTP(NMAX)
COMMON /RESP2/ NTMAX, TIMINP, TIMAXP, TIP(NTMAX), DIFTP(NMAX)

PRESP = 0.
IF(T .LE. TIMINP) RETURN
IF(T .LT. TIMAXP) GO TO 10
PRESP = SENSI(NTMAX)
RETURN

C T IS WITHIN RANGE
C
DO 20 I=2,NTMAX
IF(T .GT. TIP(I)) GO TO 20
PRESP = SENSI(I-1) + (T-TIP(I-1))*(SENSI(I)-SENSI(I-1))/DIFTP(I)
RETURN

CONTINUE
RETURN
END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CODE1</td>
<td>000260</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>$DATA</td>
<td>000004</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>$VARS</td>
<td>000002</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>$TEMPS</td>
<td>000002</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>RESP0</td>
<td>000766</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>RESP2</td>
<td>000520</td>
<td>RW, D, OVR, GBL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 001776 511
FUNCTION DELRES(T)

C ... ROUTINE TO INTERPOLATE AT TIME, T, IN CUMULATIVE RESPONSE FUNCTION
C ... SNSI(I) WHICH IS ENTERED AS TABLE VS TIT(I).

C ... NTMX = NUMBER OF ENTRIES.
C ... TMIN = TIT(1) = TIME BEFORE WHICH DELRES(T) = 0.
C ... TMAX = TIT(NTMX) = TIME BEYOND WHICH DELRES(T) = SNSI(NTMX).
C ... TIT = TIME ENTRIES RE TIME NEUTRON ARRIVES AT ZSS.
C ... SNSI = CUMULATIVE PROBABILITY OF RESPONSE BY TIME, TIT.
C ... DIFTT = DIFFERENCE BETWEEN ITH AND (I-1)TH TIME ENTRIES.

COMMON /RESP3/ NTMX, TMIN, TMAX, TIT(41), SNSI(41), DIFTT(41)

0002 DELRES = 0.
0003 IF(T .LE. TMIN) RETURN
0004 IF(T .LT. TMAX) GO TO 10
0005 DELRES = SNSI(NTMX)
0006 RETURN

C ... T IS WITHIN RANGE

C

0008 10 DO 20 I=2,NTMX
0009 IF(T .GT. TIT(I)) GO TO 20
0010 DELRES = SNSI(I-1) + (T-TIT(I-1))*(SNSI(I)-SNSI(I-1))/DIFTT(I)
0011 RETURN
0012 20 CONTINUE
0013 RETURN
0014 END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE1</td>
<td>000260</td>
<td>88 RW, I, CON, LCL</td>
</tr>
<tr>
<td>DATA</td>
<td>000004</td>
<td>2 RW, D, CON, LCL</td>
</tr>
<tr>
<td>VARS</td>
<td>000002</td>
<td>1 RW, D, CON, LCL</td>
</tr>
<tr>
<td>TEMPS</td>
<td>00002</td>
<td>1 RW, D, CON, LCL</td>
</tr>
<tr>
<td>RESP3</td>
<td>000766</td>
<td>251 RW, D, OVR, GBL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 001256 343
SUBROUTINE TINT(EX, TX, E, T, N)

C ... ROUTINE TO LINEARLY INTERPOLATE IN ARRAY T(E) TO FIND TX AT EX.

C

DIMENSION E(N), T(N)

IF(EX .GT. E(1)) GO TO 10

TX = T(1)

RETURN

IF(EX .LT. E(N)) GO TO 20

TX = T(N)

RETURN

IF(EX .LT. E(I)) GO TO 40

C

CON I NUE

SL = (T(I) - T(I-1)) / (E(I) - E(I-1))

RETURN

END

NAME     SIZE   Attributes

$CODE1   000366  123       RW, I, CON, LCL
$DATA    00040   16        RW, D, CON, LCL
$VARS    00096   3          RW, D, CON, LCL
$TEMPS   00002   1          RW, D, CON, LCL

Total Space Allocated = 000436   143
FUNCTION TOFF(E,Z,EO)

