LBL-12244

MASTER

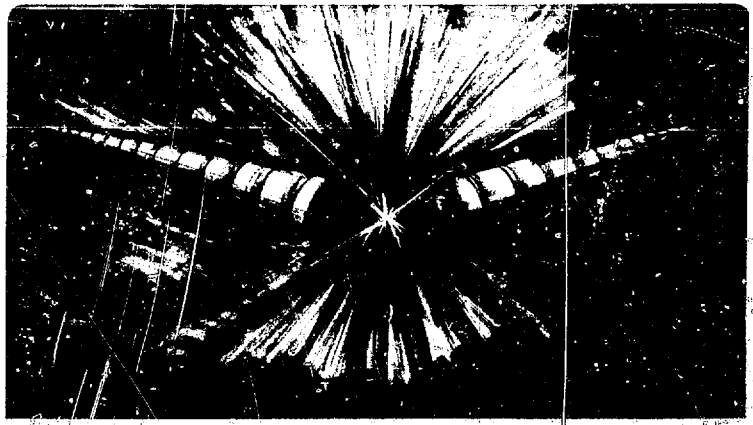
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Accelerator & Fusion Research Division

ACDOS1: A COMPUTER CODE TO CALCULATE DOSE RATES FROM NEUTRON ACTIVATION OF NEUTRAL BEAMLINES AND OTHER FUSION-REACTOR COMPONENTS

Gregory S. Keney
(M.S. thesis)

August 1981



Prepared for the U.S. Department of Energy under Contract W-7405-ENG-48

DIST

LBL-12244

DISCLAIMER

ACDOS1: A Computer Code to Calculate Dose Rates From Neutron Activation of Neutral Beamlines and Other Fusion-Reactor Components

By Gregory S. Keney

(M.S. Thesis)

Accelerator and Fusion Research Division
Lawrence Berkeley Laboratory
University of California
Berkeley, California, 94720

August 1981

Submitted in partial fulfillment of the requirements for the M.S. Degree, to the Department of Nuclear Engineering, University of California, Berkeley, California

This work was supported by the Director, Office of Energy Research, Office of Fusion Energy, Development and Technology Division of the U.S. Department of Energy under Contract No. W-7405-Eng-48.

This manuscript was printed from originals provided by the author.

ABSTRACT

A computer code has been written to calculate neutron induced activation of neutral-beam injector components and the corresponding dose rates as a function of geometry, component composition, and time after shutdown. The code, ACDOS1, was written in Fortran IV to calculate both activity and dose rates for up to 30 target nuclides and 50 neutron groups. Sufficient versatility has also been incorporated into the code to make it applicable to a variety of general activation problems due to neutrons of energy less than 20 MeV.

Table of Contents

Page	Subject
1	Introduction
4	Neutron Production Model
10	Derivation and Modification of the Activation Equation
15	Determination of Weighting Flux and Flux Weighted Group Cross Sections
37	Program to Calculate Activities
41	Program to Calculate Dose Rates
49	Code Description
64	Solved Sample Problem
68	Project Conclusions
75	Appendix A - Solutions to Transcendental Equations
86	Appendix B - Newton-Raphson Method
88	Appendix C - ACDOS1 Source Listing

Introduction

This project has originated from a need in the fusion community to calculate dose rates from neutron-induced activation of fusion components. In particular, the subject matter of this work revolves around the calculations required to determine the dose rates from neutral-beam injector components as a function of geometry, component composition, and time after shutdown.

The methodology for solving the overall problem is broken down into essentially five major steps. The first is that of determining a mathematical model for the production of neutrons by the neutral-beam injector. In practice, neutron production is not uniform but occurs during a succession of evenly spaced pulses of short time duration. During the pauses, no neutrons are produced. An effective steady neutron source term is generated by utilizing an appropriate duty factor over the time duration of the pulses. The neutron production rate is then calculated on the basis of the energy and current of the neutral beam. The second step required writing and solving the pertinent ordinary differential equation that describes the system undergoing activation and then algebraically modifying the standard solution to make it amenable to the arbitrary injector test schedule and adapting the results to multigroup calculations. With the unity-normalized

average group fluxes assumed to be available from a previous calculation, the third step involves calculating the flux weighted group cross sections by making use of a library containing microscopic neutron cross sections. Prerequisite to this calculation is the determination of a weighting flux for use in averaging cross section data over the appropriate energy groups. This was accomplished by assuming a room-temperature Maxwell-Boltzmann distribution for the thermal region, a 1/E function for intermediate energies, and an exponential function for the fast neutron groups. The forms of the equations are all known. Arbitrary constants in the weighting fluxes are determined by integral and/or boundary conditions on the known flux for the particular energy group in question. With the weighting flux a determinable function of energy, the required flux weighted group cross sections can be evaluated. The fourth step involves a calculation of the resultant activities given the neutron production rate, the flux weighted group cross sections, target nuclide mass and type, and the specific times after shutdown at which induced activities are desired. The routine which calculates activity interrogates the cross section library to find the atomic weight of the specific target nuclide and in addition, the product nuclide and its half-life. The fifth and final step is a calculation of the dose rates as a function of time and geometry given the

previously calculated activities. This last routine interrogates a decay library and extracts the necessary gamma ray energy and intensity information. The program incorporates a gamma-flux-to-dose-rate function for several simple geometries. The user specifies the type of geometry to be used in the actual calculations-point, spherical, or cylindrical-on-axis, in addition to the distance from the activated component at which the dose rate is desired.

