LA-9412

UC-34c

Issued: August 1982

**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

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

CS

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

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

$$r_{T,D} = R_{T,D} \sqrt{\text{RAN}} \quad (1)$$

and an angle

$$\phi_{T,D} = 2\pi \times \text{RAN} \quad (2)$$

on both target and detector. This procedure gives a uniform distribution over both surfaces. These (r, ϕ) pairs are transformed to (X, Y) pairs. Symmetry is invoked to place points at (X_T, Y_T) , $(X_T, -Y_T)$, $(-X_T, Y_T)$, and $(-X_T, -Y_T)$ on the target and at (X_D, Y_D) and $(-X_D, Y_D)$ 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

$$E_p = E_n \cos^2 \theta \quad , \quad (3)$$

the range-energy relationship for protons in CH₂ and aluminum, and the average proton angle relative to the normals to the CH₂ target and aluminum filter (and ΔE 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, DTAU) 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 θ_j is

$$\Delta R(E, X, \theta_j) = \frac{zeN_A}{1000A} \frac{d\sigma}{d\Omega}(E, \theta_j) \Delta\Omega W(J) Q(E_p) \Delta X(E, \theta_j) , \quad (4)$$

where

- z = number of protons/molecule of target (z = 2 for CH₂),
- e = electronic charge (Coulombs),
- N_A = Avogadro's number,
- 1000 = number of mg/g,
- A = molecular wt (A = 14 for CH₂),
- ΔX(E, θ_j) = thickness of each target slice in mg/cm² for neutrons,
- W(θ_j) = geometrical weight, computed by GEOM,
- ΔΩ = solid angle of detector, computed by GEOM,
- dσ/dΩ (E, θ_j) = (n,p) differential scattering cross section (cm²/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 Gammel² as described in Ref. 1. The effective charge of the proton in CH₂ and aluminum as a function of E_p is derived from the evaluation of Janni³ and the information on the negative charge state in Ref. 4. Figures 1 and 2 show Q(E_p) for CH₂ 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 CH₂ and aluminum from 1-KeV to 10-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_2 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_l and E_u of the protons at the bin boundaries.

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

$$Q_{EFF} = 1 - Z_{EFF}(E_p) , \quad (5)$$

where E_p is the energy of the proton as it leaves the ΔE foil. Hence, protons of up to 0.4-MeV energy leaving the ΔE foil will contribute to this signal. The dotted lines in Fig. 3 (c) and (d) show how Q_{EFF} is computed in this case. If the protons do not penetrate the ΔE 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, $SNSI(l)$, for the l th time entry is

$$SNSI(l) = \sum_{i=1}^l R_i . \quad (6)$$

For each index l , a time of proton detection $T(l)$ 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 $SNSI(l)$ and $T(l)$.

RESPONSE FUNCTION: ANGLE INTEGRATED

The total TDR for a given neutron energy is determined by summing the responses for all possible angles of scatter θ . 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 θ . 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 TDMX is chosen to be the lesser of the average between TIMIN, the smallest possible response time, and TIMAX, 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 TDMX. The time bins earlier than TDMX are scaled to cover the region from TIMIN to TDMX and the later bins to span TDMX to TIMAX.

Fig. 3. Schematic representation involved for computing the average effective charges in a bin bounded by proton energies E_L and E_U . (a) and (b) refer to spectrum or thresholded detectors. (c) and (d) refer to penetration of protons through a ΔE 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

Fig. 4. Differential TDR functions for a thick spectrum detector (127- μ CH₂ foil).

Fig. 5. Cumulative TDR functions for a thick spectrum detector $[\int_0^t R(t)dt]$.

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:

$$Q(t_k) = \sum_i \sum_j FL(E_j, T_i) \left[\text{SENSI}(TU_k - TO_i) - \text{SENSI}(TL_k - TO_i) \right], \quad (7)$$

Fig. 6. Schematic representation of the method used for folding a finite width time bin with the cumulative response function.

where E_i, T_i are the energy and time bins for the neutron flux FL , $SENS_i$ is the folded cumulative response as a function of time after arrival of the neutron, TU_k and TL_k are the upper and lower time limits of the k th bin, and TO_i 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

TKS PINEAU-7M

Fig. 7. Comparison of time-integrated NUEXS response (dotted line) with the old FCUP method (solid line) for a thick spectrum detector.

TKT4 PINEAU-7M

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

TKT PINEAU-7M

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 $\leq 5\%$ 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 $\leq 5\%$ (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

1. M. L. Stelts, D. W. Glasgow, B. E. Wood, and A. D. Craft, "A Description of the FCUP Code Used to Compute Currents Due to Recoil Protons from CH₂ Foils," Los Alamos National Laboratory report LA-9351 (July 1982).
2. J. L. Gammel in *Fast Neutron Physics*, Part II, J. B. Marion and J. L. Fowler, Eds. (Wiley Interscience, New York, 1963), p. 2185.
3. J. F. Janni, *At. Data Nucl. Data Tables* 27, 2/3, 4/5, 147 ff (March 1982).
4. J. B. Marion and F. C. Young, *Nuclear Reaction Analysis, Graphs and Tables* (North-Holland Publishing Company, Amsterdam, 1968), p.36.

APPENDIX

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

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

16:22:23 20-May-82
/TR: BLOCKS/WR

Page 1

```

0001      C      FORTRAN IMAGE ON DM1: CLASS AT [223,202]NUEXS.FTN
          C      PROGRAM NUEXS
          C
          C      THIS PROGRAM NEEDS MODULES IN NUEXSUB.DBJ
          C
          C      THIS PROGRAM COMPUTES THE RESPONSE OF FARADAY CUPS TO
          C      PROTONS RECOILING FROM NEUTRONS INCIDENT ON CH2 FOILS.
          C
          C      ALUMINUM FILTERS BEFORE THE FARADAY CUPS ARE ALLOWED
          C      ALUMINUM DELTA-E DETECTORS ARE PROVIDED FOR
          C
          C      PROGRAM DEVELOPED BY DR ALBA CRAFT, EG&G SUNSET
          C      TRANSFERRED TO LANL JUNE 1981.
          C
          C      MODIFIED JULY 1981 FOR PDP-11 (B. E. WOOD)
          C      ADDED DELTA - E CALCULATION
          C
          C      31-OCT-81 CHANGED GEOM IN RESPONSE TO COMM. FROM A. CRAFT
          C      THIS CORRECTS ERRORS IN ORIGINAL TRANSMITTED FROM AL.
          C
          C      20 NOV 81, ADDED COMMENT CARDS IN ANTICIPATION OF DISCUSSIONS
          C      WITH AL CRAFT. CHANGED CALCULATION OF DER(EN) RE TDF AT THEOR
          C      STATION FROM INSIDE DO LOOP COMPUTING SIGNAL TO AN ARRAY
          C      CALCULATED WHEN THE ENERGIES ARE READ.
          C
          C      7-DEC-81 CORRECTED SUBROUTINE RESP TO COMPUTE CORRECT DELTA-E
          C      RESPONSE, SUBJECT TO STATED PHYSICS ASSUMPTIONS.
          C
          C      5-JAN-82 CHANGED COMPUTATION OF TIME SCALE TO MAKE TDF OF
          C      NEUTRON WITH AVERAGE ENERGY AT MAXIMUM SIGNAL MATCH TIME SCALE
          C
          C      6-JAN-81 CHANGED TO ALLOW READING MC VERSION 2A OR 2B FILES.
          C      THE DIFFERENCES IN FORMAT BETWEEN 2A AND 2B FILES ARE TAKEN
          C      INTO ACCOUNT BY PARAMETER NENCOL, THE COLUMN IN WHICH THE WORD
          C      'ENERGY' BEGINS.
          C
          C      12-JAN-81 CHANGED TO DO EITHER RELATIVISTIC OR NONREL.
          C      CALCULATION. IT IS INTENDED THAT THIS CODE USE THE SAME KINEMATICS
          C      AS WERE DONE IN MONTE CARLO.
          C
          C      26-JAN-82, MAJOR CHANGE IN STRUCTURE OF FCUP TO ALLOW
          C      COMPUTATION OF A TIME DEPENDENT RESPONSE FUNCTION FOR
          C      EACH ENERGY AND ANGLE. THIS RESPONSE IS DISTRIBUTED
          C      OVER OUTPUT BINS WHICH ARE IDENTICAL TO THE INPUT BIN
          C      STRUCTURE. THIS CHANGE ACCOUNTS PROPERLY FOR THE TIME
          C      DEPENDENCE OF THE PROTON TIME DISTRIBUTION, ACCOUNTS
          C      FOR THE LOSS IN RESPONSE AT LOW PROTON ENERGIES AS THE
          C      PROTON PICKS UP ELECTRONS, AND PRESERVES THE PROPER
          C      TIME SCALE.
          C      AT THIS TIME THE PARAMETERIZATION OF THE RANGE-ENERGY
          C      RELATIONS FOR AL AND CH2 HAVE BEEN CHANGED TO REFLECT
          C      THE RECENT CALCULATIONS OF JOSEPH F. JANNI.
          C
          C      8-FEB-82
          C      CHANGED TO COMPUTE TIME DEPENDENT RESPONSE FUNCTION, SUMMED
    
```

```

C      OVER ANGLE.
C
C      2-MAR-82
C      MODIFIED GEOM ROUTINE TO USE MONTE CARLO METHOD OF COMPUTING
C      QUANTITIES WHICH ARE FUNCTION OF ANGULAR MESH.
C
C      11-MAR-82
C      MODIFIED CODE TO USE FINER BINNING FOR TARGET AND A
C      MORE ACCURATE ALGORITHM FOR COMPUTING EFFECTIVE CHARGE
C      IN A GIVEN BIN.
C
C      13-APR-82
C      CHANGED I/O TO REMOVE DEPENDENCE ON TECO PREPROCESSING
C
0002      COMMON/A/CS(75),WT(75),CST(75),CSQ(75),AVR(75),NA
0003      COMMON/B/E(60),SENS(60),TNL,TNM,TNH,NE,TOF(60),DTAU,ZFV,RV
0004      COMMON/C/RT,RD,ZS,ZTAD,THETA,TTHICK,FTHICK,ZAD,THNX,THMN,OMEGA
0005      COMMON/D/ DTHICK,DERUN
0006      COMMON /MCCOM/ CTMC
0007      COMMON /RESP0/ NTMAX,TIMIN,TIMAX,TI(41),SENSI(41),DIFT(41)
0008      COMMON /RESP1/ SIGH,TOFP,NTARGM,ZSS,TMID
0009      COMMON /RESP2/ TIMINP,TIMAXP,TIP(41),DIFTP(41)
0010      COMMON /RESP3/ NTMX,TMINT,TMAXT,TIT(41),SNSI(41),DIFTT(41)
0011      VIRTUAL FL(150,60),TT(300),SIG(300),DT(150),QT(150),TIMEX(150)
0012      VIRTUAL QI(150),EBAR(150),QE(60),QN(60)
0013      DIMENSION F2D(60),FD(2),FSUM(2),ETEXT(3),TTEXT(3,5)
0014      LOGICAL DERUN, PRTRSP
0015      BYTE FILNAM(14),TITLE(50),STITL(80),ANS
C      NEUTRON REST MASS, SPEED OF LIGHT
0016      DATA EON/939.53/,C/29.97928/,NTARGM/20/
C
C      INITIALIZE 1D-2D FACTORS TO 1.
C
0017      DATA F2D/60*1. /
C
0018      1      FORMAT(10I5)
0019      2      FORMAT(8F10.0)
0020      3      FORMAT(/,49X,'AVERAGE(E<8)=' ,F6.3/,49X,'AVERAGE(E>8)=' ,F6.3)
0021      4      FORMAT(2X,'DETECTOR PARAMETERS'//5X,'FILTER THICKNESS=' ,F8.3,
1      ' MG/CM**2' ,T52,'*FTHICK*' ,/,21X,'=' ,F8.3,' MICRONS' /5X,
2      '/5X,'DELT E THICKNESS=' ,F8.3,' MG/CM**2' ,T52,'*DTHICK*' ,
3      '/,21X,'=' ,F8.3,' MICRONS' )
0022      5      FORMAT(5X,'APERTURE RADIUS=' ,F7.3,' CM' ,
1      T52,'*RD*' ,/5X,'TARGET TO APERTURE DISTANCE=' ,F7.3,' CM' ,
2      T52,'*ZTAD*' ,/5X,'TARGET TO DETECTOR DISTANCE=' ,F7.3,' CM' ,/
3      5X,'APERTURE TO DETECTOR DISTANCE=' ,F8.3,' CM' ,T52,'*ZAD*' )
0023      6      FORMAT(5X,'FOV DISTANCE (FROM TARGET)=' ,F9.3,' CM' ,T52,'*ZFV*' ,
1      /5X,'FOV RADIUS (DETECTOR)=' ,F8.3,' CM' ,T52,'*RFV*' ,
2      /5X,'TARGET TO DETECTOR SOLID ANGLE=' ,1PE10.3,
3      /5X,'APPROXIMATE TIME RESOLUTION=' ,OPF6.3,' NSEC' ,T52,'*DTAU*' ,
4      /5X,' NUMBER OF M.C. MESH POINT PAIRS=' ,F9.0/,/5X,
5      'CENTRAL PROTON SCATTERING ANGLE=' ,OPF7.3,' DEG' ,T52,'*THETA*' )
0024      7      FORMAT(Q,128A1)
0025      8      FORMAT(/,2X,'TARGET PARAMETERS'//5X,'TARGET THICKNESS=' ,F8.3,
1      ' MG/CM**2' ,T52,'*TTHICK*' /21X,'=' ,F8.3,' MICRONS' /5X,'TARGET
2      RADIUS=' ,F7.3,' CM' /5X,'EXPERIMENTAL STATION=' ,F9.3,' CM' ,T52,

