LA--9351

DE82 021173

A Description of the FCUP Code used to Compute Currents due to Recoil Protons from CH₂ Foils

Marion L. Stelts
Dale W. Glasgow
Bernard E. Wood
Alba D. Craft*

DISCLAIMER

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

*EG&G/Las Vegas, P. O. Box 1912, Las Vegas, NV 89101.

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

A DESCRIPTION OF THE FCUP CODE USED TO COMPUTE CURRENTS DUE TO RECOIL PROTONS FROM CH₂ FOILS

by

Marion L. Stelts, Dale W. Glasgow, Bernard E. Wood, and Alba D. Craft

ABSTRACT

A computer code, FCUP, was developed at EG&G during the period from 1973 to the present to compute proton currents produced by a time- and energy-dependent neutron flux striking a CH₂ foil and knocking protons into a detector placed at an angle with respect to the target foil and the neutron beam. This report describes the methods of calculation used and the physical assumptions and limitations involved and suggests possibilities for improving the calculations.

I. DESCRIPTION OF SPECTROMETER

Figure 1 is a schematic representation of the experimental arrangement, showing the neutron source at the bottom. A circular collimator system (RCOL, ZCOL) defines the solid angle for neutrons striking the CH₂ foil at ZS. Another collimation system (RFOV, ZFOV) restricts the field of view (FOV) at the source. Recoil protons from the CH₂ foils are collected by a detector at angle Θ from the neutron beam at a distance (ZTAO + ZAD) from the CH₂ foil. The detector is normally a Faraday cup, designed to have a 50- Ω impedance match and fast response.* If an energy threshold in the response is desired, an aluminum filter may be placed between the target and the Faraday cup. If response caused only by neutrons of a selected energy band is desired, then the Faraday cup is replaced by an aluminum foil with foil and filter thicknesses chosen so that only protons from neutrons within the proper energy band are stopped in the foil. This arrangement is referred to as a ΔE detector. A circular collimator (RD, ZTAO) defines the detector solid angle, and the FOV may be restricted by a collimator at (RFV, ZFV).

II. GENERAL METHOD OF COMPUTATION OF SIGNAL

The program to compute the proton recoil currents superseding SHIV9¹ was evolved by A. D. Craft. This section describes the general method used to calculate the signals.

*This equipment was designed by J. Winston Churchill of EG&G, Inc., 680 Sunset Road, Las Vegas, Nevada.

Fig. 1. Schematic layout of proton recoil spectrometer showing pertinent geometrical parameters.

Because the differential (n, p) scattering cross section² and the range-energy relationships for protons³ in CH₂ and aluminum are well known, it should be possible to compute a response function accurately (Coulombs/neutron). However, as Fig. 1 shows, a variety of scattering angles is possible. A subroutine, GEOM, sets up an angular grid from the geometrical parameters and computes the various average quantities associated with the grid that are needed in computing the response function.

Another subroutine, RESP, uses these quantities (and the CH₂ foil and aluminum filter thicknesses) to compute an energy-dependent sensitivity, SENSE, in Coulombs per neutron for neutrons striking the CH₂ foil. To compute currents accurately, it is necessary to account for the time dispersion of protons (caused by energy losses in matter) from neutrons of a given energy striking the CH₂ foil. The method to be described requires quantities involving the average time of flight (TOF) and its average derivative with respect to neutron energy. Therefore, RESP computes

$$\text{TOF}(E_n) = T_n(E_n) + \left\langle T_p \right\rangle_{E_p, \Theta} \quad (1)$$

where TOF is the total TOF and n and p refer to neutrons and protons for the experimental station and detector geometry. The averaging is over both proton energy, E_p , and possible scattering angles, Θ . RESP also computes

$$\frac{dT}{dE_n} = \frac{dT_n(E_n)}{dE_n} + \left\langle \frac{dT_p}{dE_n} \right\rangle_{E_p, \Theta} \quad (2)$$

The routines GEOM and RESP are discussed in detail in later sections.

We use a separate Monte Carlo neutron transport code to compute the number of neutrons, $FL(t, E)$, arriving in each time and energy bin at a spherical surface of radius ZSS centered about the source. This flux array, the beginning of each time bin, and the average energy in each energy bin are read in and stored. The width of each time bin, $\Delta TB^0(t)$, is computed and stored. The values of flux, $FL(t, E)$, can be adjusted by an energy-dependent factor if desired.

When all parameters describing the Monte Carlo calculation and the experimental configuration are read in, the routine RESP is called and the current is computed. To compute a current, we must know the detected charge in the time bin and the width of the time bin. The charge is computed from $FL(t, E)$, the detector sensitivity, and the solid angle factor for the source. The FCUP method of calculation does not allow protons to be distributed over time bins other than the one where the neutrons that produced them originated. Instead, the time-bin widths are adjusted to compensate for time smearing caused by proton energy dispersion and to account for any differences in foil position between Monte Carlo and experiment.

The following discussion of the method used in adjusting time-bin widths assumes time broadening from TOF effects. If we consider the total flight time T from a point source with delta function output in time for a single neutron energy E_n , then

$$T = T_n(ZS, E_n) + \left\langle T_p(ZP, E_n) \right\rangle_{E_p, \Theta}, \quad (3)$$

where T_n and T_p are the flight times of neutrons and protons, ZS is the neutron flight path, and ZP is the average proton flight path. The average is over all scattering angles, Θ , and all possible proton energies, E_p . We can compute the derivative of T re E_n as

$$\frac{dT}{dE_n} = \frac{\partial T_n}{\partial E_n}(ZS, E_n) + \left\langle \frac{\partial T_p(ZP, E_n)}{\partial E_n} \right\rangle_{E_p, \Theta}, \quad (4)$$

assuming that the partial derivative of the average of a function is approximately equal to the average of the partial derivative of a function. If we replace the differentials with finite differences, we find

$$\Delta T(E_n) = \left(\frac{\partial T_n}{\partial E_n}(ZS, E_n) + \left\langle \frac{\partial T_p(ZP, E_n)}{\partial E_n} \right\rangle_{E_p, \Theta} \right) \Delta E_n. \quad (5)$$

If the Monte Carlo time bins are determined by the same considerations (but without proton TOF contribution) then

$$\Delta T_{MC}(E_n) = \frac{\partial T_n}{\partial E_n}(ZSS, E_n) \Delta E_n, \quad (6)$$

where ZSS is the source-to-CH₂-foil distance in the Monte Carlo calculation. We can now form a ratio that accounts for neutron flight path adjustments and proton time dispersion for a given neutron energy:

$$\frac{\Delta T}{\Delta T_{MC}} = \frac{\frac{\partial T_n}{\partial E_n}(ZS, E_n) + \left\langle \frac{\partial T_p}{\partial E_n}(ZP, E_n) \right\rangle_{E_p, \Theta}}{\frac{\partial T_n}{\partial E_n}(ZSS, E_n)}. \quad (7)$$

New time bins ΔTB_i are then calculated as

$$\Delta TB_i = \Delta TB_i^0 \left\langle \frac{\Delta T}{\Delta T_{MC}} \right\rangle_{E_n}, \quad (8)$$

where the average is over all neutron energies in the i th time bin weighted by the signal contribution from each energy. Note that relativistically

$$\frac{\partial T}{\partial E} = \frac{-Z}{cE_0} \left[\frac{E}{E_0} \left(2 + \frac{E}{E_0} \right) \right]^{-3/2}, \quad (9)$$

where E_0 is the rest mass energy of the particle. Classically,

$$\frac{\partial T}{\partial E} = -\frac{Z}{cE_0} \left(\frac{2E}{E_0} \right)^{-3/2}. \quad (10)$$

The code uses relativistic formulas throughout. However, the Monte Carlo calculation presently uses the classical expression for velocity; hence the nonrelativistic form should be used for $\partial T_n / \partial E_n$ (ZSS, E_n) in Eq. (7). As of January 1982, relativistic formulas are used in the Monte Carlo code as well as in FCUP.

Once the new time-bin widths have been computed, a new time scale is obtained where the beginning of the i th time bin is given by

$$T_i = \sum_{j=1}^{i-1} \Delta TB_j. \quad (11)$$

This time scale is *not* the average TOF of the average energy neutron in that time bin, because $T = 0$ in the Monte Carlo calculation is not the time of maximum neutron production TM . Therefore, if the time scale is to correspond to an average TOF for that bin, it must be translated appropriately. Because of the method of calculating the time widths, a simple translation by TM does not give the correct results. In the past, the time scale was shifted by assuming the first peak in the current to be at the TOF of 14.0-MeV neutrons. This procedure does not work when there are no 14-MeV neutrons in the source, when the detector is not sensitive to 14-MeV neutrons, or when the first peak is not due to 14.0-MeV neutrons. We have therefore adopted a procedure of finding the bin with peak signal and computing the average energy, $EBAR$, for that bin by weighting each energy according to its contribution to the signal. The TOF for this average energy is computed and the time scale shifted so that this time bin has the correct TOF. This procedure does not account for any scattering that occurs in the source but gives a reasonably close time scale based on a bin with strong signal.

We now return to the calculation of the signal. If we know the detector sensitivity as a function of neutron energy, we can compute the charge detected in the i th time bin, $QT(I)$, as

$$QT(I) = \sum_J C(I,J), \quad (12)$$

where

$$C(I,J) = SENS(E_J) \times FL(I,J) \times GFAC \quad (13)$$

and GFAC is the fraction of 4π solid angle subtended by the defining collimator in the neutron line of sight. The adjusted width of the i th time bin is computed as described earlier:

$$\Delta TB(i) = \frac{TB^0(i)}{QT(i)} \sum_j DTNP(E_j) DEDT(E_j) C(i,j) \quad (14)$$

where DTNP is defined by Eq. (2) and DEDT is the inverse of the multiplier of ΔE_n in Eq. (6).

The current, SIG(i), is charge per time interval or

$$SIG(i) = QT(i)/\Delta TB(i) \quad (15)$$

in amperes.

For each time bin, the average energy EBAR(i) is computed as

$$EBAR(i) = \frac{\sum_j E(j) \times C(i,j)}{QT(i)} \quad (16)$$

At this point, we examine the signal to find the bin, IMX, with the greatest signal. A TOF is calculated by interpolating for EBAR in the array defined by Eq. (1). We calculate a time shift, to be added to all times, as

$$TSHFT = TOF[EBAR(IMX)] - 0.5\Delta TB(IMX) + \sum_{j=1}^{IMX-1} \Delta TB(j) \quad (17)$$

The calculation is now complete and the signal and other quantities of interest can be read out for comparison with experiment. A listing of the program is included as the Appendix.