C ... ROUTINE TO COMPUTE TIME-OF-FLIGHT OF PARTICLE WITH ENERGY, E,
C ... REST MASS, EO, OVER FLIGHT PATH, Z. UNITS ARE MEV, CM.
C ... A LOWER LIMIT OF 0.001MEV IS IMPOSED. THIS IS BASED ON PROTON
C ... ENERGY BELOW WHICH AVERAGE CHARGE STATE IS 0.
C

DATA C/29.97928/

EP = E

IF(EP .LT. 0.001) EP = 0.001

X = EP/EO

TOFF = Z*(1.+X)/(C*SQRT(X*(2.+X)))

RETURN

END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>#CODE1</td>
<td>000130</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>#DATA</td>
<td>000004</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>#IDATA</td>
<td>000004</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>#VARS</td>
<td>000014</td>
<td>RW, D, CON, LCL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 000154 54
FUNCTION ZEFF(Z, E)
C
FUNCTION WHICH COMPUTES THE EFFECTIVE DETECTED CHARGE OF A
C.. PROTON IN MATTER AT ENERGY E. 'MATTER' IS DESCRIBED BY THE
C.. COEFFICIENTS, Z, OF A 6TH ORDER POLYNOMIAL FIT TO CALCULATIONS
C.. OF JOSEPH F. JANNI (P.C. JAN '82) TO BE PUBLISHED IN NDT) FOR
C.. CH2 AND AL.
C.. THE POLYNOMIAL IS FIT TO LOG(ZEFF) VS LOG(E)
C.. ZEFF = 0. FOR E .LE. 0.004MEV
C.. ZEFF = 1. FOR E .GE. 0.400MEV
C
DIMENSION Z(7)
C
IF(E .GT. 0.004) GO TO 10
ZEFF = 0.
RETURN
10 IF(E .LT. 0.400) GO TO 20
ZEFF = 1.
RETURN
20 ELN = ALOG(E)
Y = 0.
DO 30 I=1,6
K = 8-I
Y = (Y + Z(K)) * ELN
30 CONTINUE
ZEFF = EXP(Z(1) + Y)
RETURN
END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE1</td>
<td>95</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>PDATA</td>
<td>4</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>IDATA</td>
<td>7</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>VARS</td>
<td>6</td>
<td>RW, D, CON, LCL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 000340 112
SUBROUTINE GEOM

COMPUTES GEOMETRICAL FRAMEWORK FOR INTEGRATION OF RESPONSE

CHANGED CODE TO USE MONTE CARLO METHOD TO CHOOSE POINT PAIRS ON TARGET AND DETECTOR APERATURE INSTEAD OF DISCRETE ORDNATES.

CHANGE METHOD OF SELECTING RANDOM POINTS TO UTILIZE SYMMETRY AND HENCE REDUCE COMPUTATION TIME.

CODE ASSUMES UNIFORMLY ILLUMINATED TARGET PERPENDICULAR TO NEUTRON BEAM. TARGET AND APERATURE ARE CIRCULAR.

APERATURE IS PERPENDICULAR TO CENTRAL SCATTERING ANGLE.

COMMON/A/CS (75), WT(75), CST(75), CSG(75), AVR(75), NA
COMMON/B/E(60), SENS(60), TNL, TNM, TNH, NE, TOF(60), DTAU, ZFV, RFV
COMMON/C/RT, RD, ZS, ZTAO, THETA, TTHICK, ZAD, TTH, ZSN, OMEGA
COMMON /MCCORD/ CTMC
DIMENSION THTA(75), DTHT(76), XTTT(2), YTTT(2), XDTT(2)
DATA RAD/0.01745328/, EOP/938.256/, CD/29.79728/

THETA = ZTAO

ZDO = ZTAO*SNO

RDCS = RD*CSO

RDSN = RD*SNO

OMEGA = 0.0

DO 30 I=1,75
30 CONTINUE

THMX = THTA1/RAD

THMN = THTA1/RAD

ZTDO = ZTAO+ZAD

DO 30 I=1,75
30 CONTINUE

ZMX = ZAD+(ZO+RDSN)/COS(THTA2)