```

```

0026      9      3 '*ZS*',/5X,'THEORETICAL STATION=',F9.3,' CM',T52,'*ZSS*')
          10      1 'FORMAT(5X,'DEFINING COLLIMATOR DISTANCE=',F9.3,' CM',T52,
          11      2 '*ZCOL*',/5X,'DEFINING COLLIMATOR RADIUS=',F8.3,' CM',T52,
          12      3 '*RCOL*',/5X,'SOURCE TO TARGET SOLID ANGLE=',1PE10.3,/5X,
          13      4 'SOURCE GEOMETRICAL FACTOR=',1PE10.3/5X,'FOV DISTANCE FROM WP=',
          14      5 'OPF9.3,' CM',T52,'*ZFOV*',/5X,
          15      6 'FOV RADIUS=',F8.3,' CM',T52,'*RFOV*',//)
0027      10      1 'FORMAT(1H1,///,2X,50A1)
0028      11      1 'FORMAT(5X,'MAXIMUM SCATTERING ANGLE=',F7.3,' DEG'/5X,'MINIMUM
          12      2 'SCATTERING ANGLE=',F7.3,' DEG'/5X,'NUMBER OF ANGLES='15,
          13      3 '(MAX=75)',//2X,'GENERAL CORRECTION FACTOR=',1PE10.3,T52,
          14      4 '*GCDR*',//2X,'TIME SHIFT=',OPF9.3,' NSEC',T52,'*TMAX*',//2X,
          15      5 'INTEGRAL FROM EMAX TO ',F3.1,' =',F10.3//2X,
          16      6 'INTEGRAL FROM ',F3.1,' TO ',F3.1,' =',F10.3//2X,'FWHM=',F9.3/)
0029      12      1 'FORMAT(/' T(I) T(I+1) CURRENT CHARGE CHGE
          13      2 'SUM',6X,'AVE EN'//)
0030      13      1 'FORMAT(3X,I3,OP2F10.3,1P4E12.3)
0031      14      1 'FORMAT(4(OPF10.3,1PE10.3))
0032      15      1 'FORMAT(/,2X,'TIME-INTEGRATED TOTALS')
0033      16      1 'FORMAT(2X,'TOF(',F3.1,')=',F10.3//2X,'TOF('F3.1')=',F10.3//
          17      2 '2X,'TOF(14.0)=',F9.3//2X,'TOF('F3.1')-TOF(14.0)=',F9.3//
          18      3 '2X,'TOF('F3.1')-TOF(14.0)=',F9.3/)
0034      17      1 'FORMAT(/, ' E(I) TOF(E) N(E) Q(E)
          18      2 'RESPONSE 1D-2D'//)
0035      18      1 'FORMAT(/, ' E(I) TOF(E) N(E) Q(E)
          19      2 'D. E. RESP 1D-2D'//)
0036      19      1 'FORMAT(1X,F10.3,F12.3,1P3E12.3,OPF9.3)
0037      25      1 'FORMAT(1X,50A1,/,80A1,2E10.3)
0038      26      1 'FORMAT(2X,80A1)
0039      30      1 'FORMAT(/,I10)
0040      31      1 'FORMAT(2X,2(4XF11.0))
0041      32      1 'FORMAT(21X,F11.0)
C
C ... FIND OUT WHICH MC VERSION, SET UP REQUIRED PARAMETERS
0042      40      1 'WRITE(5,42)
0043      42      1 'FORMAT(1X,'enter MC version 2A or 2B: '$)
0044      43      1 'READ(5,43) NMCC
0045      43      1 'FORMAT(A2)
0046      43      1 'IF((NMCC.NE.'2A').AND.(NMCC.NE.'2B')) GO TO 40
0047      43      1 'NENCOL=7
0048      43      1 'NTCOL=10
0049      43      1 'IF(NMCC.EQ.'2A') NENCOL=5
0050      43      1 'IF(NMCC.EQ.'2A') NTCOL=4
C
C ... OPEN PRINT FILE ON LUN 4 TO AVOID HOGGING LINE PRINTER
0051      43      1 'OPEN(UNIT=4,NAME='FCUPRINT.PRT',TYPE='NEW',
          44      2 '1 CARRIAGECONTROL='FORTAN',DISPOSE='PRINT')
C
C ... GET FILE NAMES AND OPEN
0052      50      1 'ASSIGN 50 TO NRET
0053      50      1 'WRITE(5,1000)
0054      1000     1 'FORMAT(1X,'enter name of PARAMETER file: '$)
0055      1000     1 'READ(5,7) NCH,(FILNAM(I),I=1,MINO(NCH,13))
0056      1000     1 'IF((NCH.LE.0).OR.(NCH.GT.13)) GO TO 50
0057      1000     1 'FILNAM(NCH+1) = 0
0058      1000     1 'OPEN(UNIT=1,NAME=FILNAM,TYPE='OLD',READONLY,ERR=2100)

```

```

0059      60  ASSIGN 60 TO NRET
0060      WRITE(5,1010)
0061      1010  FORMAT(1X,'enter name of MONTE file: '$)
0062      READ(5,7) NCH, (FILNAM(I), I=1, MINO(NCH,13))
0063      IF((NCH.LE.0).OR.(NCH.GT.13)) GO TO 60
0064      FILNAM(NCH+1) = 0
0065      OPEN(UNIT=2,NAME=FILNAM,TYPE='OLD',READONLY,ERR=2100)
C
C ... PRINT TIME-DEPENDENT RESPONSE FUNCTIONS?
C
0066      WRITE(5,1011)
0067      1011  FORMAT(1X,'do you wish time-dependent response printed? '$)
0068      READ(5,47) ANS
0069      47  FORMAT(A1)
0070      PRTRSP = ((ANS.EQ.1HY).OR.(ANS.EQ.1Hy))
C
C ... READ PARAMETER FILE
C
C ... ZSS = THEORETICAL STATION(CM)
C ... ZCOL = DEFINING COLLIMATOR STATION (CM)
C ... RCOL = RADIUS OF DEFINING COLLIMATOR (CM)
C ... ZFOV = LOCATION OF COLLIMATOR DEFINING FOV RE W.P. (CM)
C ... RFOV = RADIUS OF COLL. DEFINING FOV (CM)
C ... DTAU = APPROXIMATE TIME RESPONSE OF DETECTOR
C ... (DROPOUT = 0.3NSEC)
C ... GCOR = GENERAL CORRECTION FACTOR (DROPOUT = 1.0)
C ... TCOR = TMAX FROM MONTE CARLO (NSEC) *** NOTE CHANGE ***
C
0071      READ(1,2) ZSS,ZCOL,RCOL,ZFOV,RFOV,DTAU,GCOR,TCOR
C
0072      IF(TCOR.GT.0.0) GO TO 65
0073      WRITE(5,1015)
0074      1015  FORMAT(1X,'enter TCOR = TMAX or 0.0 to do Ebar calculation: '$)
0075      READ(5,2) TCOR
0076      65  IF(GCOR.LE.0.0) GCOR = 1.0
0077      IF(DTAU.LE.0.0) DTAU = 0.3
0078      GCOL = RCOL/ZCOL
C ... SLOPE OF PENUMBRA
0079      RCF = (RCOL+RFOV)/(ZCOL-ZFOV)
C ... GEOMETRICAL FACTOR OF DEFINING COLL.
0080      GF = GCOL*GCOL/4.0
C ... APPLY GCOR AND CONVERSION TO NSEC TIME BASE TO GF
0081      GFAC = GCOR*GF*1.0E+9
0082      OMEG = 3.1415927*GCOL*GCOL
C
C ... IF I2D .NE. 0 READ 1D TO 2D FACTORS
C ... IF ITPE .NE. 0 WRITE SIGNAL VS TIME TO OUTPUT FILE
C ... IF IPLOT .NE. 0 PLOT SIGNAL VS. TIME
C ... NOTE IPLOT .NE. 0 IMPLIES ITPE .NE. 0
0083      READ(1,1) I2D,ITPE,IPLOT
0084      IF(IPLOT.NE.0) ITPE=-1
C
C ... GET OUTPUT FILE IF REQUIRED
0085      IF(ITPE.EQ.0) GO TO 80
0086      70  WRITE(5,1020)
0087      1020  FORMAT(1X,'enter name of OUTPUT file: '$)

```

```

0088      READ(5,7) NCH, (FILNAM(I), I=1, MINO(13, NCH))
0089      IF((NCH. LE. 0). OR. (NCH. GT. 13)) GO TO 70
0090      FILNAM(NCH+1)=0
0091      OPEN(UNIT=3, NAME=FILNAM, TYPE='NEW')
C
C ... READ SOURCE TITLE, THEN POSITION TO GET ENERGY AND TIME BINS
0092      80 READ(2,7) NSTITL, (STITL(I), I=1, MINO(80, NSTITL))
0093      IF(NSTITL .GT. 80) NSTITL = 80
C
C ... GET FROM MC OUTPUT TAPE-
C ... NE = NUMBER OF ENERGY BINS
C ... E(I) = 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
C
0094      CALL FINDIT(2, 'TIME:', 5, NTCOL)
0095      CALL FINDIT(2, 'ENERGY', 6, NENCOL)
0096      ELOW=0.0
0097      CONST = ZSS/(EON*29.979)
0098      IEN=0
0099      85 IF(IEN.GE.60) STOP 'TOO MANY ENERGIES'
0100      READ(2,1022) ETEXT
0101      1022 FORMAT(<NENCOL-3>X, 3A4)
0102      IF((NMCC.EQ. '2A'). AND. (ETEXT(2).EQ. 'TOT')) GO TO 90
0103      IF((NMCC.EQ. '2B'). AND. (ETEXT(2).EQ. 'TAL ')) GO TO 90
0104      DECODE(10,1025,ETEXT) EHIGH
0105      1025 FORMAT(F10.0)
0106      IEN=IEN+1
C ... COMPUTE MEAN ENERGY IN EACH BIN
0107      E(IEN) = 0.5*(ELOW + EHIGH)
0108      ELOW=EHIGH
0109      GO TO 85
0110      90 NE=IEN
C
0111      REWIND 2
0112      IT=0
0113      TIMEX(1)=0.0
0114      100 IF(IT.GE.150) STOP 'TOO MANY TIMES'
0115      CALL FINDIT(2, 'TIME:', 5, NTCOL)
0116      BACKSPACE 2
0117      READ(2,1027) TTEXT
0118      1027 FORMAT(<NENCOL+14>X, 5(3A4, 9X))
0119      DO 105 I=1, 5
0120      IF(TTEXT(1, I).EQ. ' TO') GO TO 110
0121      IT=IT+1
0122      DECODE(10,1025,TTEXT(1, I)) TTEMP
0123      TIMEX(IT+1) = 10.*TTEMP
0124      105 DT(IT) = TIMEX(IT+1)-TIMEX(IT)
0125      GO TO 100
0126      110 NT=IT
C
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.
C