I. RESPONSE FUNCTION: GEOMETRICAL CONSIDERATIONS

The geometrical parameters are defined in Fig. 1. These parameters, (RCOL, ZCOL) and (RFOV, ZFOV), are used to compute the fraction of 4π -sr solid angle subtended by the defining collimator,

$$GFAC = (RCOL/ZCOL)^2/4 \quad (18)$$

and the radius of the illuminated portion of the CH_2 target foil at ZS (as defined by the neutron beam penumbra),

$$RT = (ZS - ZFOV) \frac{(RCOL - RFOV)}{(ZCOL - ZFOV)} - RFOV \quad (19)$$

The detector is at a distance (ZTAO + ZAD) from the center of the target and at an angle Θ with respect to the neutron beam. The CH_2 target is normal to the neutron beam. The defining detector aperture (RD, ZTAO) is normal to the line connecting target center and detector. An FOV constraint (RFV, ZFV) for the detector can be applied.

The absolute response of the detector is a strong function of scattering angle and the total solid angle subtended by the detector. The proton TOF and time dispersion are functions of the average proton flight path, which in turn is a function of the angle of scatter. The velocities of the protons depend on their energy losses in the CH_2 foil and aluminum filter and hence on the

angles of the paths of these protons with respect to the normals of the foil and filter. A subroutine, GEOM, was written to compute these quantities, subject to the assumption of uniform illumination over the radius, RT, of the target.

We first define the quantities generated by GEOM and then describe the method of calculation. An array in scattering angle is chosen such that

$$\Delta\Theta = \frac{d\Theta}{dT} \cdot \Delta T = \frac{d\Theta}{dE_p} \cdot \frac{dE_p}{dT} \cdot \Delta T \quad , \quad (20)$$

where T is the proton TOF, E_p is the proton energy after scattering, and ΔT is the minimum time resolution required. $\Delta\Theta$ is computed at maximum proton flight path (that is, maximum angle of scatter) at 0.1-MeV proton energy. The range in angle from minimum to maximum angle of scatter is defined by the geometries of target and detector apertures. Note that $\Delta\Theta$ is a function of Θ , which minimizes the number of angular elements at the cost of not using a linear array. For each angle, the following quantities are calculated:

$CS(I) = \cos \Theta_i$ = cosine of angle of scatter,

$CSQ(I) = \cos^2 \Theta_i$,

$WT(I)$ = average weight to be given to each angle (normalized to 1 for summation over all angles),

$AVR(I)$ = average proton flight path from foil to detector at each angle, and

$CST(I)$ = average cosine of angle of proton path re the normal to the detector aperture for each scattering angle.

In addition, we compute the average solid angle Ω of the detector aperture as viewed by the target.

These quantities are computed by superimposing a rectangular mesh on both target and detector apertures. The mesh size, DT, is chosen so that

$$DT = \frac{1}{\sin \Theta_{\max}} \frac{dZ}{dT} \Delta T \quad , \quad (21)$$

where $\Theta_{\max}$ is the maximum scattering angle and Z the proton flight path. DT is evaluated at 0.1-MeV proton energy. Then for each pair of mesh points lying to one side of the central scattering plane within radius RT for the target and radius RD for the detector aperture, several calculations are performed.

For the calculation of Ω , a counter is incremented for each mesh point pair. If the line connecting these mesh points is within the FOV of the detector, then

$$\Delta\Omega = \frac{\vec{r} \cdot \vec{dA}}{r^3} \quad (22)$$

is accumulated, where $\vec{r}$ is the vector between mesh points and $\vec{dA}$ is the unit vector normal to the aperture plane. When all possible pairs have been tallied and the average $\Delta\Omega$ computed, the average solid angle

$$\Omega = \pi(RD)^2 \langle \Delta\Omega \rangle \quad (23)$$

is calculated, where $\vec{dA}$ has been renormalized to the area of the detector aperture.

At each mesh point pair that does not conflict with the FOV, we calculate the angle of scatter; we find the index K, determining which angular bin the scattering angle matches. Then WT(K) is incremented by 1.0, AVR(K) is incremented by r (the distance between the mesh point pair), and CST(K) by $r \cos \Theta_r$, where Θ_r is the angle between the direction of scatter and the normal-to-the-detector-aperture plane. After the calculation has been done for each mesh point pair, CST(K) is divided by AVR(K) to give $\langle \cos \Theta_r \rangle$, AVR(K) is divided by WT(K) to give $\langle r_K \rangle$, and WT(K) is normalized to an integrated weight of 1.0 for all angles.

These quantities are used by the routine RESP to compute the energy-dependent response function.

IV. RESPONSE FUNCTION: ENERGY-DEPENDENT QUANTITIES

The routine RESP computes the energy-dependent quantities, which are used to determine the sensitivity and to adjust the TOF scale for differences between theoretical and experimental foil positions and for time dispersion caused by the energy spread of the recoil protons.

For a given neutron energy E and angle of scatter Θ , the contribution to the sensitivity $\Delta S(E, \Theta)$ (in units of Coulombs/neutron) is given by

$$\Delta S(E, \Theta) = \frac{zeN_A}{1000A} t(E, \Theta) W(\Theta) \frac{d\sigma}{d\Omega}(E, \Theta) \Delta\Omega F(E, \Theta) \quad , \quad (24)$$

where

z = number of protons/molecule (2 for CH_2),

e = proton charge (Coulombs),

N_A = Avogadro's number,

A = molecular weight (14 g for CH_2),

1000 = number of mg/g,

$t(E, \Theta)$ = effective target thickness in mg/cm² for neutrons,

$W(\Theta) = WT(\Theta)$, computed by GEOM (the geometrical weighting factor for each scattering angle in angular grid),

$d\sigma/d\Omega(E, \Theta) = (n, p)$ differential scattering cross section (cm²/sr),

$\Delta\Omega$ = solid angle of detector (as computed by GEOM), and

$F(E, \Theta)$ = fractional response for ΔE detectors. ($F = 1.0$ for Faraday cups.)

The total (n,p) scattering cross section is taken from Ref. 2. Assuming *isotropic* scattering in the center-of-mass system, the laboratory differential cross section is

$$\frac{d\sigma}{d\Omega}(E, \Theta) = \text{SIG}(E) \times \cos \Theta \quad , \quad (25)$$

where

$$\text{SIG}(E) = \left[3.0/(f_a^2 + f_c) + 1.0/(f_b^2 + f_c) \right] \text{ bn/sr} , \quad (26)$$

$$f_a = (0.09415 + 0.0001306E) E - 1.86 ,$$

$$f_b = 0.4223 + 0.13E, \text{ and}$$

$$f_c = 1.206E .$$

Any anisotropy would affect the results by less than 1% at 14 MeV and even less for lower neutron energies.

At neutron energy E and scattering angle θ , the energy of the recoil proton will be

$$E_p = E \cos^2 \Theta . \quad (27)$$

The effective target thickness $t(E, \Theta)$ is computed as follows. At a given angle Θ , the maximum amount of CH_2 penetrated by the recoil proton TEFF will be the lesser of

$$\text{TK} = \text{TTHICK}/\cos \Theta$$

or

$$\text{RL} = \text{RANGE (EP) in } \text{CH}_2 ,$$

where TTHICK is the target thickness in mg/cm^2 . If an aluminum filter is used it will have an effective thickness

$$\text{FTHK} = \text{FTHICK}/\text{CST}(\Theta) , \quad (28)$$

where FTHICK is the aluminum filter thickness in mg/cm^2 and $\text{CST}(\Theta)$ has been previously defined. If EPF is the energy of a proton that just penetrates the aluminum filter, then TEFF is the lesser of TK or (RL – Range of EPF in CH_2). Then the effective thickness of CH_2 to neutrons is

$$t(E, \Theta) = \text{TEFF} \times \cos \Theta . \quad (29)$$

Once TEFF has been determined, we can find E_{max} and E_{min} (the maximum and minimum energies) for protons leaving the CH_2 foil and reaching the detector. With the range-energy relationship for aluminum, we can calculate the maximum and minimum energies E'_{max} and E'_{min} of protons penetrating the aluminum filter and striking the detector.

We now consider the detector efficiency $F(E, \Theta)$. If the detector is a Faraday cup (that is, thick to all energies of recoil protons) the $F(E, \Theta) = 1.0$. If, however, the detector is a finite thickness of aluminum, as in a ΔE detector, we compute $F(E, \Theta)$ as follows. The effective thickness of the ΔE foil is

$$\text{DTHK} = \text{DTHICK}/\text{CST}(\Theta) . \quad (30)$$

where DTHICK is the thickness of the ΔE foil in mg/cm^2 . We compute EPDF = (energy of a proton with range DTHK). If $E_{\text{max}} = 0$, no protons reach the detector, so $F(E, \Theta) = 0$. If $\text{EPDF} < E_{\text{min}}$, then no protons would stop in the ΔE detector and $F(E, \Theta) = 0$. If $E_{\text{max}} < \text{EPDF}$, then all protons that pass through the filter would stop in the ΔE and $F(E, \Theta) = 1.0$. For $E_{\text{max}} > \text{EPDF}$,

$$F(E, \Theta) = (EPDF - E'_{min}) / (E'_{max} - E'_{min}) \quad (31)$$

This assumes that the protons are distributed uniformly as a function of proton energy, which is equivalent to assuming that dE/dx is constant over the range of proton energies involved. This approximation becomes poorer as the proton energy decreases.