COMPUTE MIN AND MAX SCATTERING ANGLES

THTA1 = ATAN((XDO-RDCS-RT)/(ZO+RDSN))

THTA2 = ATAN((XDO+RDCS+RT)/(ZO-RDSN))

THMX = THTA1/RAD

THMN = THTA1/RAD

ZTDO = ZTAO+ZAD

DO 30 I=1,75
30 CONTINUE

ZMX = ZAD+(ZO+RDSN)/COS(THTA2)

CE... Compute integration step size
C DEN = (DTAU/EP)*(DER(EP) RE TOF(EP)) AT EP=0.1MEV, ZMX
C
0026 X   = 1.0D0
0027 DEN = (DTAU*CO)/(X*ZMX)***(X*(2.+X))**1.5
0030 DEN2 = DEN/2.0
0031 RDS = RD*RD
C TARGET FOV PARAMETERS
0032 RFS = RFV*RFV
0033 XAO = XDO*ZFV/ZTAO
C
C COMPUTE ANGULAR BINS
C NA = NUMBER OF ANGULAR BINS
C DTHT = (DER(THETA) RE TOF) * DTAU
C = (DER(THETA) RE EP) * (DER(EP) RE TOF) * DTAU
C = (DER(THETA) RE EP) * EP * DEN
C
0034 CS(1) = COS(THTA1)
0035 CSG1 = CS(1)*CS(1)
0036 THTA(1) = THTA1
0037 DTHT(1) = DEN2/TAN(THTA1)
0038 DO 50 I=2,75
0039 DTHT(I) = DEN2/TAN(THTA(I-1))
0040 THTA(I) = THTA(I-1)+DTHT(I)
0041 IF(THTA(I) .GE. THTA2) GO TO 55
0042 CS(I) = COS(THTA(I))
0043 CSG(I) = CS(I)*CS(I)
0044 50 CONTINUE
0045 NA = 75
0046 GO TO 60
0047 55 NA = I
0048 60 THTA(NA) = THTA2
0049 CS(NA) = COS(THTA2)
0050 DTHT(NA) = THTA(NA)-THTA(NA-1)
0051 DTHT(NA+1) = DTHT(NA)
0052 CSG(NA) = CS(NA)*CS(NA)
C
C REPLACE DTHT BY 0.5*DTHT FOR LATER USE IN SORTING
C
0053 DO 62 I=1,NA+1
0054 DTHT(I) = 0.5*DTHT(I)
0055 62 CONTINUE
0056 ZXDO = ZTAO/CSO
C COUNTER FOR MATCH ON SOLID ANGLE BETWEEN MESH POINTS
C
C CNT = 0.
C
C DO COMPUTATION FOR GEOMETRICAL PARAMETERS FOR 20000 RANDOMLY
C CHosen POINT PAIRS ON TARGET AND DETECTOR.
C
C CHOOSE SEED FOR STARTING RANDOM NUMBER GENERATOR
C
0058 SEED = 3.3333*SECNDS(0.)
C
C FORM AVERAGES FOR 1000*NA TARGET-DETECTOR PAIRS
DO 300 I = 1, 125

C ... CHOOSE POINT WITHIN TARGET
C
RTMC = RT*SQRT(RAN(SEED))
TMC = 6.283185*RAN(SEED)
XTTT(1) = RTMC*COS(TMC)
YTTT(1) = RTMC*SIN(TMC)
XTTT(2) = -XTTT(1)
YTTT(2) = -YTTT(1)

C ... CHOOSE POINT WITHIN DETECTOR APERTURE
C
RDMC = RD*SQRT(RAN(SEED))
TMC = 6.283185*RAN(SEED)
XDTT(1) = RDMC*COS(TMC)
YD = RDMC*SIN(TMC)
XDTT(2) = -XDTT(1)