```



```

0127      REWIND 2
0128      NFULL = NT/5
0129      IF(NFULL .EQ. 0) GO TO 120
0130      DO 115 IPAGE=1,NFULL
0131      CALL FINDIT(2, 'TIME', 5, NTCOL)
0132      CALL FINDIT(2, 'ENERGY', 6, NENCOL)
0133      ITBASE = 1+5*(IPAGE-1)
0134      DO 115 IEN=1,NE
0135      READ(2,1033) (FL(IT, IEN), IT=ITBASE, ITBASE+4)
0136      115 CONTINUE
C
0137      120 NTLEFT = NT-5*NFULL
0138      IF(NTLEFT .EQ. 0) GO TO 150
0139      CALL FINDIT(2, 'ENERGY', 6, NENCOL)
0140      ITBASE = 1+5*NFULL
0141      DO 130 IEN=1,NE
0142      READ(2,1033) (FL(IT, IEN), IT=ITBASE, ITBASE+NTLEFT-1)
0143      130 CONTINUE
0144      1033 FORMAT(<NENCOL>X, 5(10X, F11.0))
C
C ... POSITION MC FILE TO GET UNATTENUATED TOTAL COUNTS
0145      150 CALL FINDIT(2, 'RAD', 3, 2)
0146      IGTPOS = 22+21*NTLEFT + (NENCOL-5)
0147      CALL FINDIT(2, 'TOTAL', 5, IGTPOS)
C
C ... READ FROM MC OUTPUT TAPE QE(I)
C ... QE(I) = UNATTENUATED NUMBER OF NEUTRONS INTO 4PI IN ITH ENERGY BIN
C ... TOTALLED FOR ALL TIMES.
C
0148      CALL FINDIT(2, 'ENERGY', 6, NENCOL)
0149      NSKIP = IGTPOS-7
0150      DO 180 IEN=1,NE
0151      READ(2,1035) QE(IEN)
0152      180 CONTINUE
0153      1035 FORMAT(<NSKIP>X, F11.0)
0154      CLOSE(UNIT=2)
C
C
C
C
C ... INITIALIZE FD ACCUMULATORS AND READ 2D FACTORS IF REQD
0155      FD(1) = 0.0
0156      FD(2) = 0.0
0157      IF(I2D .EQ. 0) GO TO 290
C
C ... READ IN ENERGY DEPENDENT 1D TO 2D CORRECTION FACTORS
C
0158      READ(1,2) (F2D(I), I=1, NE)
C
C ... COMPUTE AVERAGE 1D TO 2D CORR FACTORS FOR FISSION, FUSION.
C
0159      J = 1
0160      FD(1) = QE(1)*F2D(1)
0161      FD(2) = 0.0
0162      FSUM(1) = QE(1)
0163      FSUM(2) = 0.0

```

```

0164      DO 220 K=2,NE
0165      IF(F2D(K) .LE. 0.0) F2D(K) = F2D(K-1)
0166      IF(E(K) .GT. 8.0) J = 2
0167      FSUM(J) = FSUM(J)+QE(K)
0168      FD(J) = FD(J)+F2D(K)*QE(K)
0169      220 CONTINUE
0170      FD(1) = FD(1)/FSUM(1)
0171      FD(2) = FD(2)/FSUM(2)
C
C ... ADJUST FL(T,E) BY ENERGY DEPENDENT 1D TO 2D F2D(E)
C
0172      DO 230 I=1,NT
0173      DO 230 J=1,NE
0174      FL(I,J) = FL(I,J)*F2D(J)
0175      230 CONTINUE
C
C ... PUT DATE INTO FCUP RUN TITLE
0176      290 CALL DATE(TITLE(32))
C
C ... PUT TIME INTO FCUP RUN TITLE AND READ IT
0177      300 TITLE(41)="40
0178      TITLE(42)="40
0179      CALL TIME(TITLE(43))
0180      READ(1,7,END=999) NTITLE, (TITLE(I), I=1,MINO(30,NTITLE))
C ... FILL UNUSED TITLE SPACE WITH BLANKS
0181      DO 301 I=NTITLE+1,31
0182      301 TITLE(I)="40
C
C ... ZS = EXPERIMENTAL TARGET STATION (CM)
C ... ZTAD = TARGET TO APERTURE (CM)
C ... ZAD = APERTURE TO DETECTOR (CM)
C ... RD = APERTURE RADIUS (CM)
C ... ZFV = DISTANCE FROM TARGET OF FOV APERTURE (CM)
C ... RFV = RADIUS (CM) OF FOV APERTURE
C ... THETA = DETECTOR ANGLE (DEGREES) (DROPOUT = 30.)
C
0183      READ(1,2) ZS,ZTAD,ZAD,RD,ZFV,RFV,THETA
C
C ... TTHICK = TARGET THICKNESS (MG/CM**2)
C ... FTHICK = FILTER THICKNESS (MG/CM**2) (DROPOUT = 0.)
C ... DTHICK = DELT E THICKNESS (MG/CM**2) (DROPOUT = 0.)
C
0184      READ(1,2) TTHICK,FTHICK,DTHICK
0185      IF(RD .LE. 0.0) RD = 1.27
0186      IF(THETA .LE. 0.0) THETA = 30.0
0187      IF(FTHICK .LT. 0.0) FTHICK = 0.0
0188      IF(DTHICK .LT. 0.0) DTHICK = 0.0
C
C ... COMPUTE THICKNESSES IN MICRONS. RHO(CH2) = 0.92, (AL) = 2.70
C
0189      TTHICKM = TTHICK/0.092
0190      FTHICKM = FTHICK/0.27
0191      DTHICKM = DTHICK/0.27
0192      DERUN = (DTHICKM.NE.0.)
C ... COMPUTE RADIUS OF PENUMBRA AT TARGET
0193      RT = (ZS-ZFOV)*RCF-RFOV

```

```

C
C ... INITIALIZE OUTPUT ARRAYS
C
0194      QTOT = 0.0
0195      DO 310 I=1,300
0196      310  SIG(I) = 0.0
0197      DO 311 I=1,150
0198      QT(I) = 0.0
0199      EBAR(I) = 0.0
0200      311  GI(I) = 0.0
0201      DO 312 I=1,60
0202      SENS(I) = 0.
0203      TOF(I) = 0.
0204      QE(I) = 0.0
0205      312  QN(I) = 0.0
C
C ... SET UP ANGULAR ARRAY AND PERTINENT PARAMETERS FOR RESPT
C
0206      CALL GEOM
C
C ... COMPUTE FOR ALL NEUTRON ENERGIES .....
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
C ... REFERENCE: J.L. GAMMEL IN 'FAST NEUTRON PHYSICS', PART II, P.2185
C ... ED. BY MARION AND FOWLER.
C
0208      EN = E(J)
0209      FA = (0.09415+0.0001306*EN)*EN-1.86
0210      FB = 0.4223+0.13*EN
0211      FC = 1.206*EN
0212      SIGH = (3.0/(FC+FA*FA))+(1.0/(FC+FB*FB))
C
C ... GET TIME DEPENDENT RESPONSE FOR EN, SUMMED OVER ALL ANGLES
C
0213      CALL RESPT(EN,IPOS)
C ... CHECK FOR ZERO RESPONSE
0214      IF(IPOS.EQ. 0) GO TO 320
0215      SENS(J) = SENSI(NTMAX)
0216      TOF(J) = TOFP
C
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 = ',E12.3,
1  2X,I3,' POINTS',///,9X,'TIME INTERVAL',3X,
2  'SENS(COUL/NSEC/NEUT)', ' INTEGRAL TO T2',/)
0220      DO 7313 I=1,NTMAX-1
0221      T1 = TI(I)
0222      T2 = TI(I+1)
0223      XX = (SENSI(I+1) - SENSI(I))/(T2 - T1)

```

```

0224      7313      WRITE(4,7001) T1,T2,XX,SENSI(I+1)
0225      7001      FORMAT(1X,1PE12.3, ' TO',3E12.3)
C
C ... FOLD RESPONSE FROM ALL INPUT TIME BINS .....
C ... NOTE: POSSIBLE TROUBLE IF ZSS .GT. ZS
C ... TL = LOWER TIME LIMIT FOR INPUT TIME BIN
C ... TU = UPPER TIME LIMIT FOR INPUT TIME BIN
C
0226      313      TU = 0.
0227              DO 316 KK=1,NT
0228              TL = TU
0229              TU = TIMEX(KK+1)
0230              XNEUT = GFAC*FL(KK,J)
0231              IF(XNEUT .EQ. 0.) GO TO 316
0232              TO = 0.5*(TU + TL)
C
C ... BROADEN TIME BASE OF INTEGRAL RESPONSE FUNCTION TO ACCOUNT
C ... FOR THE WIDTH OF THE INPUT TIME BIN. BROADENING IS RELATIVE
C ... TO THE CENTER OF THE TIME BIN AND MAXIMUM OF RESP(T).
C
0233              DELTP = TO - TL
0234              TIMINP = TIMIN - DELTP
0235              TIMAXP = TIMAX + DELTP
C
C ... FIND INDEX OF AVERAGE TIME OF RESPONSE
C
0236              TEST = 0.5*SENS(J)
0237              DO 3000 IM=1,NTMAX
0238              IF(SENSI(IM) .GE. TEST) GO TO 3001
0239      3000      CONTINUE
0240      3001      IL = IM - 1
0241              IU = NTMAX - IM
C ... ADJUST LOWER HALF TIME SCALE
0242              DTTL = DELTP/IL
0243              DTT = 0.
0244              DO 3002 I=IL,1,-1
0245              DTT = DTT + DTTL
0246              TIP(I) = TI(I) - DTT
0247      3002      CONTINUE
C ... ADJUST UPPER HALF TIME SCALE
0248              DTTU = DELTP/IU
0249              DTT = -DTTU
0250              DO 3003 I=IM,NTMAX
0251              DTT = DTT + DTTU
0252              TIP(I) = TI(I) + DTT
0253      3003      CONTINUE
0254              DIFTP(1) = 1.
0255              DO 3004 I=2,NTMAX
0256              DIFTP(I) = TIP(I) - TIP(I-1)
0257      3004      CONTINUE
C
C ... INTO ALL OUTPUT TIME BINS.
C ... TM = LOWER LIMIT OF OUTPUT TIME BIN RE TO.
C ... TP = UPPER LIMIT OF OUTPUT TIME BIN RE TO.
C
0258              TP = -TO

```

```

0259      RINTP = PRES P(TP)
0260      DO 314 KKK=1,NT
0261      TM = TP
0262      RINTM = RINTP
0263      TP = TIMEX(KKK+1) - TO
C ... IS TP BEFORE FINITE RESPONSE TIME?
0264      IF(TP .LE. TIMINP) GO TO 314
C ... IS TM AFTER FINITE RESPONSE TIME?
0265      IF(TM .GE. TIMAXP) GO TO 316
C
C ... COMPUTE CONTRIBUTION IN KKK TH TIME BIN
C
0266      RINTP = PRES P(TP)
0267      CONT = XNEUT*(RINTP - RINTM)
C ... ACCUMULATE CHARGE IN KKK
0268      QT(KKK) = QT(KKK) + CONT
C ... ACCUMULATE SIGNAL WEIGHTED AVERAGE NEUTRON ENERGY IN KKK
0269      EBAR(KKK) = EBAR(KKK) + CONT*EN
0270      314 CONTINUE
C ... END OF LOOP ON OUTPUT TIME BINS
0271      316 CONTINUE
C ... END OF LOOP ON INPUT TIME BINS
0272      320 CONTINUE
C
C ... END OF LOOP ON NEUTRON ENERGY
C
C ... NORMALIZE EBAR, COMPUTE TIME INTEGRAL OF CHARGE, COMPUTE CURRENT
C
0273      K = 0
0274      DO 322 I=1,NT
0275      IF(EBAR(I) .LE. 0.0) EBAR(I) = -1.0
0276      IF(EBAR(I) .GT. 0.0) EBAR(I) = EBAR(I)/QT(I)
0277      QTOT = QTOT + QT(I)
0278      QI(I) = QTOT
0279      K = K+2
C ... COMPUTE CURRENT
0280      SIG(K-1) = QT(I)/DT(I)
0281      SIG(K) = SIG(K-1)
0282      322 CONTINUE
C
C ... ACCUMULATE CHARGE AND NUMBER OF NEUTRONS IN EACH ENERGY BIN
C
0283      DO 326 J=1,NE
0284      DO 324 I=1,NT
0285      XNEUT = FL(I,J)
C ... CHARGE IN EACH ENERGY BIN
0286      QE(J) = QE(J) + SENS(J)*XNEUT*GFAC
C ... NUMBER OF NEUTRONS IN EACH ENERGY BIN
0287      QN(J) = QN(J) + XNEUT
0288      324 CONTINUE
0289      326 CONTINUE
C
C ... COMPUTE TOTAL TOF FOR EACH ENERGY BIN
C
C ... TOF = TIME-OF-FLIGHT TO ZSS + <T-O-F FROM ZSS TO DETECTOR>
C