Because the recoil protons make a finite contribution to the TOF as they travel from foil to detector and also contribute to smearing the response time, it is necessary to compute the average TOF(E) and the average time smearing, DTNP(E), for each neutron energy. The protons make different contributions over the two portions of their flight path, that is, target to filter and filter to detector. For a given angle these distances are AVR(Θ) [the average target-to-defining-collimator (filter) distance that has been calculated in GEOM] and ZAD/CST(Θ), where ZAD is the normal distance from aperture to detector. For each of these distances, Z, the average TOF_p(E) caused by protons is

$$\overline{TOF}_p(E, \Theta) = \frac{\int_{T(\beta_{min})}^{T(\beta_{max})} t(dN_p/dt) dt}{\int_{T(\beta_{min})}^{T(\beta_{max})} (dN_p/dt) dt} \quad (32)$$

where t is the flight time of protons of velocity $V = c\beta$, c is the velocity of light, and dN_p/dt is the distribution of protons as a function of t . $T(\beta)$ gives the limits on t as a function of minimum and maximum β . Note that

$$t = \frac{Z}{c\beta} \quad (33)$$

and

$$\beta = \sqrt{x(2+x)/(1+x)} \quad (34)$$

where $x = E_p/E_{p0}$ and E_{p0} = rest mass energy of the proton. If it is assumed that dN_p/dE_p is a constant, then Eq. (32) can be transformed to

$$\begin{aligned} \overline{TOF}_p(E, \Theta) &= \frac{Z(\Theta)}{c} \frac{\int_{x_{min}}^{x_{max}} 1/\beta dx}{\int_{x_{min}}^{x_{max}} dx} \\ &= \frac{Z(\Theta)}{c} \left[\frac{\sqrt{x_{max}(2+x_{max})} - \sqrt{x_{min}(2+x_{min})}}{(x_{max} - x_{min})} \right] \end{aligned} \quad (35)$$

For calculating the spreading of the time bins caused by energy dispersion of the protons, we must compute

$$\overline{(dt_p/dE_n)}(E, \Theta) = \left\langle \frac{dt_p}{dE_p} \cdot \frac{dE_p}{dE_n} \right\rangle_{E_p} \quad (36)$$

over the proton energy distribution. Note that

$$E_p = E_n \cos^2 \Theta \quad (37)$$

or

$$\frac{dE_p}{dE_n} = \frac{E_p}{E_n} \quad (38)$$

Thus following the derivation of TOF_p we find

$$\begin{aligned} \overline{(dt_p/dE_n)}(E, \Theta) &= \frac{Z}{c} \frac{\int_{\beta_{\max}}^{\beta_{\min}} [d/dE_p (1/\beta)] (E_p/E_n) (dN_p/dE_p) dE_p}{\int_{\beta_{\max}}^{\beta_{\min}} (dN_p/dE_p) dE_p} \\ &= -\frac{Z}{cE_n(x_{\max} - x_{\min})} \int_{x_{\min}}^{x_{\max}} \frac{x/[x(2+x)]^{3/2}}{dx} \\ &= \frac{Z}{cE_n} \left[\frac{\sqrt{x_{\max}/(2+x_{\max})} - \sqrt{x_{\min}/(2+x_{\min})}}{(x_{\max} - x_{\min})} \right] \quad (39) \end{aligned}$$

This routine then sums $\Delta S(E, \Theta)$ over all angles to generate an energy-dependent response

$$SENS(E) = \sum_{\Theta} \Delta S(E, \Theta) \quad (40)$$

for each energy E tabulated in the incident flux. Similarly, for each neutron energy, an average TOF is computed as

$$TOF(E) = \frac{ZS}{c\beta_n} + \frac{\sum_{\Theta} \{ [\overline{TOF}^-(E, \Theta) + \overline{TOF}^+(E, \Theta)] \Delta S(E, \Theta) \}}{SENS(E)} \quad (41)$$

where the superscripts $-$, $+$ refer to proton times of flight before and after the filter.

The following discussion concerns some of the approximations involved in these calculations. The quantities calculated in this routine are highly dependent on knowing accurately the *range-energy relationships* for protons on CH_2 and aluminum. The calculations of Janni³ have been fit by A. Craft with seventh-order polynomials in $\log E$ vs $\log R$ between 0.1 and 20 MeV to derive functions $R(E)$ and $E(R)$ for CH_2 and aluminum. Comparisons of $R(E)$ with Janni's calculations are shown in Figs. 2 and 3. As would be expected, they show rather unreasonable behavior below 100 keV. Still, if $R(E)$ and $E(R)$ were consistent with each other, then $E(R(E))$ would equal E . The absolute value of $[E(R(E)) - E]/E$ vs E is shown in Figs. 4 and 5 for CH_2 and aluminum, revealing huge deviations below 0.1 MeV. As of December 1981, the polynomial fits have been replaced by the dashed lines in Figs. 2 and 3. This is a linear interpolation between 0

Fig. 2. Range vs energy for protons in CH_2 . The crosses are calculations of Ref. 3, the solid line is polynomial fit, and the dotted line shows interpolation that will be used below 100 keV.

Fig. 3. Range vs energy for protons in aluminum. The crosses are calculations of Ref. 3, the solid line is polynomial fit, and the dotted line shows interpolation that will be used below 100 keV.

and the 0.1-MeV calculation of Janni. At some time in the near future, range-energy data that have just become available will be incorporated.

It is also assumed that each proton leaving the target and arriving at the detector will deposit a charge in the detector. Figure 6 shows the average proton charge state as a function of energy for hydrogen passing through matter (Ref. 4). Note that at 0.01 MeV the net positive charge is only 22% of a proton charge. The program should be modified to include this effect.

The assumption is made in computing average TOF and dT_p/dE_n for protons that the protons are uniformly distributed in energy between the minimum and maximum values they can have and still reach the detector, which is equivalent to assuming that the ionization energy loss per unit thickness is independent of energy over the energies of the recoiling protons and that the charge state of the proton is +1 for all energies. The approximation is reasonable at high energies but breaks down at low energies. This deficiency should be corrected in the near future.

Fig. 4. Accuracy of polynomial fits to range energy for CH_2 as a function of energy.

Fig. 5. Accuracy of polynomial fits to range energy for aluminum as a function of energy.

Fig. 6. Equilibrium charge states of hydrogen ions in matter as a function of energy. Reprinted from *Nuclear Reaction Analysis Graphs and Tables* with permission of the publisher and author. See Ref. 4.

V. CONCLUSIONS

The code has proven capable of predicting the experimental currents reasonably accurately. Its accuracy should be best for those portions of the current caused by detected protons from neutrons with energies above a few MeV. If the code is to be used for detailed comparisons with experiment, some areas need attention.

Currently we keep all protons in the same time bin as the neutrons that produce them and then adjust the time-bin widths to account for TOF broadening. This procedure causes a loss of absolute time information relative to the time of maximum neutron production. The problem can be solved by translating the time scale so that the time of the bin with maximum signal is equated to the TOF of the average neutron energy in that bin, but this solution ignores any scattering that may occur in the source. The problem could be eliminated by computing the proton time distribution and distributing the protons into the appropriate fixed time bins.

We have found a problem with the proton range-energy relationships below 100 keV and have applied a temporary solution by using a linear interpolation from 100 keV to 0 energy. The range-energy relationships used in the code were calculated in 1966. More accurate tables will be available soon.

The response function routine assumes that, for a fixed scattering angle, the protons are uniformly distributed in energy. This assumption should be quite good at high proton energies and for thin targets but is suspect otherwise. The problem could be avoided by computing the proton time distribution from a set of planes distributed through the thickness of the CH_2 target. The calculation of the response function assumes that all protons arriving at the detector carry one unit of positive charge. At low proton energies, pickup of one or more electrons is important and there is a net decrease in sensitivity. If the proton time distribution was calculated, this effect could be easily included.

The finite geometrical effects appear to be accounted for quite adequately. One problem might occur where a ΔE detector precedes a thresholded detector. Then the approximation that the filter is located at the defining collimator is not valid.

In conclusion, the existing code should provide a good approximation to response where the proton energy is above a few MeV and should be quite accurate for computing current integrals over the high-energy regions. To offer a more precise tool as experiments become more demanding, the code should be upgraded to provide an absolute time scale and to improve calculation of the low-energy response.

REFERENCES

1. Morris B. Pongratz and J. Lewis Detch, "dE/dx or Threshold Neutron Detectors", EG&G Los Alamos Operations internal memorandum LAO-SPD-275 (10 December 1973).
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, "Calculations of Energy Loss, Range, Pathlength, Straggling, Multiple Scattering, and the Probability of Inelastic Nuclear Collisions for 0.1 to 1000 MeV Protons," Air Force Weapons Laboratory report AFWL-TR-65-150 (1966).
4. J. B. Marion and F. C. Young, *Nuclear Reaction Analysis Graphs and Tables* (North-Holland Publishing Company, Amsterdam, 1968), p. 36.

APPENDIX

FCUP PROGRAM LISTING

```

0001      C      FORTRAN IMAGE ON DM1:CLASS AT [223,202]DEFCUP.FTN
          C      PROGRAM DEFCUP
          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      21-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 TOF 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 TOF OF
          C      NEUTRON WITH AVERAGE ENERGY AT MAXIMUM SIGNAL MATCH TIME SCALE
          C
          C      6-JAN-81 CHANGED TO ALLOW READING MCC 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      8-MAR-82
          C      CORRECTED ERROR IN 'RDSS' IN GEOM. (ROUND-OFF ERROR ON PDP11)
          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),CSG(75),AVR(75),NA,ND,NF
0003      COMMON/B/E(63),SENS(60),DTNP(60),T1,T8,T14,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      VIRTUAL FL(150,60),TT(300),SIG(300),DT(150),GT(150),TIMEX(150)
0007      VIRTUAL DELT(150),GI(150),EBAR(150),GE(60),GN(60),DEDT(60)
0008      DIMENSION F2D(60),FD(2),FSUM(2),ETEXT(3),TTEXT(3,5)
0009      LOGICAL DERUN, RELRUN
0010      BYTE FILNAM(14),TITLE(50),STITL(80),ANS
          C      NEUTRON REST MASS
0011      DATA EDN/939.53/
          C