C ... COMPUTE FOR ALL COMBINATIONS OF +, - X, Y, X*D
C ... USES INCIDENT SYMMETRY, DISCARDS Y-SYM(DET) AS REDUNDANT.
C
DO 300 IXT=1,2
DO 300 IYT=1,2
DO 300 IXD=1,2
XT = XTTT(IXT)
YT = YTTT(IYT)
XD = XDO + CSO*XDTT(IXD)

C ... ZOS = DISTANCE FROM TARGET MESH TO DETECTOR PLANE.
C ...
ZOS = ZTAO-XT*SINO
C ...
RXF = RATIO OF DISTANCES FROM (DET. TO FOV) TO (DET. TO TARG.)
C ...
RXF = (ZOS+ZFV-ZTAO)/ZOS
DXS = (XT-XD)**2
XD = ((XD-XDO)/CSO)**2
C ...
ZTDS = DISTANCE PARALLEL TO BEAM AXIS BETWEEN TARG. MESH
C ...
AND DETECTOR MESH.
C ...
ZTDS = ZXDO-XD*TNO
CNT = CNT+1.

C ... CHECK FOR FOV CONSTRAINT
C
IF(RFV .LE. 0.0) GO TO 160
XA = (XD-XT)*RXF+XT
C ...
XC = X-COMPONENT OF (TARG-DET) LINE AT ZFV.
XC = XAO = X-COMPONENT OF CENTER OF FOV AT ZFV.
C ...
XAS = SQUARE OF X-COMPONENT OF (TARG-DET) LINE AT ZFV
C ...
RELATIVE TO CENTER OF FOV.
XAS = ((XA-XAO)/CSO)**2
C ...
YAS = SQUARE OF Y-COMPONENT OF (TARG-DET) LINE AT ZFV.
YAS = ((YD-YT)*RXF+YT)**2
C ...
IF NOT IN FOV, SKIP REST OF LOOP
C
IF((XAS+YAS) .GT. RFS) GO TO 300
C
160 CONTINUE
FORTRAN IV-FLUS V3.0-3
16:24.35  20-May-82  Page 36
NUEXSUB.FTN: 3 /TR: BLOCKS/WR