```

```

0290      DO 330 I=1,NE
0291      TERM = 0.
0292      IF(SENS(I) .GT. 0.) TERM = TOF(I)/SENS(I)
0293      TOF(I) = TOFF(E(I),ZSS,EON) + TERM

0294      330      CONTINUE
C
C ... DEFINE INTEGRATION LIMITS- NOMINALLY 1-, 8-, 14-MEV.
C
0295      ENL = 1.
0296      IF(ENL .LT. E(1)) ENL = E(1)
0297      ENM = 8.
0298      IF(ENM .LT. E(1)) ENM = E(1)
0299      ENH = 14.
C
C ... FIND TIMES-OF-FLIGHT FOR THESE ENERGIES
C
0300      CALL TINT(ENL,TNL,E,TOF,NE)
0301      CALL TINT(ENM,TNM,E,TOF,NE)
0302      CALL TINT(ENH,TNH,E,TOF,NE)
0303      DT1 = TNL - TNM
0304      DT2 = TNM - TNH
C
C ... IF TCOR = 0, COMPUTE TORP AS TOF(EBAR(MAX SIGNAL))
C
0305      TCORP = TCOR
0306      IF(TCOR .NE. 0.) GO TO 356
0307      WRITE(4,1400)
0308      1400      FORMAT(//1X,'***** NOTE: Time shift was calc'd via Ebar',/)
C
C ... FIND TIME BIN WITH MAXIMUM SIGNAL
C
0309      IMX = 1
0310      K = 0
0311      TEST = 0.
0312      DO 351 I=1,NT
0313      K = K+2
0314      IF(SIG(K) .LT. TEST) GO TO 351
0315      TEST = SIG(K)
0316      IMX = I
0317      351      CONTINUE
C
C ... FIND AVERAGE TOF FOR EBAR(IMX)
C
0318      EBARP = EBAR(IMX)
0319      DO 352 I=1,NE-1
0320      IF(EBARP .LE. E(I)) GO TO 353
0321      352      CONTINUE
0322      WRITE(5,1500)
0323      WRITE(4,1500)
0324      1500      FORMAT(//1X,'** WARNING - FELL THROUGH FINDING EBAR BIN')
0325      353      SLP = (TOF(I)-TOF(I-1))/(E(I)-E(I-1))
0326      TCORP = TOF(I-1) + SLP*(EBARP-E(I-1))
C
C ... ADJUST TIME SCALE SUCH THAT MAXIMUM SIGNAL HAS TOF OF
C ... AVERAGE NEUTRON ENERGY IN THAT TIME BIN

```

```

C
0327      TCORP = TIMEX(IMX) - 0.5*DT(IMX) - TCORP
C
C      SHIFT TO PROPER TIME-OF-FLIGHT SCALE
C
0328      356      TT(1) = -TCORP
0329              NTS = 2*NT
0330              J = 0
0331              DO 360 I=2,NTS,2
0332                  J = J+1
0333                  TT(I) = TT(I-1) + DT(J)
0334                  TT(I+1) = TT(I)
0335      360      CONTINUE
0336              IF(TT(NTS) .LT. 100000.) GO TO 361
0337              TT(NTS) = 99999.99
0338              TT(NTS+1) = 99999.99
0339      361      CONTINUE
C
C      IF NOT A DELTA-E, COMPUTE FWHM OF 14 MEV PEAK
C
0340              FWHM = 0.0
0341              IF(DERUN) GO TO 365
0342              JMX = 2
0343              FM = 0.
0344              DO 362 I=2,NTS,2
0345                  IF(TT(I) .GE. TNM) GO TO 363
0346                  IF(SIG(I) .LT. FM) GO TO 362
0347                  JMX = I
0348                  FM = SIG(I)
0349      362      CONTINUE
0350      363      CONTINUE
0351              FHM = FM/2.0
0352              DO 20 I=1,JMX,2
0353                  IF(SIG(I) .GT. FHM) GO TO 21
0354      20      CONTINUE
0355      21      TA = (TT(I)+TT(I-2))/2.0
0356              TB = (TT(I)+TT(I+2))/2.0
0357              TFA = TA+(TB-TA)*(FHM-SIG(I-2))/(SIG(I)-SIG(I-2))
0358              DO 22 I=JMX,NTS,2
0359                  IF(SIG(I) .GE. FHM) GO TO 22
0360                  IF(SIG(I+2) .LT. FHM) GO TO 23
0361      22      CONTINUE
0362      23      J = I-1
0363              TA = 0.5*(TT(J)+TT(J-2))
0364              TB = 0.5*(TT(J)+TT(J+2))
0365              FWHM = (TA+(TB-TA)*(SIG(I-2)-FHM)/(SIG(I-2)-SIG(I)))-TFA
C
C      COMPUTE CHARGE INTEGRALS E = (0,8) AND (8,INFINITY)
C      NOTE TERP1 ARGUMENTS TT AND QI ARE VIRTUAL ARRAYS
0366      365      QSUM1 = TERP1(TNM,NT,TT,QI)
0367              QSUM2 = TERP1(TNL,NT,TT,QI)-QSUM1
0368              ZTDO = ZTAD+ZAD
0369              WRITE(4,10) TITLE
0370              WRITE(4,33)
0371      33      FORMAT(' VERSION NUEXS- 11-MAR-82, TIME-DEPENDENT RESPONSE. ')
0372              WRITE(4,26) (STITL(I),I=1,NSTITL)

```

```

C
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
0373      WRITE(4,8) TTHICK,TTHIKM,RT,ZS,ZSS
0374      WRITE(4,9) ZCOL,RCOL,OMEG,GF,ZFOV,RFOV
C
C ... FTHICK = AL FILTER THICKNESS (MG/CM**2)
C ... FTHIKM = AL FILTER THICKNESS (MICRONS)
C ... RD     = DETECTOR APERTURE RADIUS (CM)
C ... ZTAO   = TARGET TO APERTURE DISTANCE (CM)
C ... ZTDO   = TARGET TO DETECTOR DISTANCE (CM)
C ... ZAD    = APERTURE TO DETECTOR (CM)
C ... ZFV    = TARGET TO TARGET FOV APERTURE (CM)
C ... RFV    = RADIUS OF TARGET FOV APERTURE (CM)
C ... OMEGA   = TARGET TO DETECTOR SOLID ANGLE
C ... CTMC   = NUMBER OF M.C. MESH POINT PAIRS USED IN GEOM.
C ... DTAU   = APPROX. TIME RESPONSE OF DETECTOR (NS)
C ... THETA  = CENTRAL PROTON SCATTERING ANGLE
C
0375      WRITE(4,4) FTHICK,FTHIKM,DTHICK,DTHIKM
0376      WRITE(4,5) RD,ZTAO,ZTDO,ZAD
0377      WRITE(4,6) ZFV,RFV,OMEGA,DTAU,CTMC,THETA
C
C ... THMX   = MAXIMUM SCATTERING ANGLE
C ... THMN   = MINIMUM SCATTERING ANGLE
C ... NA     = NUMBER OF ANGLES
C ... GCOR   = GENERAL CORRECTION FACTOR
C ... TCOR   = TIME SHIFT (NS)
C ... ENM    = ENERGY BOUNDRY BETWEEN FUSION AND FISSION
C ... QSUM1  = INTEGRAL CHARGE FOR FUSION
C ... ENL    = LOW BOUND FOR FISSION ENERGY
C ... QSUM2  = INTEGRAL CHARGE FOR FISSION
C ... FWHM   = WIDTH OF 14-MEV PEAK IN NSEC.
C
0378      WRITE(4,11) THMX,THMN,NA,GCOR,TCOR,ENM,
1 QSUM1,ENM,ENL,QSUM2,FWHM
C
C ... ENL    = ENERGY OF LOWER ENERGY BOUND FOR INTEGRALS
C ... TNL    = TIME OF FLIGHT FOR LOWEST ENERGY.
C ... ENM    = ENERGY OF UPPER BOUND FOR FISSION INTEGRAL.
C ...       (LOWER BOUND FOR FUSION INTEGRAL.)
C ... TNM    = TIME OF FLIGHT FOR THAT ENERGY.
C ... TNH    = TIME OF FLIGHT FOR 14 MEV NEUTRON.
C ... DT1    = TNL - TNM
C ... DT2    = TNM - TNH
C

```



```

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
C ... TT = TIME BIN BOUNDARY (NSEC)
C ... SIG = SIGNAL IN TIME BIN (AMPS)
C ... QT = CHARGE IN TIME BIN
C ... QI = 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), QI(I), EBAR(I)
0388      J = J+2
0389      370 CONTINUE
0390      IF(N.EQ.NT) GO TO 380
0391      WRITE(4,10) TITLE
0392      WRITE(4,26) (STITL(I), I=1, NSTITL)
0393      WRITE(4,12)
0394      DO 375 I=51,NT
0395      WRITE(4,13) I, TT(J), TT(J+1), SIG(J), QT(I), QI(I), EBAR(I)
0396      J = J+2
0397      375 CONTINUE
0398      380 CONTINUE
C
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(3,1) NTS
0402      WRITE(3,14) (TT(I), SIG(I), I=1, NTS)
0403      WRITE(3,1) NE
0404      WRITE(3,14) (E(I), SENS(I), I=1, NE)
0405      390 CONTINUE
C
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
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 ... QE = 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), QE(I), SENS(I), F2D(I)
0413      400 CONTINUE
C

```

```

      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

Name	Size	Attributes
#CODE1	016136 3631	RW, I, CON, LCL
#PDATA	005522 1449	RW, D, CON, LCL
#IDATA	000160 56	RW, D, CON, LCL
#VARS	001414 390	RW, D, CON, LCL
#TEMPS	000024 10	RW, D, CON, LCL
A	002736 751	RW, D, OVR, GBL
B	001352 373	RW, D, OVR, GBL
C	000054 22	RW, D, OVR, GBL
D	000006 3	RW, D, OVR, GBL
MCCOM	000004 2	RW, D, OVR, GBL
RESP0	000766 251	RW, D, OVR, GBL
RESP1	000022 9	RW, D, OVR, GBL
RESP2	000520 168	RW, D, OVR, GBL
RESP3	000766 251	RW, D, OVR, GBL

Total Space Allocated = 034614 7366

Total Virtual Array Storage = 659

```

0001      SUBROUTINE RESPT(EN, IPOS)
C
C ... COMPUTES TIME DEPENDENT RESPONSE OF NEUTRONS OF ENERGY, EN.
C ... IPOS = 0 IMPLIES ZERO RESPONSE.
C
C ... ROUTINE GEOM MUST BE CALLED BEFORE CALLING RESPT TO SET UP
C ... ANGULAR ARRAYS AND COMPUTE OMEGA.
C
C ... CALLED BY - MAIN
C
C ... CALLS      - ARESPT
C                DELRES
C
C ... 8-FEB-82
C
C ... /RESPO/
C ... NTMAX = NUMBER OF ENTRIES IN CUMULATIVE RESPONSE.
C ... TIMIN = MINIMUM TIME OF RESPONSE.
C ... TIMAX = MAXIMUM TIME OF RESPONSE.
C ... TI    = TIME ENTRIES RELATIVE TO NEUTRONS AT ZSS.
C ... SENSI = CUMULATIVE RESPONSE AT TI
C ... DIFT  = TI(I) - TI(I-1)
C
C ... /RESP1/ SEE SUBROUTINE ARESPT
C ... NTARGM = MAXIMUM NUMBER OF DIVISIONS OF CH2 FOIL.
C
C ... /RESP3/ SAME AS /RESPO/ EXCEPT INTERMEDIATE STORAGE
C ...        WHILE SUMMING OVER ANGLE.
C
0002      COMMON/A/CS(75),WT(75),CST(75),CSQ(75),AVR(75),NA
0003      COMMON/B/E(60),SENS(60),TNL,TNM,TNH,NE,TOF(60),DTAU,ZFV,RFV
0004      COMMON/D/ DTHICK,DERUN
0005      COMMON /RESPO/ NTMAX,TIMIN,TIMAX,TI(41),SENSI(41),DIFT(41)
0006      COMMON /RESP1/ SIGH,TOFP,NTARGM,ZSS,TMID
0007      COMMON /RESP3/ NTMX,TMINT,TMAXT,TIT(41),SNSI(41),DIFTT(41)
0008      DIMENSION TRP(41)
0009      LOGICAL DERUN
C
C ... SEE IF EN GREAT ENOUGH TO GIVE FINITE RESPONSE
C
0010      IF(EN .GT. 0.006) GO TO 1
0011      IPOS = 0
0012      RETURN
C
C ... FIND TIMIN
C
0013      1 IF(DERUN) GO TO 10
C
C ... FOR F.C. OR THRESH
C
0014      CALL ARESPT(0,EN,1,IPOS,2)
0015      NDMN = 1
0016      IF(IPOS .EQ. 0) RETURN
0017      GO TO 12
C
C ... FOR DELTA-E

```