```

```

C      INITIALIZE 1D-2D FACTORS TO 1.
C
0012      DATA F2D/60*1./
C
0013      1  FORMAT(10I5)
0014      2  FORMAT(1PBE10.3)
0015      3  FORMAT(11X,E11.4)
0016      4  FORMAT(E12.6,17X,E12.6)
0017      5  FORMAT(BF10.0)
0018      6  FORMAT(/,49X,'AVERAGE(E08)=' ,F6.3,/49X,'AVERAGE(E08)=' ,F6.3)
0019      7  FORMAT(2X,'DETECTOR PARAMETERS'//5X,'FILTER THICKNESS=' ,F8.3,
1  ' MG/CM**2' ,T52,'*FTHICK*'//21X,'=' ,F8.3,' MICRONS'//5X,
1  //5X,'DELT E THICKNESS=' ,F8.3,' MG/CM**2' ,T52,'*DTHICK*',
1  //21X,'=' ,F8.3,' MICRONS'//5X,'APERTURE
1  RADIUS=' ,F7.3,' CM' ,T52,'*RD*'//5X,'TARGET TO APERTURE DISTANCE=' ,
1  F7.3,' CM' ,T52,'*2TAD*'//5X,'TARGET TO DETECTOR DISTANCE=' ,F7.3,
1  ' CM'//5X,'APERTURE TO DETECTOR DISTANCE=' ,F8.3,' CM' ,T52,'*ZAD*',
1  //5X,'FOV DISTANCE (FROM TARGET)=' ,F9.3,' CM' ,T52,'*ZFV*'//5X,'FOV
1  RADIUS (DETECTOR)=' ,F8.3,' CM' ,T52,'*RFV*'//5X,'TARGET TO DETECTOR
1  SOLID ANGLE=' ,1PE10.3//5X,'APPROXIMATE TIME RESOLUTION=' ,OPF6.3,
1  ' NSEC' ,T52,'*DTAU*'//5X,'NUMBER OF DETECTOR MESH POINTS=' ,I5//5X,
1  'CENTRAL PROTON SCATTERING ANGLE=' ,OPF7.3,' DEG' ,T52,'*THETA*')
0020      8  FORMAT(Q,128A1)
0021      9  FORMAT(/,2X,'TARGET PARAMETERS'//5X,'TARGET THICKNESS=' ,F8.3,
1  ' MG/CM**2' ,T52,'*TTHICK*'//21X,'=' ,F8.3,' MICRONS'//5X,'TARGET
1  RADIUS=' ,F7.3,' CM'//5X,'EXPERIMENTAL STATION=' ,F9.3,' CM' ,T52,
1  '*ZS*'//5X,'THEORETICAL STATION=' ,F9.3,' CM' ,T52,'*ZSS*'//5X,
1  'DEFINING COLLIMATOR DISTANCE=' ,F9.3,' CM' ,T52,'*ZCOL*'//5X,
1  'DEFINING COLLIMATOR RADIUS =' ,F8.3,' CM' ,T52,'*RCOL*'//5X,'SOURCE
1  TO TARGET SOLID ANGLE=' ,1PE10.3//5X,'SOURCE GEOMETRICAL FACTOR=' ,
1  1PE10.3//5X,'FOV DISTANCE FROM WP=' ,OPF9.3,' CM' ,T52,'*ZFOV*',
1  //5X,'FOV RADIUS=' ,F8.3,' CM' ,T52,'*RFOV*'//5X,'NUMBER OF TARGET
1  MESH POINTS=' ,I5//)
0022      10 FORMAT(1H1,///,2X,50A1)
0023      11 FORMAT(5X,'MAXIMUM SCATTERING ANGLE=' ,F7.3,' DEG'//5X,'MINIMUM
1  SCATTERING ANGLE=' ,F7.3,' DEG'//5X,'NUMBER OF ANGLES=' ,I5,
1  ' (MAX=75)'//2X,'GENERAL CORRECTION FACTOR=' ,1PE10.3,T52,
1  '*GCOR*'//2X,'TIME SHIFT=' ,OPF9.3,' NSEC' ,T52,'*-TMAX*'//2X,
1  'INTEGRAL FROM EMAX TO ' ,F3.1,' =' ,F10.3//2X,
1  'INTEGRAL FROM ' ,F3.1,' TO ' ,F3.1,' =' ,F10.3//2X,'FWHM=' ,F9.3//)
0024      12 FORMAT(/,
1  T(I)      T(I+1)      CURRENT      CHARGE      CHGE
1  SUM' ,6X,'AVE EN'//)
0025      13 FORMAT(3X,I3,OP2F10.3,1P4E12.3)
0026      14 FORMAT(4(OPF10.3,1PE10.3))
0027      15 FORMAT(/,2X,'TIME-INTEGRATED TOTALS')
0028      16 FORMAT(2X,'TOF(' ,F3.1,' )=' ,F10.3//2X,'TOF(' ,F3.1,' )=' ,F10.3//
1  2X,'TOF(14.0)=' ,F9.3//2X,'TOF(' ,F3.1,' )-TOF(14.0)=' ,F9.3//
1  2X,'TOF(' ,F3.1,' )-TOF(14.0)=' ,F9.3//)
0029      17 FORMAT(/,
1  E(I)      TOF(E)      N(E)      Q(E)
1  RESPONSE      1D-2D'//)
0030      18 FORMAT(/,
1  E(I)      TOF(E)      N(E)      Q(E)
1  D. E. RESP      1D-2D'//)
0031      19 FORMAT(1X,F10.3,F12.3,1P3E12.3,OPF9.3)
0032      24 FORMAT(2E12.6)
0033      25 FORMAT(1X,50A1,/,80A1,2E10.3)
0034      26 FORMAT(2X,80A1)

```

```

0035      30  FORMAT(/,I10)
0036      31  FORMAT(2X,2(4XF11.0))
0037      32  FORMAT(21X,F11.0)

C
C ... FIND OUT WHICH MCC VERSION, SET UP REQUIRED PARAMETERS
0038      40  WRITE(5,42)
0039      42  FORMAT(1X,'enter MC version 2A or 2B: ($)')
0040      READ(5,43) NMCC
0041      43  FORMAT(A2)
0042      IF((NMCC.NE.'2A').AND.(NMCC.NE.'2B')) GO TO 40
0043      NENCOL=7
0044      NTCOL=10
0045      IF(NMCC.EQ.'2A') NENCOL=5
0046      IF(NMCC.EQ.'2A') NTCOL=4

C
C ... ASK IF USE REL. KIN.
0047      45  WRITE(5,46)
0048      46  FORMAT(1X,'did MC use relativistic kinematics? ($)')
0049      READ(5,47) ANS
0050      47  FORMAT(A1)
0051      RELRUN=((ANS.EQ.'1H').OR.(ANS.EQ.'1Hq'))

C
C ... OPEN PRINT FILE ON LUN 4 TO AVOID HOGGING LINE PRINTER
0052      OPEN(UNIT=4,NAME='FCUPRINT.PRT',TYPE='NEW',
           1 CARRIAGECONTROL='FORTRAN',DISPOSE='PRINT')

C
C ... GET FILE NAMES AND OPEN
0053      50  ASSIGN 50 TO NRET
0054      WRITE(5,1000)
0055      1000 FORMAT(1X,'enter name of PARAMETER file: ($)')
0056      READ(5,8) NCH, (FILNAM(I), I=1, MINO(NCH,13))
0057      IF((NCH.LE.0).OR.(NCH.GT.13)) GO TO 50
0058      FILNAM(NCH+1) = 0
0059      OPEN(UNIT=1,NAME=FILNAM,TYPE='OLD',READONLY,ERR=2100)
0060      60  ASSIGN 60 TO NRET
0061      WRITE(5,1010)
0062      1010 FORMAT(1X,'enter name of MONTE file: ($)')
0063      READ(5,8) NCH, (FILNAM(I), I=1, MINO(NCH,13))
0064      IF((NCH.LE.0).OR.(NCH.GT.13)) GO TO 60
0065      FILNAM(NCH+1) = 0
0066      OPEN(UNIT=2,NAME=FILNAM,TYPE='OLD',READONLY,ERR=2100)

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
0067      READ(1,5) ZSS, ZCOL, RCOL, ZFOV, RFOV, DTAU, GCOR, TCOR

C

```

```

C ... TCOR IS IGNORED IN THIS VERSION
C ... IF(TCOR .LT. 0.0) GO TO 65
C ... WRITE(5,1015)
C1015  FORMAT(1X, 'I need TCOR = -TMAX: (%)')
C ... READ(5,5) TCOR
0068      65  IF(GCOR .LE. 0.0) GCOR = 1.0
0069      IF(DTAU .LE. 0.0) DTAU = 0.3
0070      GCOL = RCOL/ZCOL
C ... SLOPE OF PENUMBRA
0071      RCF = (RCOL+RFOV)/(ZCOL-ZFOV)
C ... GEOMETRICAL FACTOR OF DEFINING COLL.
0072      GF = GCOL*GCOL/4.0
C ... APPLY GCOR AND CONVERSION TO NSEC TIME BASE TO GF
0073      GFAC = GCOR*GF*1.0E+9
0074      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
0075      READ(1,1) I2D, ITPE, IPLOT
0076      IF(IPLOT.NE.0) ITPE=-1
C
C ... GET OUTPUT FILE IF REQUIRED
0077      IF(ITPE.EQ.0) GO TO 80
0078      70  WRITE(5,1020)
0079      1020  FORMAT(1X, 'enter name of OUTPUT file: (%)')
0080      READ(5,8) NCH, (FILNAM(I), I=1, MINO(13, NCH))
0081      IF((NCH.LE.0).OR.(NCH.GT.13)) GO TO 70
0082      FILNAM(NCH+1)=0
0083      OPEN(UNIT=3, NAME=FILNAM, TYPE='NEW')
C
C ... READ SOURCE TITLE, THEN POSITION TO GET ENERGY AND TIME BINS
0084      80  READ(2,8) NSTITL, (STITL(I), I=1, MINO(80, NSTITL))
0085      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
0086      CALL FINDIT(2, 'TIME', 5, NTCOL)
0087      CALL FINDIT(2, 'ENERGY', 6, NENCOL)
0088      ELOW=0.0
0089      CONST = ZSS/(EON*29.979)
0090      IEN=0
0091      85  IF(IEN.GE.60) STOP 'TOO MANY ENERGIES'
0092      READ(2,1022) ETEXT
0093      1022  FORMAT((NENCOL-3)X, 3A4)
0094      IF((NMCC.EQ.'2A').AND.(ETEXT(2).EQ.'TOT')) GO TO 90
0095      IF((NMCC.EQ.'2B').AND.(ETEXT(2).EQ.'TAL')) GO TO 90
0096      DECODE(10,1025,ETEXT) EHIGH
0097      1025  FORMAT(F10.0)
0098      IEN=IEN+1

```

```

C ... COMPUTE MEAN ENERGY IN EACH BIN
0099      E(IEN) = 0.5*(ELOW + EHIGH)
C ... COMPUTE DER(EN) RE TOF AT ZSS
0100      X      = E(IEN)/EON
C
C ... RELATIVISTIC
0101      IF(RELRUN) DEDT(IEN) = ((X*(2.+X))**.5)/CONST
C ... NONRELATIVISTIC
0102      IF(.NOT.RELRUN) DEDT(IEN) = ((2.*X)**.5)/CONST
0103      ELOW=EHIGH
0104      GO TO 85
0105      90      NE=IEN
C
0106      REWIND 2
0107      IT=0
0108      TIMEX(1)=0.0
0109      100     IF(IT.GE.150) STOP 'TOO MANY TIMES'
0110      CALL FINDIT(2, 'TIME', 5, NTCOL)
0111      BACKSPACE 2
0112      READ(2,1027) TTEXT
0113      1027    FORMAT(<NENCOL+14>X, 5(3A4, 9X))
0114      DO 105 I=1, 5
0115      IF(TTEXT(1, I) EQ. 'TO') GO TO 110
0116      IT=IT+1
0117      DECODE(10, 1025, TTEXT(1, I)) TTEMP
0118      TIMEX(IT+1)=10.0*TTEMP
0119      105     DT(IT)=TIMEX(IT+1)-TIMEX(IT)
0120      GO TO 100
0121      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
0122      REWIND 2
0123      NFULL = NT/5
0124      IF(NFULL .EQ. 0) GO TO 120
0125      DO 115 IPAGE=1, NFULL
0126      CALL FINDIT(2, 'TIME', 5, NTCOL)
0127      CALL FINDIT(2, 'ENERGY', 6, NENCOL)
0128      ITBASE = 1+5*(IPAGE-1)
0129      DO 115 IEN=1, NE
0130      READ(2, 1033) (FL(IT, IEN), IT=ITBASE, ITBASE+4)
0131      115     CONTINUE
C
0132      120     NTLEFT = NT-5*NFULL
0133      IF(NTLEFT .EQ. 0) GO TO 150
0134      CALL FINDIT(2, 'ENERGY', 6, NENCOL)
0135      ITBASE = 1+5*NFULL
0136      DO 130 IEN=1, NE
0137      READ(2, 1033) (FL(IT, IEN), IT=ITBASE, ITBASE+NTLEFT-1)
0138      130     CONTINUE
0139      1033    FORMAT(<NENCOL>X, 5(10X, F11.0))
C
C ... POSITION MC FILE TO GET UNATTENUATED TOTAL COUNTS
0140      150     CALL FINDIT(2, 'RAD', 3, 2)