0090  \texttt{RTD = SQRT(DXS + (YT-YD)**2)}
C
\texttt{C .. RTD = DISTANCE PERPENDICULAR TO BEAM AXIS BETWEEN TARGET MESH}
\texttt{C . .. AND DETECTOR MESH.}
\texttt{C . . ZTDS = DISTANCE PARALLEL TO BEAM AXIS BETWEEN TARGET MESH}
\texttt{C . .. AND DETECTOR MESH.}
\texttt{C .. THT = ANGLE OF SCATTERING.}
\texttt{C ... = ATAN(RTD/ZTDS)}
\texttt{C}
\texttt{C}
0071  \texttt{THT = ATAN(RTD/ZTDS)}
\texttt{C}
\texttt{C . THTN = ANGLE OF LINE OF SCATTER RELATIVE TO TARGET NORMAL}
\texttt{C . ZOS = COS(THTN) * (DISTANCE BETWEEN TARG. MESH AND DET. MESH)}
\texttt{C . . = DISTANCE FROM SCATTERING POINT TO DETECTOR PLANE}
\texttt{C . . R = ZTDS/COS(THT) = DISTANCE BETWEEN TARG. MESH AND DET. MESH.}
\texttt{C . OMEGA = R*COS(THT) * (1/R)**3}
\texttt{C}
0092  \texttt{COST = COS(THT)}
0093  \texttt{OMEGA = OMEGA+ZOS*(COST/ZTDS)**3}
\texttt{C}
\texttt{C . CHECK TO SEE IF SCATTERING ANGLE MATCHES ANGULAR MESH}
\texttt{C . IF SO, ACCUMULATE WT, AVR, CST.}
\texttt{C . SUCCESSIVE APPROXIMATION IS USED TO FIND BIN}
\texttt{C}
0094  \texttt{KMIN = 1}
0095  \texttt{KMAX = NA}
0096  \texttt{DO 200 IK=1,8}
0097  \texttt{K = (KMIN+KMAX)/2}
0098  \texttt{TEST = THTA(K)}
0099  \texttt{IF(THT .GE. (TEST-DTHT(K))) KMIN = K}
0100  \texttt{IF(THT .LE. (TEST+DTHT(K+1))) KMAX = K}
0101  \texttt{IF(KMIN .EG. IR:.MAX) GO TO 201}
0102  \texttt{200 CONTINUE}
\texttt{C}
\texttt{WT = TOTAL MATCHES OF MESH POINTS AT ANGLE K.}
0103  \texttt{201 WT(K) = WT(K)+1.0}
\texttt{C}
\texttt{AVR = SUM OF DISTANCES BETWEEN TARGET MESH AND DETECTOR MESH}
\texttt{C . AT ANGLE K.}
0104  \texttt{AVR(K) = AVR(K)+ZTDS/COST}
\texttt{C . AT THIS POINT CST = SUM OF DISTANCES FROM SCATTERING POINT}
\texttt{C . TO DETECTOR PLANE AT ANGLE K.}
0105  \texttt{CST(K) = CST(K)+ZOS}
\texttt{C}
0106  \texttt{300 CONTINUE}
0107  \texttt{CTMC = 1000. *NA}
\texttt{C}
\texttt{**DEBUG**}
\texttt{C}
0108  \texttt{WRITE(4,7000) CTMC, (THTA(I),I=1,NA)}
0109  \texttt{7000 FORMAT(/,1X,1PE12.3,': PARTICLES',/,(1X,5E10 3))}
0110  \texttt{WRITE(4,7001) (WT(I),I=1,NA)}
0111  \texttt{7001 FORMAT(/,1X,1PE10.3)}
\texttt{C}
\texttt{C END OF INTEGRATION OVER ALL VARIABLES}
\texttt{C}
\texttt{COMPUTE WT,CST,AVR FOR EXTREME ANGLES}
\texttt{C}
```fortran
IF(WT(1) . LE. 0.0) WT(1) = 1.0
IF(WT(NA) . LE. 0.0) WT(NA) = 1.0
CST(I) = (ZTAO-RT*SNO)*WT(I)
CST(NA) = (ZTAO+RT*SNO)*WT(NA)
AVR(I) = (ZTAO-CSO+RD*SNO)*WT(I)/COS(THTA1)
AVR(NA) = (ZTAO-CSO-RD*SNO)*WT(NA)/COS(THTA2)

C ... NORMALIZE RESULTS
C
SUM = 0.0
DO 600 I = 1, NA
  IF(WT(I) . LE. 0.0) WT(I) = 1.0
  SUM = SUM+WT(I)
  IF(AVR(I) . LE. 0.0) GO TO 600
C ... CST IS REDEFINED TO BE AVERAGE COS(THTN)
  CST(I) = CST(I)/AVR(I)
  AVR(I) = AVR(I)/WT(I)
  600 CONTINUE
C ... CHECK FOR POSSIBLE ERRORS AND NORMALIZE WT TO 1 FOR INTEGRAL.
C
DO 700 I=2, NA-1
  IF(CST(I) . LE. 0.0) CST(I) = (CST(I+1)+CST(I-1))/2.0
  IF(AVR(I) . LE. 0.0) AVR(I) = (AVR(I+1)+AVR(I-1))/2.0
C ... NORMALIZE WEIGHTS SUCH THAT TOTAL WT = 1
  700 WT(I) = WT(I)/SUM
  WT(1) = WT(1)/SUM
  WT(NA) = WT(NA)/SUM
C ... AVERAGE THE 1/R**2 CONTRIBUTIONS AT EACH MESH POINT PAIR
C ... AND MULTIPLY BY THE AREA OF DEFINING COLLIMATOR
C
OMEGA = 3.1415927*RDS*OMEGA/CNT
RETURN
END
```