```

      C
0018      10      DO 11 I=1,NA
0019              CALL ARESPT(0,EN,I,IPOS,2)
0020              NDMN = I
0021              IF(IPOS.NE. 0) GO TO 12
0022      11      CONTINUE
0023              RETURN
0024      12      TIMIN = TMINT
      C
      C ... FIND TIMAX, TDMX
      C
0025              IF(DERUN) GO TO 14
      C
      C ... FOR F.C. OR THRESH
      C
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
      C ... FOR DELTA-E
      C
0032      14      DO 15 I=NA,1,-1
0033              CALL ARESPT(0,EN,I,IPOS,2)
0034              NDMX = I
0035              IF(IPOS.NE. 0) GO TO 16
0036      15      CONTINUE
0037              RETURN
0038      16      TIMAX = TMAXT
0039              TDMX = TMID
      C
      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).
      C
0040              TEST = 0.5*(TIMIN+TIMAX)
0041              IF(TDMX.GT. TEST) TDMX = TEST
      C
      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 = NDMX/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(NTMX.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      NPR = NTMX - 1
0054      NFC = (2*NTARGM)/NPR
0055      IF(NFC .EQ. 1) GO TO 20
0056      NTMAX = NFC*NPR + 1
0057      NTT = NTMX + 1
0058      NT = NTMAX + 1
0059 17    NTT = NTT - 1
0060      DO 18 I=1,NFC
0061        NT = NT - 1
0062        DIFTT(NT) = DIFTT(NTT)/NFC
0063 18    CONTINUE
0064      IF(NTT .GT. 1) GO TO 17
0065      TIT(1) = TMINT
0066      DO 19 I=2,NTMAX
0067        TIT(I) = TIT(I-1) + DIFTT(I)
0068 19    CONTINUE
C
C    SCALE TI FROM 1 TO NM=NTMAX/2 TO COVER INTERVAL (TIMIN,TDMX)
C
0069 20    NM = NTMAX/2
0070      FACT = (TDMX-TIMIN)/(TIT(NM)-TMINT)
0071      TI(1) = TIMIN
0072      DIFT(1) = 1.
0073      DO 21 I=2,NM
0074        DIFT(I) = FACT*DIFTT(I)
0075        TI(I) = TI(I-1) + DIFT(I)
0076 21    CONTINUE
C
C    SCALE TI FROM NM+1 TO NTMAX TO COVER INTERVAL (TDMX,TIMAX)
C
0077      FACT = (TIMAX-TDMX)/(TIT(NM)-TIT(NM))
0078      DO 22 I=NM+1,NTMAX
0079        DIFT(I) = FACT*DIFTT(I)
0080        TI(I) = TI(I-1) + DIFT(I)
0081 22    CONTINUE
C
C    FIND RESPONSE FOR EACH ANGLE AND ACCUMULATE IN TRP
C
0082      NTGBNS = 2*NTARGM
0083      DO 30 I=1,NTMAX
0084        TRP(I) = 0.
0085 30    CONTINUE
0086      TOFPP = 0.
0087      DO 40 K=NDMN,NDMX
0088        CALL ARESPT(1,EN,K,IPOS,NTGBNS)
0089        IF(IPOS .EQ. 0) GO TO 40
0090        TOFPP = TOFPP + TOFP
0091      TP = TI(1)
0092      RP = 0.
0093      DO 35 J=2,NTMAX
0094        TM = TP
0095        RM = RP

```

```

0096          TP = TI(J)
0097          RP = DELRES(TP)
0098          TRP(J-1) = TRP(J-1) + (RP - RM)
0099      35      CONTINUE
0100      40      CONTINUE
C
C ... FORM CUMULATIVE RESPONSE
C
0101          TOT = 0.
0102          SENSI(1) = 0.
0103          DO 50 I=2,NTMAX
0104              TOT = TOT + TRP(I-1)
0105          SENSI(I) = TOT
0106      50      CONTINUE
0107          TOFP = TOFPP
0108          IPOS = 1
0109          RETURN
0110          END
  
```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	002212 581	RW, I, CON, LCL
\$PDATA	000024 10	RW, D, CON, LCL
\$IDATA	000074 30	RW, D, CON, LCL
\$VARS	000342 113	RW, D, CON, LCL
\$TEMPS	000006 3	RW, D, CON, LCL
A	002736 751	RW, D, OVR, GBL
B	001352 373	RW, D, OVR, GBL
D	000006 3	RW, D, OVR, GBL
RESP0	000766 251	RW, D, OVR, GBL
RESP1	000022 9	RW, D, OVR, GBL
RESP3	000766 251	RW, D, OVR, GBL

Total Space Allocated = 011216 2375

```

0001      SUBROUTINE ARESPT(ICNTRL,EN,J,IPOS,NTRT)
C
C ... COMPUTES TIME DEPENDENT RESPONSE OF NEUTRONS OF ENERGY, EN,
C ... AT JTH ANGULAR MESH POINT. IPOS = 0 IMPLIES ZERO RESPONSE.
C ... IF(ICNTRL.EQ. 0) ONLY TIME BINS ARE COMPUTED.
C ... NTRT IS MAXIMUM NUMBER OF CH2 FOIL DIVISIONS.
C
C ... GEOMETRY IS ASSUMED TO BE: TARGET PERPENDICULAR TO
C ... NEUTRON BEAM, CIRCULAR APERTURES.
C
C ... CALLED BY - RESPT
C
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
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 ... DIFTT  = TIT(I) - TIT(I-1).
C
0002      COMMON/A/CS(75),WT(75),CST(75),CSQ(75),AVR(75),NA
0003      COMMON/B/E(60),SENS(60),TNL,TNM,TNH,NE,TOF(60),DTAU,ZFV,RFV
0004      COMMON/C/RT,RD,ZS,ZTAD,THETA,TTHICK,FTHICK,ZAD,THMX,THMN,OMEGA
0005      COMMON/D/ DTHICK,DERUN
0006      COMMON /RESP1/ SIGH,TOFP,NTARGM,ZSS,TMID
0007      COMMON /RESP3/ NTMX,TMINT,TMAXT,TIT(41),SNSI(41),DIFTT(41)
0008      DIMENSION EP(41),GEFF(41),PRES(41)
0009      DIMENSION C(25),D(25),CA(25),DA(25),ZPOLY(7),ZAL(7)
0010      LOGICAL DERUN
0011      DATA C/ 0.001, 0.00291, 0.10,
1      8.2677984E-1, 1.5249348E+0, -2.0432448E-1, -6.9163345E-2,
2      1.5429533E-1, 1.1416341E-1, 3.7130889E-2, 6.8796021E-3,
3      7.5200538E-4, 4.5281766E-5, 1.1606486E-6,
4      7.4128067E-1, 1.6310188E+0, 4.9908146E-2, -1.6055908E-2,
5      1.6253492E-2, -4.3543098E-3, -2.1013061E-3, 1.0201402E-3,
6      3.7088954E-5, -6.9876078E-5, 8.4383119E-6/
0012      DATA D/ 0.00291, 0.001, 0.10173,
1      7.0718777E-1, 1.0397812E+0, -1.4575658E+0, -6.9523209E-1,
2      8.3148688E-1, 9.7903919E-1, 4.3677825E-1, 1.0537574E-1,

```

```

3 1.4558905E-2, 1.0887752E-3, 3.4321158E-5,
4 -4.6247172E-1, 6.3773197E-1, -2.4754656E-2, 1.0563388E-2,
5 -3.7713239E-3, 5.6535547E-4, 2.7127993E-5, -1.8015751E-5,
6 1.6906509E-6, -8.4548892E-9, -3.7291512E-9/
0013 DATA CA/ 0.001, 0.00627, 0.10,
1 1.2281270E+0, 9.0592396E-1, -4.3483263E-1, -5.4568924E-2,
2 2.0528728E-1, 1.4969474E-1, 5.2055866E-2, 1.0596559E-2,
3 1.2829266E-3, 8.5492058E-5, 2.4139579E-6,
4 1.3682901E+0, 1.4668549E+0, 1.0515825E-1, -1.1806433E-2,
5 -6.7958888E-3, 2.3810312E-3, 1.0640459E-3, -5.5370503E-4,
6 -2.5508705E-5, 4.2737487E-5, -5.3098165E-6/
0014 DATA DA/ 0.00627, 0.001, 0.24051,
1 -5.8794409E-1, 1.9262463E+0, 9.4625008E-1, 1.1669281E-1,
2 -4.2174953E-1, -3.4479511E-1, -1.4193271E-1, -3.6508426E-2,
3 -5.8908788E-3, -5.4044055E-4, -2.1328990E-5,
4 -1.0082754E+0, 8.0021322E-1, -5.3594884E-2, 7.3744599E-3,
5 -2.8793879E-3, 1.4175350E-3, -3.1559233E-4, 2.1258860E-5,
6 2.7488445E-6, -5.1122214E-7, 2.2528020E-8/
0015 DATA ZPOLY/0.7235886, 1.891709, 1.753375, 0.662257, 7.159398E-02,
1 -5.914641E-3, -1.190180E-3/
0016 DATA ZAL/1.761918, 4.868560, 5.106298, 2.591938, 0.6625297,
1 8.307397E-2, 3.023415E-3/
0017 DATA PI/3.1415927, EOP/938.256, EON/939.55/

```

```

C
C ... THE ARRAYS C,D,CA,DA CONTAIN THE COEFFICIENTS WHICH FIT
C ... THE RANGE-ENERGY TABLES OF JOSEPH F. JANNI,
C ... (PRIVATE COMMUNICATION JAN'82, TO BE PUBLISHED NDT)
C ... A 10TH ORDER POLYNOMIAL FIT IN LOG(R) VS LOG(E) WAS USED FOR
C ... THE REGIONS (0.001-,0.100-MEV) AND (0.100-,40-MEV).
C ... A LINEAR INTERPOLATION TO (0,0) IS USED BELOW 0.001-MEV.
C ... THE COEFFICIENTS ARE DEFINED AS FOLLOWS.
C ... COEF(1) = (ENERGY OR RANGE) AT 0.001
C ... COEF(2) = (RANGE OR ENERGY) AT 0.001
C ... COEF(3) = (ENERGY OR RANGE) AT 0.100
C ... REMAINING 22 COEFFICIENTS ARE FOR FIT IN REGIONS (0.001-0.100) AND
C ... (0.100-40.0)
C
C ... C GIVES RANGE OF PROTONS IN POLYETHYLENE (MG/CM**2)
C ... D GIVES PROTON ENERGY FOR A GIVEN RANGE IN POLY.
C ... CA,DA GIVE CORRESPONDING VALUES FOR ALUMINUM.
C
C ... ZPOLY AND ZAL CONTAIN THE COEFFICIENTS OF A 6TH ORDER FIT OF A
C ... POLYNOMIAL TO LOG(Z) VS LOG(E) TO DATA DERIVED FROM CALCULATIONS
C ... OF JOSEPH F. JANNI (P.C. JAN'82, TO BE PUBLISHED IN NDT). THE
C ... DATA DESCRIBE THE DETECTED CHARGE AS A FUNCTION OF PROTON ENERGY.
C ... ZEFF = 0. IF E .LE. 0.004MEV, ZEFF = 1.0 IF E .GE. 0.400MEV.
C
C ... FOR EACH ANGLE IN THE MESH THE FOLLOWING APPLY
C ... CS = COSINE OF ANGLE OF SCATTERING
C ... CSQ = COSINE**2
C ... CST = COSINE OF ANGLE RE NORMAL TO FILTER
C ... WT = GEOMETRICAL WEIGHTING FACTOR
C ... AVR = AVERAGE PROTON FLT. PTH. TO DEFINING COLLIMATOR
C
C ... CONST = (2*(ELECT. CHRG)* (AVOGADROS #)/(1000MG/GM*MDL.WT.)*OMEG
C

```