```

```

0141      IGTPOS = 22+21*NTLEFT + (NENCOL-5)
0142      CALL FINDIT(2, 'TOTAL', 5, IGTPOS)
C
C ... READ FROM MC OUTPUT TAPE GE(I)
C ... GE(I) = UNATTENUATED NUMBER OF NEUTRONS INTO 4PI IN ITH ENERGY BIN
C ... TOTALLED FOR ALL TIMES.
C
0143      CALL FINDIT(2, 'ENERGY', 6, NENCOL)
0144      NSKIP = IGTPOS-7
0145      DO 180 IEN=1, NE
0146      READ(2, 1035) GE(IEN)
0147      180 CONTINUE
0148      1035 FORMAT((NSKIPDXX, F11.0)
0149      CLOSE(UNIT=2)
C
C
C
C
C ... INITIALIZE FD ACCUMULATORS AND READ 2D FACTORS IF REQD
0150      FD(1) = 0.0
0151      FD(2) = 0.0
0152      IF(I2D .EQ. 0) GO TO 290
C
C ... READ IN ENERGY DEPENDENT 1D TO 2D CORRECTION FACTORS
C
0153      READ(1, 5) (F2D(I), I=1, NE)
C
C ... COMPUTE AVERAGE 1D TO 2D CORR FACTORS FOR FISSION, FUSION.
C
0154      J = 1
0155      FD(1) = GE(1)*F2D(1)
0156      FD(2) = 0.0
0157      FSUM(1) = GE(1)
0158      FSUM(2) = 0.0
0159      DO 220 K=2, NE
0160      IF(F2D(K) .LE. 0.0) F2D(K) = F2D(K-1)
0161      IF(E(K) .GT. 8.0) J = 2
0162      FSUM(J) = FSUM(J)+GE(K)
0163      FD(J) = FD(J)+F2D(K)*GE(K)
0164      220 CONTINUE
0165      FD(1) = FD(1)/FSUM(1)
0166      FD(2) = FD(2)/FSUM(2)
C
C ... ADJUST FL(T,E) BY ENERGY DEPENDENT 1D TO 2D F2D(E)
C
0167      DO 230 I=1, NT
0168      K = 1
0169      DO 230 J=1, NE
0170      FL(I, J) = FL(I, J)*F2D(J)
0171      230 CONTINUE
C
C ... PUT DATE INTO FCUP RUN TITLE
0172      290 CALL DATE(TITLE(32))
C
C ... PUT TIME INTO FCUP RUN TITLE AND READ IT
0173      300 TITLE(41)="40

```

```

0174      T = (42) = "40
0175      CALL TIME(TITLE(43))
0176      READ(1,8,END=999) NTITLE, (TITLE(I), I=1, MINO(30, NTITLE))
C      FILL UNUSED TITLE SPACE WITH BLANKS
0177      DO 301 I=NTITLE+1, 31
0178      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
0179      READ(1, 5) 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
0180      READ(1, 5) TTHICK, FTHICK, DTHICK
0181      IF(RD .LE. 0.0) RD = 1.27
0182      IF(THETA .LE. 0.0) THETA = 30.0
0183      IF(FTHICK .LT. 0.0) FTHICK = 0.0
0184      IF(DTHICK .LT. 0.0) DTHICK = 0.0
C
C      COMPUTE THICKNESSES IN MICRONS. RHO(CH2) = 0.92, (AL) = 2.70
C
0185      TTHICKM = TTHICK/0.092
0186      FTHICKM = FTHICK/0.27
0187      DTHICKM = DTHICK/0.27
0188      DERUN = (DTHICKM.NE.0.)
C      COMPUTE RADIUS OF PENUMBRA AT TARGET
0189      RT = (ZS-ZFOV)*RCF-RFOV
C
C      CALL RESP TO GET DETECTOR RESPONSE FUNCTION
C      THIS GIVES CORRECT DELTA-E RESPONSE AS WELL AS THRESHOLDED
C      AND UNTHRESHOLDED RESPONSES.
C
0190      CALL RESP
C
0191      GTOT = 0.0
0192      DT1 = T1-T14
0193      DT2 = T8-T14
0194      DO 310 I=1, 300
0195      310  SIG(I) = 0.0
0196      DO 311 I=1, 150
0197      311  QT(I) = 0.0
0198      DELT(I) = 0.0
0199      EBAR(I) = 0.0
0200      311  QI(I) = 0.0
0201      DO 312 I=1, 60
0202      312  QE(I) = 0.0
0203      QN(I) = 0.0
0204      K = 0

```

```

C
C   COMPUTE SIGNALS FOR EACH TIME AND ENERGY BIN
C
0205      DO 350 I=1,NT
C
C   SUM OVER CONTRIBUTIONS FROM EACH ENERGY BIN
C
0206      DO 330 J=1,NE
0207          SNS = SENS(J)*FL(I,J)*GFAC
C   TOTAL CHARGE IN TIME BIN
0208          QT(I) = QT(I)+SNS
C   TOTAL CHARGE IN ENERGY BIN
0209          QE(J) = QE(J)+SNS
C   TOTAL NEUTRONS IN ENERGY BIN
0210          QN(J) = QN(J)+FL(I,J)
C   AVERAGE ENERGY IN EACH TIME BIN
0211          EBAR(I) = EBAR(I) + E(J)*SNS
C
C   DTA = DT(I) * (DER(TOF) RE EN,EP) AT ZS * (DER(EN) RE TOF) AT ZSS
C
0212          DTA = DT(I)*DTNP(J)*DEDT(J)
C
C   WEIGHT BY RESPONSE
C
0213          DELT(I) = DELT(I)+SNS*DTA
0214      330  CONTINUE
C
C   NORMALIZE TOTAL WEIGHTING TO UNITY
C
0215          IF(QT(I) .GT. 0.0) DELT(I) = DELT(I)/QT(I)
0216          IF(DELT(I) .LE. 0.0) DELT(I) = DT(I)
0217          IF(EBAR(I) .LE. 0.0) EBAR(I) = -1.0
0218          IF(EBAR(I) .GT. 0.0) EBAR(I) = EBAR(I)/QT(I)
0219          QTOT = QTOT + QT(I)
0220          QI(I) = QTOT
0221          K = K+2
C   COMPUTE CURRENT
0222          SIG(K-1) = QT(I)/DELT(I)
0223          SIG(K) = SIG(K-1)
0224      350  CONTINUE
C
C   SHIFT TO PROPER TIME-OF-FLIGHT SCALE
C
C
C   FIND TIME BIN WITH MAXIMUM SIGNAL
C
0225          IMX = 1
0226          K = 0
0227          TEST = 0.0
0228          DO 351 I=1,NT
0229              K = K+2
0230              IF(SIG(K) .LT. TEST) GO TO 351
0231              TEST = SIG(K)
0232              IMX = I
0233      351  CONTINUE
C

```



```

C      FIND AVERAGE TOF FOR EBAR(IMX)
C
0234      EBARP = EBAR(IMX)
0235      DO 352 I=1,NE-1
0236      IF(EBARP .LE. E(I)) GO TO 353
0237      352 CONTINUE
0238      WRITE(5,1500)
0239      WRITE(4,1500)
0240      1500 FORMAT(//1X, '** WARNING - FELL THROUGH FINDING EBAR BIN')
0241      353 SLP = (TOF(I)-TOF(I-1))/(E(I)-E(I-1))
0242      TCRP = 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
0243      TCRP = -0.5*DELT(IMX) + TCRP
0244      DO 354 I=1,IMX-1
0245      TCRP = TCRP - DELT(I)
0246      354 CONTINUE
0247      TT(1) = TCRP
0248      NTS = 2*NT
0249      J = 0
0250      DO 360 I=2,NTS,2
0251      J = J+1
0252      TT(I) = TT(I-1) + DELT(J)
0253      TT(I+1) = TT(I)
0254      360 CONTINUE
0255      IF(TT(NTS) .LT. 100000.) GO TO 361
0256      TT(NTS) = 99999.99
0257      TT(NTS+1) = 99999.99
0258      361 CONTINUE
C
C      IF NOT A DELTA-E, COMPUTE FWHM OF 14 MEV PEAK
C
0259      FWHM = 0.0
0260      IF(DERUN) GO TO 365
0261      JMX = 2
0262      FM = 0.
0263      DO 362 I=2,NTS,2
0264      IF(TT(I) .GE. TB) GO TO 363
0265      IF(SIG(I) .LT. FM) GO TO 362
0266      JMX = I
0267      FM = SIG(I)
0268      362 CONTINUE
0269      363 CONTINUE
0270      FHM = FM/2.0
0271      DO 20 I=1,JMX,2
0272      IF(SIG(I) .GT. FHM) GO TO 21
0273      20 CONTINUE
0274      21 TA = (TT(I)+TT(I-2))/2.0
0275      TB = (TT(I)+TT(I+2))/2.0
0276      TFA = TA+(TB-TA)*(FHM-SIG(I-2))/(SIG(I)-SIG(I-2))
0277      DO 22 I=JMX,NTS,2
0278      IF(SIG(I) .GE. FHM) GO TO 22
0279      IF(SIG(I+2) .LT. FHM) GO TO 23
0280      22 CONTINUE