**PROGRAM SECTIONS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE1</td>
<td>004016</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>DATA</td>
<td>000110</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>VARS</td>
<td>001466</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>TEMPS</td>
<td>000020</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>A</td>
<td>002736</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>B</td>
<td>001352</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>C</td>
<td>000054</td>
<td>RW, D, OVR, GBL</td>
</tr>
<tr>
<td>MCCOM</td>
<td>000004</td>
<td>RW, D, OVR, GBL</td>
</tr>
</tbody>
</table>

'Total Space Allocated = 012224  2634
C ... STRING FINDER SUBROUTINE
C INTERFACE: CALL FINDIT(LUN, STRING, LEN, IPOS)
C WHERE  LUN = LOGICAL UNIT NUMBER OF THE FILE TO READ
C      STRING = BYTE ARRAY TO SEARCH FOR
C      LEN = LENGTH OF STRING IN BYTES
C      IPOS = EXPECTED POSITION OF STRING IN ITS LINE
C
C FINDIT EXECUTES A STOP IF IT CANNOT FIND STRING BEFORE EOF
C
0001 SUBROUTINE FINDIT(LUN, STRING, LEN, IPOS)
0002 BYTE LINE(128),STRING(1)
0003 LENRD=LEN+IPOS-1
0004 IF(LENRD.GT.128) STOP 'BAD FINDIT CALL'
0005 10 READ(LUN,1000,END=100) NCH,(LINE(I),I=1,MINO(LENRD,NCH))
0006 IF(NCH.LT.LENRD) GO TO 10
0007 DO 20 I=1,LEN
0008 IF(LINE(IPOS+I-1).NE STRING(I)) GO TO 10
0009 20 CONTINUE
0010 RETURN
0011 100 WRITE(5,1010) (STRING(I),I=1,LEN)
0012 CALL EXIT
0013 1000 FORMAT(G,128A1)
0014 1010 FORMAT(1X,'FINDIT CANNOT FIND STRING ',80A1)
0015 END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CODE1</td>
<td>000430</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>$PDATA</td>
<td>000072</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>$DATA</td>
<td>000012</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>$VARS</td>
<td>000206</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>$TEMPS</td>
<td>000002</td>
<td>RW, D, CON, LCL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 000744  242

No FPP Instructions Generated
FUNCTION TERPI(X, N, XB, YB)

VIRTUAL XB(300), YB(150)

TERPI = 0.0

J = 0

NN = 2*N

DO 1 I = 2, NN, 2

J = J+1

1 CONTINUE

IF(X-XB(I).LE.0.0) GO TO 2

1 CONTINUE

I = NN

2 CONTINUE

TERPI = YB(J-1) + (YB(J) - YB(J-1))* (X-XB(I-1))/ (XB(I) - XB(I-1))

3 RETURN

END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE1</td>
<td>000312</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>DATA</td>
<td>000030</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>VARS</td>
<td>000006</td>
<td>RW, D, CON, LCL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 000350 116
FUNCTION RANGE(E,C,N)
C
.. ROUTINE TO FIND RANGE(E). THE RANGE-ENERGY RELATIONSHIP FOR
C.. THE MATERIAL IS DEFINED BY THE COEFFICIENTS, C.
C.. THE DEFINITIONS OF C ARE AS FOLLOWS.
C
.. C(1) = EL = LOWER LIMIT OF POLYNOMIAL FIT. LINEAR
C.. INTERPOLATION TO (0,0) IS USED BELOW THIS
C.. VALUE.
C
.. C(2) = RL = RANGE(EL).
C
.. C(3) = EM = ENERGY DIVIDING LO AND HI REGIONS OF
C.. POLYNOMIAL FIT OF LOG(R) TO LOG(E).
C
.. C(4) = Clo(1) = FIRST OF NLO COEFFICIENTS DESCRIBING FIT
C.. IN REGION (EL,EM).
C
.. C(4+NLO) = CHI(1) = FIRST COEFFICIENT DESCRIBING FIT ABOVE EM
C
.. N = TOTAL NUMBER OF COEFFICIENTS.
C
.. NHI = N - (NLO + 3).
C
.. FOR POLY FITS, RANGE = EXP(SUM(C(I)*LOG(E)**(I-1)))
C
PARAMETER NLO=11
DIMENSION C(1)

EL = C(1)
RL = C(2)
EM = C(3)
NHI = N-NLO-3
RANGE = 0.