```

0018          CONST = 1.37565E-23*OMEGA
0019          IPOS = 0
0020          TOFP = 0.
      C
      C    . FIND RECOIL PROTON ENERGY
      C
0021          EPMX = EN*CSQ(J)
      C
      C    . CHECK FOR SUFFICIENT ENERGY FOR RESPONSE
      C
0022          IF(EPMX .LT. 0.006) RETURN
0023          IF(FTHICK .EQ. 0.) GO TO 10
      C
      C    . CAN THIS PROTON PENETRATE FILTER?
      C
0024          FTHK = FTHICK/CST(J)
0025          EPMX = ENERGY(FTHK,DA,25)
0026          IF(EPMX .GE. (EPMX+0.004)) RETURN
      C
      C    . FIND MINIMUM ENERGY PROTON WHICH WILL BE DETECTED
      C
0027          10    FTHKP = 0.0
0028          IF(FTHICK .EQ. 0.) GO TO 20
      C    . FTHKP IS CH2 EQUIV OF AL FILTER
0029          FTHKP = RANGE(EPMX,C,25)
      C    . RPMX IS THE RANGE OF EPMX IN CH2
0030          20    RPMX = RANGE(EPMX,C,25)
      C    . FIND RANGE DIFFERENCE BETWEEN MIN AND MAX PROTON ENERGIES
0031          TEFF = RPMX - FTHKP
      C    . TK IS MAXIMUM THICKNESS OF CH2 TARGET TO PROTONS
0032          TK = TTHICK/CS(J)
0033          IF(TEFF .GT. TK) TEFF = TK
0034          IF(DTHICK .EQ. 0.) GO TO 30
      C
      C    . SEE IF MINIMUM ENERGY PROTON PASSED THRU DELTA-E WITH .GT. 0.4MEV
      C    . 0.4MEV IS SUFFICIENT TO NEGLECT ELECTRON PICKUP IN DELTA-E.
      C
0035          DTHK = DTHICK/CST(J)
      C    . FIND CH2 EQUIV OF DTHK, 0.4MEV PROTON IN AL.
0036          DTDEP = RANGE(0.4,C,25)
0037          DTHKP = RANGE(ENERGY(DTHK,DA,25),C,25)
0038          IF(RPMX .GT. (DTDEP+DTHKP+FTHKP+TEFF-0.0001*TK)) RETURN
      C    . FIND PORTION OF CH2 NEAREST DELTA-E WHICH DOES NOT PARTICIPATE
0039          TINIT = RPMX - (DTDEP+DTHKP+FTHKP)
0040          IF(TINIT .LT. 0.) TINIT = 0.
0041          TEFF = TEFF - TINIT
      C
      C    . CHECK FOR SUFFICIENT THICKNESS FOR HI ENERGY PROTONS
      C
0042          IF((TINIT .GT. 0.) .AND. (TEFF .LT. 1.)) RETURN
0043          30    IF(TEFF .LT. 0.0001*TK) RETURN
      C
      C    . COMPUTE NORMALIZATION FACTOR FOR RESPONSE/(UNIT TARG. THICK.)
      C    . SIGH* $\cos(\theta)$  = DIFF. (N,P) CROSS SECTION
      C    .  $\cos(\theta)$  = FACTOR TO ACCOUNT FOR NEUTRON TARGET THICKNESS
      C    . RE TO ANGLE OF SCATTER.

```

```

      C ... WT(J)          = WEIGHTING FACTOR (FROM GEOM) FOR ANGLE J.
      C ... CONST          = EARLIER DEFINITION IN THIS ROUTINE.
      C
0044      FACTOR = CONST*SIGH*CSQ(J)*WT(J)
0045      TOT    = 0.
0046      PRES(1) = 0.
0047      SNSI(1) = 0.
      C
      C ... *****
      C
      C ... COMPUTE RESPONSE FOR UNFILTERED FARADAY CUP.
      C
0048      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
0049      Z      = AVR(J) + ZAD/CST(J)
0050      TNEUT   = TOFF(EN, ZS-ZSS, EON)
0051      TMIN    = TOFF(EPMX, Z, EOP) + TNEUT
0052      EMID    = ENERGY(RPMX-0.5*TEFF, D, 25)
0053      TMID    = TOFF(EMID, Z, EOP) + TNEUT
0054      NTARG   = 2*(TMID-TMIN)/DTAU
0055      IF(NTARG .LT. 4) NTARG = 4
0056      IF(NTARG .GT. NTRT) NTARG = NTRT
0057      DELT    = TEFF/NTARG
0058      NTMX    = NTARG + 1
      C
      C ... COMPUTE PROTON ENERGIES, QEFF, AND TOF FROM THEO. STA. AT EACH MESH
      C
0059      TPOL    = -DELT
0060      DO 100 I=1,NTMX
0061      TPOL    = TPOL + DELT
0062      EP(I)   = ENERGY(RPMX-TPOL, D, 25)
0063      EPP     = EP(I)
0064      IF(EPP .LT. 0.004) EPP = 0.004
0065      TIT(I)  = TOFF(EPP, Z, EOP) + TNEUT
0066      QEFF(I) = ZEFF(ZPOLY, EP(I))
0067      100    CONTINUE
0068      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 QEFF.
      C
0069      DRESP   = 0.5*FACTOR*DELT
0070      DO 102 I=2,NTMX
0071      QFACT   = 2.
0072      IF(EP(I) .GE. 0.4) GO TO 101
0073      QFACT   = QEFF(I-1) + QEFF(I)
0074      IF(EP(I-1) .LE. 0.4) GO TO 101
0075      DIFEF   = 2./(EP(I-1) - EP(I))
0076      QFACT   = DIFEF*((0.4-EP(I))*QEFF(I) + (EP(I-1)-0.4)*QEFF(I-1))
0077      101    PRES(I) = DRESP*QFACT
0078      TOT     = TOT + PRES(I)

```

```

0079      SNSI(I) = TOT
0080      TOFP    = TOFP + 0.5*(TIT(I) + TIT(I-1))*PRES(I)
0081      102     CONTINUE
0082      GO TO 400
C      *****
C
C      COMPUTE RESPONSE FOR AL-FILTERED FARADAY CUP
C
0083      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
0084      Z1      = AVR(J)
0085      Z2      = ZAD/CST(J)
0086      TNEUT   = TOFF(EN, ZS-ZSS, EON)
0087      EPFMX   = ENERGY(RPMX-FTHKP, D, 25)
C
C      SEE IF MAX ENERGY TOO SMALL FOR DETECTION
C
0088      IF(EPFMX .LT. 0.006) RETURN
0089      TMIN    = TOFF(EPMX, Z1, EOP) + TOFF(EPFMX, Z2, EOP) + TNEUT
0090      EMID    = ENERGY(RPMX-0.5*TEFF, D, 25)
0091      EFMID   = ENERGY(RPMX-0.5*TEFF-FTHKP, D, 25)
0092      TMID    = TOFF(EMID, Z1, EOP) + TOFF(EFMID, Z2, EOP) + TNEUT
0093      NTARG   = 2*(TMID-TMIN)/DTAU
0094      IF(NTARG .LT. 4) NTARG = 4
0095      IF(NTARG .GT. NTRT) NTARG = NTRT
0096      DELT    = TEFF/NTARG
0097      NTMX    = NTARG + 1
C
C      COMPUTE PROTON ENERGIES AFTER FILTER, QEFF FOR AL, AND
C      (NEUTRON + PROTON) TOF FROM ZSS AT EACH MESH POINT.
C
0098      TPOL   = -DELT
0099      DO 200 I=1,NTMX
0100      TPOL   = TPOL + DELT
0101      R      = RPMX - TPOL
0102      EP1    = ENERGY(R, D, 25)
0103      EP(I)  = ENERGY(R-FTHKP, D, 25)
0104      EPP    = EP(I)
0105      IF(EPP .LT. 0.004) EPP = 0.004
0106      TIT(I) = TOFF(EP1, Z1, EOP) + TOFF(EPP, Z2, EOP) + TNEUT
0107      QEFF(I) = ZEFF(ZAL, EP(I))
0108      200    CONTINUE
0109      IF(ICNTRL .EQ. 0) GO TO 400
C
C      COMPUTE PARTIAL RESPONSE, INTEGRAL RESPONSE AT EACH MESH POINT
C
0110      DRESP  = 0.5*FACTOR*DELT
0111      DO 202 I=2,NTMX
0112      QFACT  = 2.
0113      IF(EP(I) .GE. 0.4) GO TO 201
0114      QFACT  = QEFF(I-1) + QEFF(I)
0115      IF(EP(I-1) .LE. 0.4) GO TO 201
0116      DIFEF  = 2./(EP(I-1) - EP(I))

```

```

0117      QFACT      = DIFEFF*((0.4-EP(I))*QEFF(I) + (EP(I-1)-0.4)*QEFF(I-1))
0118      201      PRES(I) = DRESP*QFACT
0119      TOT      = TOT + PRES(I)
0120      SNSI(I)  = TOT
0121      TOFF      = TOFF + 0.5*(TIT(I)+TIT(I-1))*PRES(I)
0122      202      CONTINUE
0123      GO TO 400
C ... *****
C
C ... COMPUTE RESPONSE FOR DELTA-E DETECTOR.
C ... USE SAME CRITERIA TO SET UP TARGET MESH AS FOR UNFILTERED F.C.
C
0124      210      Z1      = AVR(J)
0125      Z2      = ZAD/CST(J)
0126      TNEUT    = TOFF(EN,ZS-ZSS,EON)
0127      RTEMP    = RPMX - TINIT
0128      EPMXP    = ENERGY(RTEMP,D,25)
0129      EPFMX    = ENERGY(RTEMP-FTHKP,D,25)
C
C ... SEE IF MAX ENERGY TOO SMALL FOR DETECTION
C
0130      IF(EPFMX .LT. 0.006) RETURN
0131      TMIN      = TOFF(EPMXP,Z1,EOP) + TOFF(EPFMX,Z2,EOP) + TNEUT
0132      TTEMP     = RTEMP - 0.5*TEFF
0133      EMID      = ENERGY(TTEMP,D,25)
0134      EFMID     = ENERGY(TTEMP-FTHKP,D,25)
0135      TMID      = TOFF(EMID,Z1,EOP) + TOFF(EFMID,Z2,EOP) + TNEUT
0136      NTARG     = 2*(TMID-TMIN)/DTAU
0137      IF(NTARG .LT. 4) NTARG = 4
0138      IF(NTARG .GT. NTRT) NTARG = NTRT
0139      DELT      = TEFF/NTARG
0140      NTMX      = NTARG + 1
C
C ... COMPUTE PROTON ENERGIES AFTER FILTER, QEFF, AND (NEUTRON+PROTON)
C ... TOF FROM ZSS AT EACH MESH. QEFF IS (1+ZEFF(ZAL,PROTON ENERGY
C ... AFTER LEAVING DELTA-E)) IF PROTON PASSES THRU DELTA-E.
C ... OTHERWISE QEFF = ZEFF(ZAL,PROTON ENERGY LEAVING FILTER).
C
0141      ETEST    = ENERGY(DTHKP,D,25)
0142      TPOL      = TINIT - DELT
0143      IGT       = 0
0144      DO 302 I=1,NTMX
0145      TPOL      = TPOL + DELT
0146      RTEMP     = RPMX - TPOL
0147      EP1       = ENERGY(RTEMP,D,25)
0148      EP(I)     = ENERGY(RTEMP-FTHKP,D,25)
0149      EPP       = EP(I)
0150      IF(EPP .LT. 0.004) EPP = 0.004
0151      TIT(I)    = TOFF(EP1,Z1,EOP) + TOFF(EPP,Z2,EOP) + TNEUT
0152      IF(EP(I) .GT. ETEST) GO TO 300
C ... PROTON DOES NOT PASS THRU DELTA-E
0153      QEFF(I)  = ZEFF(ZAL,EP(I))
0154      GO TO 302
C ... PROTON PASSES THRU DELTA-E
0155      300      EP(I)  = ENERGY(RANGE(EP(I),C,25)-DTHKP,D,25)
0156      QEFF(I)  = 1. - ZEFF(ZAL,EP(I))

```