```

```

0281      23   J = I-1
0282          TA = 0.5*(TT(J)+TT(J-2))
0283          TB = 0.5*(TT(J)+TT(J+2))
0284          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 GI ARE VIRTUAL ARRAYS
0285      365   GSUM1 = TERP1(TB,NT,TT,GI)
0286          GSUM2 = TERP1(T1,NT,TT,GI)-GSUM1
0287          ZTDO = ZTAO+ZAD
0288          WRITE(4,10) TITLE
0289          WRITE(4,26) (STILT(I), I=1,NSTILT)
C
C ... TTHICK = CH2 TARGET THICKNESS (MG/CM**2)
C ... TTHKM = 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 ... NF = NUMBER OF TARGET MESH POINTS.
C
0290      WRITE(4,9) TTHICK,TTHKM,RT,ZS,ZSS,ZCOL,RCOL,OMEG,GF,ZFOV,RFOV,NF
C
C ... FTHICK = AL FILTER THICKNESS (MG/CM**2)
C ... FTHKM = 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 ... ND = NUMBER OF DETECTOR MESH POINTS
C ... DTAU = APPROX. TIME RESPONSE OF DETECTOR (NS)
C ... THETA = CENTRAL PROTON SCATTERING ANGLE
C
0291      WRITE(4,7) FTHICK,FTHKM,DTHICK,DTHKM,RD,ZTAO,ZTDO,ZAD,ZFV,RFV,
1 OMEGA,DTAU,ND,THETA
C
C ... THMX = MAXIMUM SCATTERING ANGLE
C ... THMN = MINIMUM SCATTERING ANGLE
C ... NA = NUMBER OF ANGLES
C ... GCOR = GENERAL CORRECTION FACTOR
C ... TCDR = TIME SHIFT (NS)
C ... E(NE+2) = ENERGY BOUNDARY BETWEEN FUSION AND FISSION
C ... GSUM1 = INTEGRAL CHARGE FOR FUSION
C ... E(NE+1) = LOW BOUND FOR FISSION ENERGY
C ... GSUM2 = INTEGRAL CHARGE FOR FISSION
C ... FWHM = WIDTH OF 14-MEV PEAK IN NSEC.
C
0292      WRITE(4,11) THMX,THMN,NA,GCOR,TCOR,E(NE+2),

```

```

      1 GSUM1,E(NE+2),E(NE+1),GSUM2,FWHM
C
C      E(NE+1) = ENERGY OF LOWER ENERGY BOUND FOR INTEGRALS
C      T1      = TIME OF FLIGHT FOR LOWEST ENERGY.
C      E(NE+2) = ENERGY OF UPPER BOUND FOR FISSION INTEGRAL.
C              (LOWER BOUND FOR FUSION INTEGRAL.)
C      T8      = TIME OF FLIGHT FOR THAT ENERGY.
C      T14     = TIME OF FLIGHT FOR 14 MEV NEUTRON.
C      DT1     = T1 - T14
C      DT2     = T8 - T14
C
0293      WRITE(4,16) E(NE+1),T1,E(NE+2),T8,T14,E(NE+1),DT1,E(NE+2),DT2
0294      WRITE(4,10) TITLE
0295      WRITE(4,26) (STITL(I),I=1,NSTITL)
0296      WRITE(4,12)
0297      N = NT
0298      IF(NT.GT.50) N = 50
0299      J = 1
C
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
0300      DO 370 I=1,N
0301      WRITE(4,13) I,TT(J),TT(J+1),SIG(J),QT(I),GI(I),EBAR(I)
0302      J = J+2
0303      370 CONTINUE
0304      IF(N.EQ.NT) GO TO 380
0305      WRITE(4,10) TITLE
0306      WRITE(4,26) (STITL(I),I=1,NSTITL)
0307      WRITE(4,12)
0308      DO 375 I=1,NT
0309      WRITE(4,13) I,TT(J),TT(J+1),SIG(J),QT(I),GI(I),EBAR(I)
0310      J = J+2
0311      375 CONTINUE
0312      380 CONTINUE
C
C      WRITE SIGNAL TO OUTPUT FILE
C
0313      IF(ITPE.EQ.0) GO TO 390
0314      WRITE(3,25) TITLE,(STITL(I),I=1,NSTITL),FD(1),FD(2)
0315      WRITE(3,1) NTS
0316      WRITE(3,14) (TT(I),SIG(I),I=1,NTS)
0317      WRITE(3,1) NE
0318      WRITE(3,14) (E(I),SENS(I),I=1,NE)
0319      390 CONTINUE
C
C      PRINT ENERGY DEPENDENT PARAMETERS
C
0320      WRITE(4,10) TITLE
0321      WRITE(4,26) (STITL(I),I=1,NSTITL)
0322      WRITE(4,15)
0323      IF(.NOT. DERUN) WRITE(4,17)
0324      IF(DERUN) WRITE(4,18)
C

```

```

C      E      = AVERAGE NEUTRON ENERGY IN BIN (MEV)
C      TOF     = TIME-OF-FLIGHT (NSEC)
C      QN      = 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
0325      DO 400 I=1,NE
0326      WRITE(4,19) E(I),TOF(I),QN(I),QE(I),SENS(I),F2D(I)
0327      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
0328      IF (2D.NE.0) WRITE(4,6) FD(1),FD(2)
0329      GO TO 300
C
C      ALL DONE
0330      $$$ CLOSE(UNIT=1)
0331      CLOSE(UNIT=2)
0332      IF (ITPE.NE.0) CLOSE(UNIT=3)
0333      CLOSE(UNIT=4)
0334      CALL EXIT
C
C      COME HERE ON FILE - OPEN ERROR
0335      2100 WRITE(5,2500) 'FILNAM(I), I=1,NCH)
0336      2500 FORMAT(1X,'can't open file ',14A1)
0337      GO TO NRET
0338      END

```

PROGRAM SECTIONS

Name	Size	Attributes
#CODE1	014174 3134	RW, I, CON, LCL
#PDATA	005000 1280	RW, D, CON, LCL
#IDATA	000170 60	RW, D, CON, LCL
#VARS	001254 342	RW, D, CON, LCL
#TEMPS	000022 9	RW, D, CON, LCL
4	002742 753	RW, D, OVR, GBL
8	001746 499	RW, D, OVR, GBL
1	000054 22	RW, D, OVR, GBL
2	000006 3	RW, D, OVR, GBL

Total Space Allocated = 027654 6102

Total Virtual Array Storage = 673

```

0001      SUBROUTINE RESP
C
C ... COMPUTES ENERGY DEPENDENT RESPONSE FUNCTION FOR A
C ... FARADAY CUP
C
C ... COMPUTES DELTA-E RESPONSE 7-DEC-81
C
0002      COMMON/A/CS(75),WT(75),CST(75),CSG(75),AVR(75),NA,ND,NF
0003      COMMON/B/E(63),SENS(60),DTNP(60),T1,T8,T14,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      DIMENSION C(9),D(9),CA(9),DA(9)
0007      LOGICAL DERUN
0008      DATA C/7.1571451E-01,1.6505563,7.0605302E-02,-2.8448373E-02,
1 6.3265677E-03,6.8427732E-04,-3.2868693E-04,0.10,0.1023/
0009      DATA D/-4.4352142E-01,6.3176679E-01,-2.7926018E-02,1.1650942E-02,
1 -3.5412222E-03,5.4092706E-04,-3.1377325E-05,0.1023,0.10/
0010      DATA CA/1.3806928,1.4399900,1.1090166E-01,-5.5782229E-03,
1 -4.1418937E-03,-4.3407025E-05,2.0957873E-04,0.10,0.2566/
0011      DATA DA/-1.0436047,8.2863318E-01,-6.1210728E-02,6.4122788E-03,
1 -1.86109E-04,-6.0875596E-06,-4.3953575E-07,0.2566,0.10/
0012      DATA PI/3.1415927/,EOP/938.256/,EON/939.55/,CO/29.97928/
C
C ... THE ARRAYS C,D,CA,DA CONTAIN THE COEFFICIENTS WHICH FIT
C ... THE RANGE-ENERGY TABLES OF JOSEPH F. JANNI,
C ... FITTING OF TABLE BY A. CRAFT.
C ... 7 COEFFICIENTS ARE USED IN FIT. THE 8TH AND 9TH ENTRIES GIVE THE
C ... MINIMUM (X,POLY) VALUES FOR THE FIT. BELOW THIS MINIMUM, A LINEAR
C ... INTERPOLATION TO (0,0) IS USED.
C
C ... REPT # AFWL-TR-65-150
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
0013      E(NE+1) = 1.0
0014      E(NE+2) = 8.0
0015      E(NE+3) = 14.0
0016      IF(E(1) .GT. 1.0) E(NE+1) = E(1)
0017      IF(E(1) .GT. 8.0) E(NE+2) = E(1)
0018      C1 = 1.0/CO
0019      CZS = ZS/CO
0020      CZN = CZS/EON
C
C ... SET UP INTEGRATION GRID
C
0021      CALL GEOM
C
C ... FOR EACH ANGLE IN THE MESH THE FOLLOWING APPLY
C ... CS = COSINE OF ANGLE OF SCATTERING
C ... CSG = 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. CHARGE)*(AVOGADROS #)/(1000MG/GM*MOL. WT.)*OMEG
0022      CONST = 1.37565E-23*OMEGA
0023      NE3 = NE+3
C
C      PERFORM INTEGRATION FOR ALL NEUTRON ENERGIES
C
0024      DO 600 I=1,NE3
C
C      PARAMETERIZE H(N,N)H CROSS SECTION
C      SIG = (TOTAL INTEGRATED XSECT IN BARNS)/PI
C      DIFF XSECT = SIG*COS(THETA)
C
C      REFERENCE: J.L. GAMMEL IN 'FAST NEUTRON PHYSICS', PART II, P 2185
C      ED BY MARION AND FOWLER
C
0025      EN = E(I)
0026      FA = (0.09415+0.0001305*EN)*EN-1.86
0027      FB = 0.4223+0.13*EN
0028      FC = 1.206*EN
0029      SIG = (3.0/(FC+FA*FA))+(1.0/(FC+FB*FB))
0030      AVA = 0.0
0031      AVB = 0.0
0032      RSP = 0.0
0033      C2 = C1/EN
C
C      INTEGRATE OVER GRID
C
0034      DO 500 J=1,NA
C
C      ZD = PROTON FLT PATH AT ANGLE J FROM APERTURE(FILTER)
C      TO DETECTOR.
C
0035      FACT = 1./CST(J)
0036      ZD = FACT*ZAD
C      IF .NOT. DELTA-E
0037      FRAC = 1.
C      EP = MAXIMUM ENERGY OF PROTON RECOIL
C      EP = EN * COS(THETA)**2
0038      EP = EN*CSQ(J)
C      RL = RANGE OF EP IN POLY
C      RL = POLY(EP,C,7)
C      TK = POLY THICKNESS / COS(THETA(J))
0040      TK = TTHICK/CS(J)
C      EPD = MINIMUM ENERGY OF PROTON FROM POLY FOIL
C      EPD = POLY(RL-TK,D,7)
0041      IF(EPD .LT. 0.001) EPD = 0.001
0042      IF(RL .LT. TK) TK = RL
0043      TEFF = TK
0044
C
C      IN THE SUBSEQUENT CODING THE FOLLOWING QUANTITIES REGARDING
C      THE PROTON TOF ARE COMPUTED AS MEAN VALUES FROM THE PROTON
C      ENERGY DISTRIBUTION FOR A FIXED NEUTRON ENERGY.
C
C      VA  UA  VB  UB
C
C      'V' REFERS TO TOF BEFORE THE AL FILTER.