C .. E .LE. 0.? IF(E .LE. 0.) RETURN
C .. E .LE. EL IF(E .GT. EL) GO TO 10
C .. E .LE. EM RANGE = E*RL/EL
RETURN
10 ELOG = ALOG(E)
IF(E .GT. EM) GO TO 30

EL .LT. E .LE. EM
I1 = 4
NP = NLO + 4
NM = NLO - 1
Y = 0.
DO 20 I=1,NM
K = NP - I
Y = (Y + C(K))*ELOG
20 CONTINUE
RANGE = EXP(Y + C(I1))
RETURN
C .. E .GT. EM
I1 = NLO + 4
NP = NHI + NLO + 4
NM = NHI - 1
GO TO 15
END
<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>#CODE1</td>
<td>000462</td>
<td>RW, I, CON, LCL</td>
</tr>
<tr>
<td>#IDATA</td>
<td>000016</td>
<td>RW, D, CON, LCL</td>
</tr>
<tr>
<td>#VARS</td>
<td>000040</td>
<td>RW, D, CON, LCL</td>
</tr>
</tbody>
</table>

Total Space Allocated = 000540  176
FUNCTION ENERGY(R, C, N)
C . ROUTINE TO FIND ENERGY(R). THE ENERGY-RANGE RELATIONSHIP FOR
C . THE MATERIAL IS DEFINED BY THE COEFFICIENTS, C.
C . THE DEFINITIONS OF C ARE AS FOLLOWS.
C . C(1) = RL = LOWER LIMIT OF POLYNOMIAL FIT LINEAR
C . INTERPOLATION TO (0,0) IS USED BELOW THIS
C . VALUE.
C . C(2) = EL = ENERGY(RL).
C . C(3) = RM = RANGE DIVIDING LO AND HI REGIONS OF
C . POLYNOMIAL FIT OF LOG(R) TO LOG(E).
C . C(4) = CLO(1) = FIRST OF NLO COEFFICIENTS DESCRIBING FIT
C . IN REGION (EL,EM).
C . C(4+NLO) = CHI(1) = FIRST COEFFICIENT DESCRIBING FIT ABOVE EM.
C . N = TOTAL NUMBER OF COEFFICIENTS.
C . NHI = N - (NLO + 3).
C .
C . FOR POLY FITS, ENERGY = EXP(SUM(C(I)*LOG(R)**(I-1)).
C
0002  PARAMETER NLO=11
0003  DIMENSION C(1)

C 0004  RL = C(1)
0005  EL = C(2)
0006  RM = C(3)
0007  NHI = N-NLO-3
0008  ENERGY = 0.
C 0009  IF(R .LE. 0.) RETURN
0010  IF(R .GT. RL) GO TO 10
C 0011  ENERGY = R*EL/RL
0012  RETURN
0013  RLOG = ALOG(R)
0014  IF(R .GT. RM) GO TO 30
C 0015  RL = .LT. R .LE. RM
0016  I1 = 4
0017  NP = NLO + 4
0018  NM = NLO - 1
0019  10 Y = 0.
0020  DO 20 I=1,NM
0021  K = NP - I
0022  Y = (Y + C(K))*RLOG
0023  20 CONTINUE
0024  ENERGY = EXP(Y + C(I1))
0025  RETURN
C 0026  IF(R .GT. RM)
0027  30 I1 = NLO + 4
0028  NP = NHI + NLO + 4
0029  NM = NHI - 1
0030  DO TO 15
0031  END
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>#CODE 1</td>
<td>000462</td>
<td>153</td>
</tr>
<tr>
<td>#DATA</td>
<td>000016</td>
<td>7</td>
</tr>
<tr>
<td>#VAR</td>
<td>000040</td>
<td>16</td>
</tr>
</tbody>
</table>

Total Space Allocated = 000540 176