```

0157      IQT      = I
0158      302      CONTINUE
0159      IF(ICNTRL .EQ. 0) GO TO 400
C
C ... COMPUTE PARTIAL RESPONSE, INTEGRAL RESPONSE, AND RESPONSE WEIGHTED
C ... (NEUTRON+PROTON) TOF AT EACH MESH POINT.
C
0160      DRESP      = 0.5*FACTOR*DELT
0161      DO 304 I=2,NTMX
0162      QFACT      = QEFF(I-1) + QEFF(I)
0163      IF(I .LT. (IQT+1)) GO TO 1302
0164      IF(I .GT. (IQT+1)) GO TO 1301
C      IF PROTON HAS JUST PENETRATED DELTA E AT BEGINNING OF BIN
0165      QFACT      = 0.5*(QFACT+2.)
0166      GO TO 1302
C ... PROTON DOES NOT PENETRATE THRU DELTA E IN THIS BIN
0167      1301      IF((EP(I) .GE. 0.4) .OR. (EP(I-1) .LE. 0.4)) GO TO 1302
0168      DIFEF      = 2./((EP(I-1) - EP(I)))
0169      QFACT      = DIFEF*((0.4-EP(I))*QEFF(I) + (EP(I-1)-0.4)*QEFF(I-1))
0170      1302      PRES(I) = DRESP*QFACT
0171      TOT        = TOT + PRES(I)
0172      SNSI(I)    = TOT
0173      TOFP       = TOFP + 0.5*(TIT(I) + TIT(I-1))*PRES(I)
0174      304      CONTINUE
0175      400      DIFTT(1) = 1.
0176      DO 402 I=2,NTMX
0177      DIFTT(I) = TIT(I) -TIT(I-1)
0178      IF(DIFTT(I) .LE. 0.) GO TO 403
0179      402      CONTINUE
0180      GO TO 404
C
C ... TRUNCATE DISTRIBUTION IF PROTONS OF LESS THAN 4KEV ARE INVOLVED
C
0181      403      NTMX = I - 1
0182      404      TMINT = TIT(1)
0183      TMAXT     = TIT(NTMX)
0184      IPDS = 1
0185      RETURN
0186      END

```

PROGRAM SECTIONS

Name	Size	Attributes
%CODE1	006206 1603	RW, I, CON, LCL
%PDATA	000260 88	RW, D, CON, LCL
%IDATA	000044 18	RW, D, CON, LCL
%VARS	002112 549	RW, D, CON, LCL
%TEMPS	000022 9	RW, D, CON, LCL
A	002736 751	RW, D, OVR, GBL
B	001352 373	RW, D, OVR, GBL
C	000054 22	RW, D, OVR, GBL
D	000006 3	RW, D, OVR, GBL
RESP1	000022 9	RW, D, OVR, GBL
RESP3	000766 251	RW, D, OVR, GBL

FORTTRAN IV-PLUS V3.0-3
NUEXS.FTN; 4

16:23:55
/TR:BLOCKS/WR

20-May-82

Page 27

Total Space Allocated = 016270 3676

```

0001      FUNCTION PRES P(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 PRES P(T) = 0.
C . . TIMAXP = TIP(NTMAX) = TIME BEYOND WHICH PRES P(T) = SENSI(NTMAX).
C . . TIP = TIME ENTRIES RE TIME NEUTRON ARRIVES AT ZSS.
C . . SENSI = CUMULATIVE PROBABILITY OF RESPONSE BY TIME, TI.
C . . DIFTP = DIFFERENCE BETWEEN ITH AND (I-1)TH TIME ENTRIES.
C
0002      COMMON /RESPO/ NTMAX,TIMIN,TIMAX,TI(41),SENSI(41),DIFTP(41)
0003      COMMON /RESP2/ TIMINP,TIMAXP,TIP(41),DIFTP(41)
0004      PRES P = 0.
0005      IF(T .LE. TIMINP) RETURN
0006      IF(T .LT. TIMAXP) GO TO 10
0007      PRES P = SENSI(NTMAX)
0008      RETURN
C
C . T IS WITHIN RANGE
C
0009      10 DO 20 I=2,NTMAX
0010      IF(T .GT. TIP(I)) GO TO 20
0011      PRES P = SENSI(I-1) + (T-TIP(I-1))*(SENSI(I)-SENSI(I-1))/DIFTP(I)
0012      RETURN
0013      20 CONTINUE
0014      RETURN
0015      END

```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000260 88	RW, I, CON, LCL
\$IDATA	000004 2	RW, D, CON, LCL
\$VARS	000002 1	RW, D, CON, LCL
\$TEMPS	000002 1	RW, D, CON, LCL
RESPO	000766 251	RW, D, OVR, GBL
RESP2	000520 168	RW, D, OVR, GBL

Total Space Allocated = 001776 511

```

0001      FUNCTION DELRES(T)
      C
      C ... ROUTINE TO INTERPOLATE AT TIME, T, IN CUMULATIVE RESPONSE FUNCTION
      C ... SNSI(I) WHICH IS ENTERED AS TABLE VS TIT(I).
      C
      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.
      C
0002      COMMON /RESP3/ NTMX, TMIN, TMAX, TIT(41), SNSI(41), DIFTT(41)
0003      DELRES = 0.
0004      IF(T .LE. TMIN) RETURN
0005      IF(T .LT. TMAX) GO TO 10
0006      DELRES = SNSI(NTMX)
0007      RETURN
      C
      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

Name	Size	Attributes
#CODE1	000260 88	RW, I, CON, LCL
#IDATA	000004 2	RW, D, CON, LCL
#VARS	000002 1	RW, D, CON, LCL
#TEMPS	000002 1	RW, D, CON, LCL
RESP3	000766 251	RW, D, OVR, GBL

Total Space Allocated = 001256 343


```

0001      SUBROUTINE TINT(EX,TX,E,T,N)
      C
      C ... ROUTINE TO LINEARLY INTERPOLATE IN ARRAY T(E) TO FIND TX AT EX.
      C
0002      DIMENSION E(N),T(N)
      C
0003      IF(EX.GT. E(1)) GO TO 10
0004      TX = T(1)
0005      RETURN
0006      10  IF(EX.LT. E(N)) GO TO 20
0007      TX = T(N)
0008      RETURN
0009      20  DO 30 I=2,N
0010      IF(EX.LT. E(I)) GO TO 40
0011      30  CONTINUE
0012      40  SL = (T(I) - T(I-1)) / (E(I) - E(I-1))
0013      TX = T(I-1) + SL * (EX - E(I-1))
0014      RETURN
0015      END
  
```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000366 123	RW, I, CON, LCL
\$IDATA	000040 16	RW, D, CON, LCL
\$VARS	000006 3	RW, D, CON, LCL
\$TEMPS	000002 1	RW, D, CON, LCL

Total Space Allocated = 000436 143

```

0001      FUNCTION TOFF(E,Z,E0)
      C
      C ... ROUTINE TO COMPUTE TIME-OF-FLIGHT OF PARTICLE WITH ENERGY, E,
      C ... REST MASS, E0, 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
0002      DATA C/29.97928/
0003      EP = E
0004      IF(EP .LT. 0.001) EP = 0.001
0005      X = EP/E0
0006      TOFF = Z*(1.+X)/(C*SQRT(X*(2.+X)))
0007      RETURN
0008      END
  
```

PROGRAM SECTIONS

Name	Size	Attributes
#CODE1	000130 44	RW, I, CON, LCL
#PDATA	000004 2	RW, D, CON, LCL
#IDATA	000004 2	RW, D, CON, LCL
#VARS	000014 6	RW, D, CON, LCL

Total Space Allocated = 000154 54

```

0001      FUNCTION ZEFF(Z,E)
      C
      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
      C      ZEFF = 0. FOR E .LE. 0.004MEV
      C      ZEFF = 1. FOR E .GE. 0.400MEV
      C
0002      DIMENSION Z(7)
      C
0003      IF(E .GT. 0.004) GO TO 10
0004      ZEFF = 0.
0005      RETURN
0006      10  IF(E .LT. 0.400) GO TO 20
0007      ZEFF = 1.
0008      RETURN
0009      20  ELN = ALOG(E)
0010      Y = 0.
0011      DO 30 I=1,6
0012      K = 8-I
0013      Y = (Y + Z(K))*ELN
0014      30  CONTINUE
0015      ZEFF = EXP(Z(1) + Y)
0016      RETURN
0017      END
  
```

PROGRAM SECTIONS

Name	Size	Attributes
#CODE1	000276	95 RW, I, CON, LCL
#PDATA	000010	4 RW, D, CON, LCL
#IDATA	000016	7 RW, D, CON, LCL
#VARS	000014	6 RW, D, CON, LCL

Total Space Allocated = 000340 112

```

0001      SUBROUTINE GEOM
C
C ... COMPUTES GEOMETRICAL FRAMEWORK FOR INTEGRATION OF RESPONSE
C
C ... 2-MAR-82
C ... CHANGED CODE TO USE MONTE CARLO METHOD TO CHOOSE POINT
C ... PAIRS ON TARGET AND DETECTOR APERATURE INSTEAD OF DISCRETE
C ... ORDINATES.
C
C ... 6-MAR-82
C ... CHANGE METHOD OF SELECTING RANDOM POINTS TO UTILIZE
C ... SYMMETRY AND HENCE REDUCE COMPUTATION TIME.
C
C ... CODE ASSUMES UNIFORMLY ILLUMINATED TARGET PERPENDICULAR
C ... TO NEUTRON BEAM. TARGET AND APERATURE ARE CIRCULAR.
C ... APERATURE IS PERPENDICULAR TO CENTRAL SCATTERING ANGLE.
C
0002      COMMON/A/CS(75),WT(75),CST(75),CSQ(75),AVR(75),NA
0003      COMMON/B/E(60),SENS(60),TNL,TNM,TNH,NE,TDF(60),DTAU,ZFV,RFV
0004      COMMON/C/RT,RD,ZS,ZTAD,THETA,TTHICK,FTHICK,ZAD,THMX,THMN,OMEGA
0005      COMMON /MCCOM/ CTMC
0006      DIMENSION THTA(75),DTHT(76),XTTT(2),YTTT(2),XDTT(2)
0007      DATA RAD/0.01745328/,EOP/938.256/,CO/29.97928/
C
C ... THETA = CENTRAL ANGLE OF SCATTER
C ... ZTAD = TARGET TO APERATURE DISTANCE
C ... ZTDO = TARGET TO DETECTOR DISTANCE
C ... RD = RADIUS OF APERATURE
C ... RT = RADIUS OF TARGET
C
0008      OMEGA = 0.0
0009      THTA0 = THETA*RAD
0010      SNO = SIN(THTA0)
0011      CSO = COS(THTA0)
0012      TNO = TAN(THTA0)
0013      XDO = ZTAD*SNO
0014      ZO = ZTAD*CSO
0015      RDCS = RD*CSO
0016      RDSN = RD*SNO
C
C ... COMPUTE MIN AND MAX SCATTERING ANGLES
C
0017      THTA1 = ATAN((XDO-RDCS-RT)/(ZO+RDSN))
0018      THTA2 = ATAN((XDO+RDCS+RT)/(ZO-RDSN))
0019      THMX = THTA2/RAD
0020      THMN = THTA1/RAD
0021      ZTDO = ZTAD+ZAD
0022      DO 30 I=1,75
0023      CST(I) = 0.0
0024      AVR(I) = 0.0
0025      WT(I) = 0.0
0026      30 CONTINUE
C ... ZMX = MAXIMUM PROTON FLIGHT PATH
0027      ZMX = ZAD+(ZO-RDSN)/COS(THTA2)
C
C ... COMPUTE INTEGRATION STEP SIZE

```

```

C ... DEN = (DTAU/EP)*(DER(EP) RE TOF(EP)) AT EP=0.1MEV, ZMX
C
0028      X      = .1/EDP
0029      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/ZTAD
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      CSQ(1)  = 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      CSQ(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      CSQ(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 = ZTAD/CSQ
C
C ... COUNTER FOR MATCH ON SOLID ANGLE BETWEEN MESH POINTS
C
0057      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
C