```

```

C      'U' REFERS TO TOF AFTER THE AL FILTER.
C      'A' IS THE (MEAN TOF/UNIT LENGTH) IN UNITS OF C.
C      'B' IS THE MEAN ((DER(TOF) RE EP)*(DER(EP) RE EN) / UNIT LENGTH
C      IN UNITS OF (C*EN)
C
C      THE DERIVITIVE OF THE NUMBER OF PROTONS RE EP IS ASSUMED CONSTANT.
C
C      VA = <1/BETA>
C      VB = <(EP/EN)*DER(1/BETA) RE EP>
C
0045      IF(FTHICK .LE. 0.0) GO TO 150
0046      FTHK = FTHICK/CST(J)
C      EPF = ENERGY OF PROTON WHICH JUST PENETRATES AL FILTER
0047      EPF = POLY(FTHK, DA, 7)
C
C      SKIP THIS TO 510 IF PROTON ENERGY LESS THAN
C      1 KEV GREATER THAN REGD TO GET THRU FILTER
C
0048      IF(EP .LE. (EPF+1.0E-3))GO TO 510
C
C      TEFF IS THE RANGE DIFFERENCE IN POLY BETWEEN MAX PROTON
C      ENERGY AND RANGE OF PROTON WHOSE ENERGY IS JUST SUFFICIENT
C      TO PENETRATE ALUMINUM FILTER.
C
0049      TEFF = RL-POLY(EPF, C, 7)
0050      IF(TEFF .GT. TK) TEFF = TK
0051      IF(EPO .LT. EPF) EPO = EPF
C      RLFM= RANGE IN AL OF MAXIMUM ENERGY PROTON RECDIL
0052      RLFM= POLY(EP, CA, 7)
0053      XA = POLY(RLFM-FTHK, DA, 7)/EOP
C      RLF = RANGE IN AL OF MINIMUM ENERGY PROTON RECDIL
0054      RLF = POLY(EPO, CA, 7)
0055      XB = POLY(RLF-FTHK, DA, 7)/EOP
C
C      IF DELTA-E DETECTOR, CONSIDER MAXIMUM PROTON ENERGY DETECTED.
C
0056      IF(.NOT. DERUN) GO TO 120
C      XPDF = MAXIMUM PROTON ENERGY WHICH WILL PENETRATED DELTA-E
C      AND FILTER.
0057      DTHK = DTHICK*FACT
0058      XPDF = POLY(DTHK, DA, 7)/EOP
C
C      CHECK IF MINIMUM - ENERGY PROTON WILL PASS THRU DELTA-E
C
0059      IF(XB .GE. XPDF) GO TO 500
C
C      CHECK IF MAXIMUM - ENERGY PROTON WILL PASS THRU DELTA-E
C
0060      IF(XA .LT. XPDF) GO TO 120
C
C      COMPUTE STOPPING FRACTION IN DELTA-E
C      AND REPLACE XA WITH XPDF
C
0061      IF(XA .NE. XB) FRAC = (XPDF - XB)/(XA - XB)
0062      XA = XPDF
0063      120 CONTINUE

```

```

0064      XRA = XA*(2.+XA)
0065      IF(XA .EQ. XB) GO TO 130
0066      XRB = XB*(2.+XB)
0067      XN = 1./(XA-XB)
0068      UA = (SQRT(XRA)-SQRT(XRB))*XN
0069      UB = (SQRT(XA/(2.+XB))-SQRT(XB/(2.+XB)))*XN
0070      GO TO 150
0071 130    UA = (1.+XA)/SQRT(XRA)
0072      UB = XA/XRA**1.5

C
C    CALCULATION FOR NO AL FILTER
C
0073 150    CONTINUE
0074      XA = EP/EOP
0075      XB = EPD/EOP
0076      XRA = XA*(2.+XA)
0077      XRB = XB*(2.+XB)
0078      XN = 1./(XA-XB)
0079      VA = (SQRT(XRA)-SQRT(XRB))*XN
0080      VB = (SQRT(XA/(2.+XA))-SQRT(XB/(2.+XB)))*XN
0081      IF(FTHICK .GT. 0.0) GO TO 450
0082      UA = VA
0083      UB = VB
0084 450    CONTINUE

C
C    RS = RESP FOR ANGLE J
C    RS = (GEO. WT.)*(COS(FOR DIFF. SCAT))*COS*((TARGET THICK)/COS)
C    RS = WT * COS**2 * TEFF
C    FOR DELTA-E, FRAC = FRACTION OF PROTONS WHICH STOP IN
C    DELTA-E FOIL.
C
0085      RS = WT(J)*CSQ(J)*TEFF*FRAC

C
C    COMPUTE THE WEIGHTED PROTON CONTRIBUTION TO THE TIME-OF-FLIGHT
C    (IN UNITS OF C) AND THE WEIGHTED PROTON TO THE DER(TOF) RE (EN)
C    (IN UNITS OF C*EN). AVR IS THE PROTON FLT PATH TO FILTER.
C    ZD IS THE FLT PATH FROM FILTER TO DETECTOR.
C
0086      AVA = AVA+(AVR(J)*VA+ZD*UA)*RS
0087      AVB = AVB+(AVR(J)*VB+ZD*UB)*RS
0088      RSP = RSP+RS
0089 500    CONTINUE
0090 510    CONTINUE
0091      X = E(I)/EON

C
C    COMPUTE DETECTOR SENSITIVITY
C
0092      SENS(I) = CONST*RSP*SIG

C
C    COMPUTE NEUTRON TOF, DERIVATIVE OF TOF RE EN
C    DT/DE = (L/(C*EON))*(BETA2/(1-BETA2))**-1.5
C    DT/DE = (L/(C*EON))*(X(2+X))**-1.5
C
0093      TOF(I) = CZS*(1.+X)/SQRT(X*(2.+X))
0094      DTNP(I) = CZN/((X*(2.+X))**1.5)
0095      IF(RSP .LE. 0.0) GO TO 600

```



```

      C
      C ... ADD CONTRIBUTION TO TOF AND DTOF/DE BY PROTONS
      C
0096          TOF(I) = TOF(I)+C1*AVA/RSP
0097          DTNP(I) = DTNP(I)+C2*AVB/RSP
0098      600  CONTINUE
      C
      C ... END OF INTEGRATION LOOP
      C
0099          T1 = TOF(NE+1)
0100          T8 = TOF(NE+2)
0101          T14 = TOF(NE+3)
0102          RETURN
0103          END

```

PROGRAM SECTIONS

Name	Size	Attributes
%CODE1	003036 783	RW, I, CON, LCL
%PDATA	000126 43	RW, D, CON, LCL
%IDATA	000020 8	RW, D, CON, LCL
%VARS	000476 159	RW, D, CON, LCL
%TEMPS	000014 6	RW, D, CON, LCL
A	002742 753	RW, D, OVR, GBL
B	001746 499	RW, D, OVR, GBL
C	000054 22	RW, D, OVR, GBL
D	000006 3	RW, D, OVR, GBL

Total Space Allocated = 010710 2276

```

0001      SUBROUTINE GEOM
C
C      COMPUTES GEOMETRICAL FRAMEWORK FOR INTEGRATION OF RESPONSE
C
0002      COMMON/A/CS(75),WT(75),CST(75),CSG(75),AVR(75),NA,ND,NF
0003      COMMON/B/E(63),SENS(60),DTNP(60),T1,T8,T14,NE,TOF(60),DTAU,ZFV,RFV
0004      COMMON/C/RT,RD,ZS,ZTAD,THETA,TTHICK,FTHICK,ZAD,THMX,THMN,OMEGA
0005      DIMENSION THTA(75),DTHT(75)
0006      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
0007      OMEGA = 0.0
0008      THTAD = THETA*RAD
0009      SNO = SIN(THTAD)
0010      CSO = COS(THTAD)
0011      TNO = TAN(THTAD)
0012      XDO = ZTAD*SNO
0013      ZO = ZTAD*CSO
0014      RDCS = RD*CSO
0015      RDSN = RD*SNO
C
C      COMPUTE MIN AND MAX SCATTERING ANGLES
C
0016      THTA1 = ATAN((XDO-RDCS-RT)/(ZO-RDSN))
0017      THTA2 = ATAN((XDO+RDCS+RT)/(ZO-RDSN))
0018      THMX = THTA2/RAD
0019      THMN = THTA1/RAD
0020      ZTDO = ZTAD+ZAL
0021      DO 30 I=1,75
0022      CST(I) = 0.0
0023      AVR(I) = 0.0
0024      WT(I) = 0.0
0025      30 CONTINUE
C      ZMX = MAXIMUM PROTON FLIGHT PATH
0026      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
0027      X = .1/EOP
0028      DEN = ((DTAU*CO)/(X*ZMX))*(X*(2.+X))**1.5
0029      DEN2 = DEN/2.0
C
C      DT = DELTAZ/SIN(THETA)
C      DELTAZ = (DER(Z) RE TOF) * DTAU FOR EP=0.1MEV, ZMX
C      = (DER(Z) RE THETA)*(DER(THETA) RE EP) * EP * DEN
C      NOTE: EP = EN*COS(THETA)**2
C
0030      DT = (ZO-RDSN)*DEN/SIN(2.0*THTA2)
0031      NT = RT/DT
0032      ND = RDCS/DT

```

```

0033      IF(ND .LT. 5) ND = 5
0034      DT = RT/NT
0035      NTO = NT+1
0036      NT = 2*NT+1
0037      DDX = RDCS/ND
0038      DDY = RD/ND
C
C      ND =      NUMBER OF DETECTOR MESH POINTS
C      NF = NT = NUMBER OF TARGET  MESH POINTS
C
0039      ND = 2*ND+1
0040      NF = NT
0041      RTS = RT*RT
0042      RDS = RD*RD
C
C      IF ROUND OFF ERROR IS SUCH THAT THE INITIAL POINT
C      IN TRACING ACROSS DETECTOR (Y) IS OUTSIDE CIRCLE
C      ENTIRE TRACE IS SKIPPED (EXCEPTING ONE POINT).
C      THERE FOR INCREASE RDS WHICH IS USED IN TEST.
C
0043      RDSS = RDS*(1.0002)
C      TARGET FOV PARAMETERS
0044      RFS = RFV*RFV
0045      XAD = 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
0046      CS(1) = COS(THTA1)
0047      CSQ(1) = CS(1)*CS(1)
0048      THTA(1) = THTA1
0049      DO 50 I=2,75
0050      DTHT(I-1) = DEN2/TAN(THTA(I-1))
0051      THTA(I) = THTA(I-1)+DTHT(I-1)
0052      IF(THTA(I) .GE. THTA2) GO TO 55
0053      CS(I) = COS(THTA(I))
0054      CSQ(I) = CS(I)*CS(I)
0055      50  CONTINUE
0056      NA = 75
0057      GO TO 60
0058      55  NA = I
0059      60  THTA(NA) = THTA2
0060      CS(NA) = COS(THTA2)
0061      DTHT(NA-1) = THTA(NA)-THTA(NA-1)
0062      CSQ(NA) = CS(NA)*CS(NA)
C
C      START OF INTEGRATION OVER ALL VARIABLES
C
0063      XT = -RT-DT
0064      XD1 = XDO-RDCS-DDX
0065      YD1 = -RD-DDY
0066      ZXDO = ZTAD/CSQ
C

```

```

C      COUNTER FOR MATCH ON SOLID ANGLE BETWEEN MESH POINTS
C
0067      CNT = 0.
C
C      INTEGRATE OVER TARGET MESH IN DIRECTION OF BEAM-DETECTOR PLANE
C      FIRST MESH POINT IS AT -RT.
C
0068      DO 500 I=1,NT
0069          XT = XT+DT
0070          XTS = XT*XT
C      ZOS = DISTANCE FROM TARGET MESH TO DETECTOR PLANE.
0071          ZOS = ZTAD-XT*SNO
C      RXF = RATIO OF DISTANCES FROM (DET. TO FOV) TO (DET. TO TARG.)
0072          RXF = (ZOS+ZFOV-ZTAD)/ZOS
C
C      INTEGRATE TARGET MESH PERPENDICULAR TO BEAM-DETECTOR PLANE
C
0073          YT = -DT
0074          KYT = 0
C
C      CHECK TO SEE IF TARGET MESH IS WITHIN TARGET CIRCLE
C
0075          IF(XTS .LT. RTS) GO TO 90
0076          XTS = RTS
0077          KYT = 1
0078      90      CONTINUE
C
C      YT1 = MAX EXTENSION OF YT IN TRANSVERSE TARGET DIRECTION
C
0079          YT1 = SQRT(RTS-XTS)
0080          DO 400 J=1,NT0
0081              YT = YT+DT
C
C      CHECK TO SEE IF TARGET MESH POINT IS IN TARGET CIRCLE
C
0082          IF((XTS+YT*YT) .LE. RTS) GO TO 100
0083          YT = YT1
0084          KYT = 1
0085      100      CONTINUE
0086          XD = XD1
C
C      INTEGRATE OVER DETECTOR (X)
C
0087          DO 300 IJ=1,ND
0088              XD = XD+DDX
0089              DXS = (XT-XD)**2
0090              XDS = ((XD-XD0)/CSD)**2
C      ZTDS = DISTANCE PARALLEL TO BEAM AXIS BETWEEN TARG. MESH
C      AND DETECTOR MESH
0091              ZTDS = ZXDO-XD*TNO
0092              KYD = 0
0093              IF(XDS .LT. RDS) GO TO 130
0094              XDS = RDS
0095              KYD = 1
0096      130      CONTINUE
C

```

```

C ... INTEGRATE OVER DETECTOR (Y)
C ... -YD1 = STARTING POINT IN DETECTOR MESH (TRANSVERS)
C
0097      YD1 = SQRT(RDS-XDS)
0098      YD = -YD1
0099      DO 250 JK=1,ND
C
C ... CHECK TO SEE IF DET. MESH POINT IS WITHIN APERATURE
C
0100      IF((XDS+YD*YD) .LT. RDSS) GO TO 150
0101      YD = YD1
0102      KYD = 1
0103      150 CONTINUE
0104      CNT = CNT+1.
C
C ... CHECK FOR FOV CONSTRAINT
C
0105      IF(RFV .LE. 0.0) GO TO 160
0106      XA = (XD-XT)*RXF+XT
C      XA = X-COMPONENT OF (TARG-DET) LINE AT ZFV.
C      XAD = 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.
0107      XAS = ((XA-XAD)/CSD)**2
C      YAS = SQUARE OF Y-COMPONENT OF (TARG-DET) LINE AT ZFV.
0108      YAS = ((YD-YT)*RXF+YT)**2
C
C ... IF NOT IN FOV, SKIP REST OF LOOP
C
0109      IF((XAS+YAS) .GT. RFS) GO TO 240
0110      160 CONTINUE
0111      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
0112      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
0113      OMEGA = OMEGA+ZOS*(COS(THT)/ZTDS)**3
C
C ... CHECK TO SEE IF SCATTERING ANGLE MATCHES ANGULAR MESH
C ... IF SO, ACCUMULATE WT, AVR, CST.
C
0114      DO 200 K=1,NA
0115      IF(THTA(K) .LT. (THT-.5*DTHT(K))) GO TO 200
C ... WT = TOTAL MATCHES OF MESH POINTS AT ANGLE K.

```

```

0116      WT(K) = WT(K)+1.0
C      AVR = SUM OF DISTANCES BETWEEN TARGET MESH AND DETECTOR MESH
C      AT ANGLE K.
0117      AVR(K) = AVR(K)+ZTDS/COS(THT)
C      AT THIS POINT CST = SUM OF DISTANCES FROM SCATTERING POINT
C      TO DETECTOR PLANE AT ANGLE K.
0118      CST(K) = CST(K)+ZDS
0119      GO TO 240
0120      200  CONTINUE
C
0121      240  YD=YD+DDY
C
C      IF KYD .GT. 0 FINISHED WITH LOOP 250 (TRANSVERSE DET. MESH)
C
0122      IF(KYD .GT. 0) GO TO 300
0123      250  CONTINUE
0124      300  CONTINUE
C
C      IF KYT .GT. 0 FINISHED WITH LOOP 400 (TRANSVERSE TARG. MESH)
C
0125      IF(KYT .GT. 0) GO TO 500
0126      400  CONTINUE
0127      500  CONTINUE
C
C      END OF INTEGRATION OVER ALL VARIABLES
C
C      COMPUTE WT,CST,AVR FOR EXTREME ANGLES
C
0128      IF(WT(1) .LE. 0.0) WT(1) = 1.0
0129      IF(WT(NA) .LE. 0.0) WT(NA) = 1.0
0130      CST(1) = (ZTAD-RT*SNO)*WT(1)
0131      CST(NA) = (ZTAD+RT*SNO)*WT(NA)
0132      AVR(1) = (ZTAD*CSD+RD*SNO)*WT(1)/COS(THTA1)
0133      AVR(NA) = (ZTAD*CSD-RD*SNO)*WT(NA)/COS(THTA2)
C
C      NORMALIZE RESULTS
C
0134      SUM = 0.0
0135      DO 600 I = 1,NA
0136      IF(WT(I) .LE. 0.0) WT(I) = 1.0
0137      SUM = SUM+WT(I)
0138      IF(AVR(I) .LE. 0.0) GO TO 600
C      CST IS REDEFINED TO BE AVERAGE COS(THTN)
0139      CST(I) = CST(I)/AVR(I)
0140      AVR(I) = AVR(I)/WT(I)
0141      600  CONTINUE
C
C      CHECK FOR POSSIBLE ERRORS AND NORMALIZE WT TO 1 FOR INTEGRAL.
C
0142      DO 700 I=2,NA-1
0143      IF(CST(I) .LE. 0.0) CST(I) = (CST(I+1)+CST(I-1))/2.0
0144      IF(AVR(I) .LE. 0.0) AVR(I) = (AVR(I+1)+AVR(I-1))/2.0
C      NORMALIZE WEIGHTS SUCH THAT TOTAL WT = 1.
0145      700  WT(I) = WT(I)/SUM
0146      WT(1) = WT(1)/SUM
0147      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
0148      OMEGA = 3.1415927*RDS*OMEGA/CNT
0149      RETURN
0150      END

```

PROGRAM SECTIONS

Name	Size		Attributes
*CODE1	003722	1001	RW, I, CON, LCL
*PDATA	000014	6	RW, D, CON, LCL
*VARS	001446	403	RW, D, CON, LCL
*TEMPS	000030	12	RW, D, CON, LCL
A	002742	753	RW, D, OVR, GBL
B	001746	499	RW, D, OVR, GBL
C	000054	22	RW, D, OVR, GBL

Total Space Allocated = 012420 2696

```

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,MIN0(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(9,128A1)
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 POLY(X,F,N)
      C
      C ... COMPUTES POLY = EXP(Y), WHERE
      C ... Y = SUM(F(I)*LOG(E(X))**(I-1)), I=1,N
      C ... POLY(X<0) = 0.
      C
      C ... THE N+1,N+2 COEFFICIENTS GIVE THE MINIMUM (X,POLY) VALUES WHICH
      C ... CAN BE USED. LINEAR INTERPOLATION TO (0,0) IS USED BELOW.
      C
0002      DIMENSION F(1)
0003      POLY = 0.0
      C ... ARG .LE. XMIN
0004      IF(X .LE. 0.0) RETURN
      C ... ARG .LT. XMIN
0005      XMIN = F(N+1)
0006      IF(X .GT. XMIN) GO TO 5
0007      POLY = X*F(N+2)/XMIN
0008      RETURN
0009      5  XL = ALOG(X)
0010      Y = 0.0
0011      NP = N+1
0012      NM = N-1
0013      DO 10 I = 1,NM
0014      K = NP-I
0015      10 Y = (Y+F(K))*XL
0016      IF(Y.LT.-200.0)RETURN
0017      POLY = EXP(Y+F(1))
0018      RETURN
0019      END

```

PROGRAM SECTIONS

Name	Size	Attributes
%CODE1	000354 118	RW, I, CON, LCL
%IDATA	000016 7	RW, D, CON, LCL
%VARS	000024 10	RW, D, CON, LCL

Total Space Allocated = 000416 135

```

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