```

```

0059      DO 300 II=1,NA
0060      DO 300 I=1,125
      C
      C ... CHOOSE POINT WITHIN TARGET
      C
0061      RTMC = RT*SQRT(RAN(SEED))
0062      TMC = 6.283185*RAN(SEED)
0063      XTTT(1) = RTMC*COS(TMC)
0064      YTTT(1) = RTMC*SIN(TMC)
0065      XTTT(2) = -XTTT(1)
0066      YTTT(2) = -YTTT(1)
      C
      C ... CHOOSE POINT WITHIN DETECTOR APERATURE
      C
0067      RDMC = RD*SQRT(RAN(SEED))
0068      TMC = 6.283185*RAN(SEED)
0069      XDTT(1) = RDMC*COS(TMC)
0070      YD = RDMC*SIN(TMC)
0071      XDTT(2) = -XDTT(1)
      C
      C ... COMPUTE FOR ALL COMBINATIONS OF +,- XT,YT,XD
      C ... USES INCIDENT SYMETRY, DISCARDS Y-SYM(DET) AS REDUNDANT.
      C
0072      DO 300 IXT=1,2
0073      DO 300 IYT=1,2
0074      DO 300 IXD=1,2
0075      XT = XTTT(IXT)
0076      YT = YTTT(IYT)
0077      XD = XDO + CSO*XDTT(IXD)
      C .. ZOS = DISTANCE FROM TARGET MESH TO DETECTOR PLANE.
0078      ZOS = ZTAD-XT*SND
      C ... RXF = RATIO OF DISTANCES FROM (DET. TO FOV) TO (DET. TO TARG.)
0079      RXF = (ZOS+ZFV-ZTAD)/ZOS
0080      DXS = (XT-XD)**2
0081      XDS = ((XD-XDO)/CSO)**2
      C ... ZTDS = DISTANCE PARALLEL TO BEAM AXIS BETWEEN TARG. MESH
      C ... AND DETECTOR MESH.
0082      ZTDS = ZXDO-XD*TNO
0083      CNT = CNT+1.
      C
      C ... CHECK FOR FOV CONSTRAINT
      C
0084      IF(RFV .LE. 0.0) GO TO 160
0085      XA = (XD-XT)*RXF+XT
      C .. XA = X-COMPONENT OF (TARG-DET) LINE AT ZFV.
      C .. 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.
0086      XAS = ((XA-XAO)/CSO)**2
      C ... YAS = SQUARE OF Y-COMPONENT OF (TARG-DET) LINE AT ZFV.
0087      YAS = ((YD-YT)*RXF+YT)**2
      C
      C .. IF NOT IN FOV, SKIP REST OF LOOP
      C
0088      IF((XAS+YAS) .GT. RFS) GO TO 300
0089      160 CONTINUE

```

```

0090      RTD = SQRT(DXS +(YT-YD)**2)
      C
      C .   RTD = DISTANCE PERPENDICULAR TO BEAM AXIS BETWEEN TARGET MESH
      C .   AND DETECTOR MESH.
      C .   ZTDS = DISTANCE PARALLEL TO BEAM AXIS BETWEEN TARGET MESH
      C .   AND DETECTOR MESH.
      C .   THT = ANGLE OF SCATTERING.
      C .   = ATAN(RTD/ZTDS)
      C
0091      THT = ATAN(RTD/ZTDS)
      C
      C .   THTN = ANGLE OF LINE OF SCATTER RELATIVE TO TARGET NORMAL
      C .   ZOS = COS(THTN) * (DISTANCE BETWEEN TARG. MESH AND DET. MESH)
      C .   = DISTANCE FROM SCATTERING POINT TO DETECTOR PLANE
      C .   R = ZTDS/COS(THT) = DISTANCE BETWEEN TARG. MESH AND DET. MESH.
      C .   OMEGA = R*COS(THTN) * (1/R)**3
      C
0092      COST = COS(THT)
0093      OMEGA = OMEGA+ZOS*(COST/ZTDS)**3
      C
      C .   CHECK TO SEE IF SCATTERING ANGLE MATCHES ANGULAR MESH
      C .   IF SO, ACCUMULATE WT, AVR, CST.
      C .   SUCCESSIVE APPROXIMATION IS USED TO FIND BIN
      C
0094      KMIN = 1
0095      KMAX = NA
0096      DO 200 IK=1,8
0097      K = (KMIN+KMAX)/2
0098      TEST = THTA(K)
0099      IF(THT .GE. (TEST-DTHT(K))) KMIN = K
0100      IF(THT .LE. (TEST+DTHT(K+1))) KMAX = K
0101      IF(KMIN .EQ. KMAX) GO TO 201
0102      200 CONTINUE
      C
      C .   WT = TOTAL MATCHES OF MESH POINTS AT ANGLE K.
0103      201 WT(K) = WT(K)+1.0
      C
      C .   AVR = SUM OF DISTANCES BETWEEN TARGET MESH AND DETECTOR MESH
      C .   AT ANGLE K.
0104      AVR(K) = AVR(K)+ZTDS/COST
      C
      C .   AT THIS POINT CST = SUM OF DISTANCES FROM SCATTERING POINT
      C .   TO DETECTOR PLANE AT ANGLE K.
0105      CST(K) = CST(K)+ZOS
      C
0106      300 CONTINUE
0107      CTMC = 1000.*NA
      C
      C
      C ***DEBUG***
      C
0108      WRITE(4,7000) CTMC, (THTA(I), I=1, NA)
0109      7000 FORMAT(/, 1X, 1PE12.3, ' PARTICLES', /, (1X, 5E10.3))
0110      WRITE(4,7001) (WT(I), I=1, NA)
0111      7001 FORMAT(/, (1X, 1P5E10.3))
      C
      C
      C .   END OF INTEGRATION OVER ALL VARIABLES
      C
      C
      C .   COMPUTE WT, CST, AVR FOR EXTREME ANGLES
      C

```

```

0112      IF(WT(1) .LE. 0.0) WT(1) = 1.0
0113      IF(WT(NA) .LE. 0.0) WT(NA) = 1.0
0114      CST(1) = (ZTAD-RT*SND)*WT(1)
0115      CST(NA) = (ZTAD+RT*SND)*WT(NA)
0116      AVR(1) = (ZTAD*CSO+RD*SND)*WT(1)/COS(THTA1)
0117      AVR(NA) = (ZTAD*CSO-RD*SND)*WT(NA)/COS(THTA2)
      C
      C ... NORMALIZE RESULTS
      C
0118      SUM = 0.0
0119      DO 600 I = 1,NA
0120      IF(WT(I) .LE. 0.0) WT(I) = 1.0
0121      SUM = SUM+WT(I)
0122      IF(AVR(I) .LE. 0.0) GO TO 600
      C ... CST IS REDEFINED TO BE AVERAGE COS(THTN)
0123      CST(I) = CST(I)/AVR(I)
0124      AVR(I) = AVR(I)/WT(I)
0125      600 CONTINUE
      C
      C ... CHECK FOR POSSIBLE ERRORS AND NORMALIZE WT TO 1 FOR INTEGRAL.
      C
0126      DO 700 I=2,NA-1
0127      IF(CST(I) .LE. 0.0) CST(I) = (CST(I+1)+CST(I-1))/2.0
0128      IF(AVR(I) .LE. 0.0) AVR(I) = (AVR(I+1)+AVR(I-1))/2.0
      C ... NORMALIZE WEIGHTS SUCH THAT TOTAL WT = 1
0129      700 WT(I) = WT(I)/SUM
0130      WT(1) = WT(1)/SUM
0131      WT(NA) = WT(NA)/SUM
      C
      C ... AVERAGE THE 1/R**2 CONTRIBUTIONS AT EACH MESH POINT PAIR
      C ... AND MULTIPLY BY THE AREA OF DEFINING COLLIMATOR
      C
0132      OMEGA = 3.1415927*RDS*DMEGA/CNT
0133      RETURN
0134      END
  
```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	004016 1031	RW, I, CON, LCL
\$PDATA	000110 36	RW, D, CON, LCL
\$VARS	001466 411	RW, D, CON, LCL
\$TEMPS	000020 8	RW, D, CON, LCL
A	002736 751	RW, D, OVR, GBL
B	001352 373	RW, D, OVR, GBL
C	000054 22	RW, D, OVR, GBL
MCCOM	000004 2	RW, D, OVR, GBL

Total Space Allocated = 012224 2634


```

C
C ... STRING FINDER SUBROUTINE
C
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(Q,128A1)
0014      1010  FORMAT(1X,'FINDIT CANNOT FIND STRING ',80A1)
0015      END
  
```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000430 140	RW, I, CON, LCL
\$PDATA	000072 29	RW, D, CON, LCL
\$IDATA	000012 5	RW, D, CON, LCL
\$VARS	000206 67	RW, D, CON, LCL
\$TEMPS	000002 1	RW, D, CON, LCL

Total Space Allocated = 000744 242

No FPP Instructions Generated

```

0001      FUNCTION TERP1(X,N,XB,YB)
0002      VIRTUAL XB(300),YB(150)
0003      TERP1 = 0.0
0004      J = 0
0005      NN = 2*N
0006      DO 1 I = 2,NN,2
0007      J = J+1
0008      IF(X-XB(I).LE.0.0)GO TO 2
0009      1 CONTINUE
0010      I = NN
0011      2 CONTINUE
0012      TERP1 = YB(J-1)+(YB(J)-YB(J-1))*(X-XB(I-1))/(XB(I)-XB(I-1))
0013      3 RETURN
0014      END

```

PROGRAM SECTIONS

Name	Size	Attributes
#CODE1	000312 101	RW, I, CON, LCL
#IDATA	000030 12	RW, D, CON, LCL
#VARS	000006 3	RW, D, CON, LCL

Total Space Allocated = 000350 116

```

0001      FUNCTION RANGE(E,C,N)
C
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 ... 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
C ... N          = TOTAL NUMBER OF COEFFICIENTS.
C ... NHI        = N - (NLO + 3).
C
C ... FOR POLY FITS, RANGE = EXP(SUM(C(I)*LOG(E)**(I-1)))
C
0002      PARAMETER NLO=11
0003      DIMENSION C(1)
C
0004      EL = C(1)
0005      RL = C(2)
0006      EM = C(3)
0007      NHI = N-NLO-3
0008      RANGE = 0.
C ... E .LE. 0.?
0009      IF(E .LE. 0.) RETURN
0010      IF(E .GT. EL) GO TO 10
C ... E .LE. EL
0011      RANGE = E*RL/EL
0012      RETURN
0013      10      ELOG = ALOG(E)
0014      IF(E .GT. EM) GO TO 30
C ... EL .LT. E .LE. EM
0015      I1      = 4
0016      NP      = NLO + 4
0017      NM      = NLO - 1
0018      15      Y      = 0.
0019      DO 20 I=1,NM
0020      K      = NP - I
0021      Y      = (Y + C(K))*ELOG
0022      20      CONTINUE
0023      RANGE = EXP(Y + C(I1))
0024      RETURN
C ... E .GT. EM
0025      30      I1      = NLO + 4
0026      NP      = NHI + NLO + 4
0027      NM      = NHI - 1
0028      GO TO 15
0029      END

```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000462 153	RW, I, CON, LCL
\$IDATA	000016 7	RW, D, CON, LCL
\$VARS	000040 16	RW, D, CON, LCL

Total Space Allocated = 000540 176

```

0001      FUNCTION ENERGY(R,C,N)
C
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      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
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      R .LE. 0.?
0009      IF(R .LE. 0.) RETURN
0010      IF(R .GT. RL) GO TO 10
C      R .LE. RL
0011      ENERGY = R*EL/RL
0012      RETURN
0013      10      RLOG = ALOG(R)
0014      IF(R .GT. RM) GO TO 30
C      RL .LT. R .LE. RM
0015      I1 = 4
0016      NP = NLO + 4
0017      NM = NLO - 1
0018      15      Y = 0.
0019      DO 20 I=1,NM
0020      K = NP - I
0021      Y = (Y + C(K))*RLOG
0022      20      CONTINUE
0023      ENERGY = EXP(Y + C(I1))
0024      RETURN
C      R .GT. RM
0025      30      I1 = NLO + 4
0026      NP = NHI + NLO + 4
0027      NM = NHI - 1
0028      GO TO 15
0029      END

```

PROGRAM SECTIONS

Name	Size	Attributes
#CODE1	000462 153	RW, I, CON, LCL
#IDATA	000016 7	RW, D, CON, LCL
#VARS	000040 16	RW, D, CON, LCL

Total Space Allocated = 000540 176