The result of the five steps is a code called ACDOS1 which performs the activation and dose rate calculations given the required input information. The code can handle from one to fifty neutron groups with up to thirty target nuclides and has an option which allows the user to substitute a neutron source term other than that from a neutral-beam injector, thereby increasing the versatility of the code. In addition, the entire code is written in ANSI FORTRAN IV which should minimize compiler compatability problems at other user sites. Interested users should submit a written request for additional information. Address requests to: Head, Neutral Beam Development Group, Bldg. 5, Lawrence Berkeley Laboratory, Berkeley, California, 94720.

The end result of this project, then, is a code that is tailored to solve injector activation problems, but contains enough versatility to be useful in solving a variety of general activation problems produced by neutrons of energy less than 20 MeV.

Neutron Production Model

The first major step in solving the overall activation problem is to determine a mathematical model for the production of neutrons by the injector assembly. As previously mentioned, neutron production is not uniform but occurs during an arbitrary succession of evenly spaced injector pulses of short time duration. Neutron production coincides with the injector pulses. Figure 1 depicts schematically how this pulse-pause nature of neutron production might be viewed. INSNPS is the instantaneous number of neutrons produced per second. T1 is the length of time the short pulses and pauses are occurring and T2 is the length of time in which no pulsing-pausing occurs. It is assumed throughout the activation calculations that T1 and T2 do not vary during the operation of the injector.

The neutrons originate from two sources within the injector system. Gaseous deuterium that leaks from the neutralizer and ion-source adsorbs on the surface of the copper target located in the ion-dump and neutral-target sections of the injector. A (d,n) reaction occurs at the far end of the injector as D° impinges upon the deuterium. The reaction is:

$$^{2}H + {}^{2}H \longrightarrow {}^{3}He + {}^{1}n + MeV$$

In addition, a similar reaction occurs as D^+ impinges upon deuterium adsorbed on the copper target located in the ion-dump section.

The calculation done to determine the instantaneous number of neutrons produced per second is empirical in nature and taken under accelerator test conditions.[1] The expression used is:

$$\text{INSNPS} = 8.64 \times 10^4 \frac{\text{neutrons}}{\mu\text{coulomb}} \times A \times 1.0 \times 10^6 \frac{\mu\text{coulomb}}{\text{sec}} \times F \times CF$$

where the variables are defined as below:

INSNPS: the instantaneous number of neutrons produced per second

A: current in amperes

F: fraction of the beam that is monatomic

CF: yield correction factor for voltages different from 150 keV

The yield correction factor for voltages different than 150 keV is taken from the calculation[2] reproduced in Figure 2. To facilitate the calculation of INSNPS during execution of the program, a power fit of the form $y=a^x$ was applied to the points read from the curve to determine an analytical expression for CF. Three different fits were used to insure a high coefficient of determination. These expressions are incorporated into subroutine SOURCE which

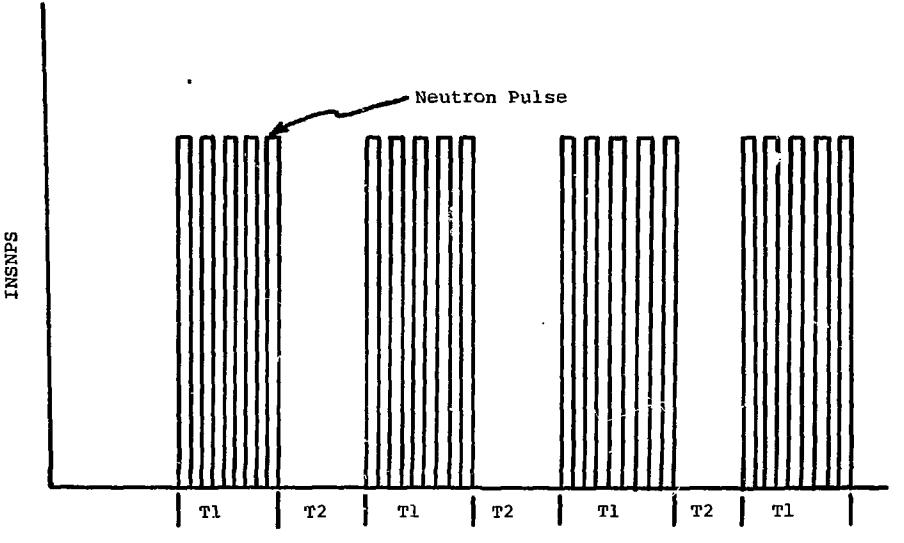
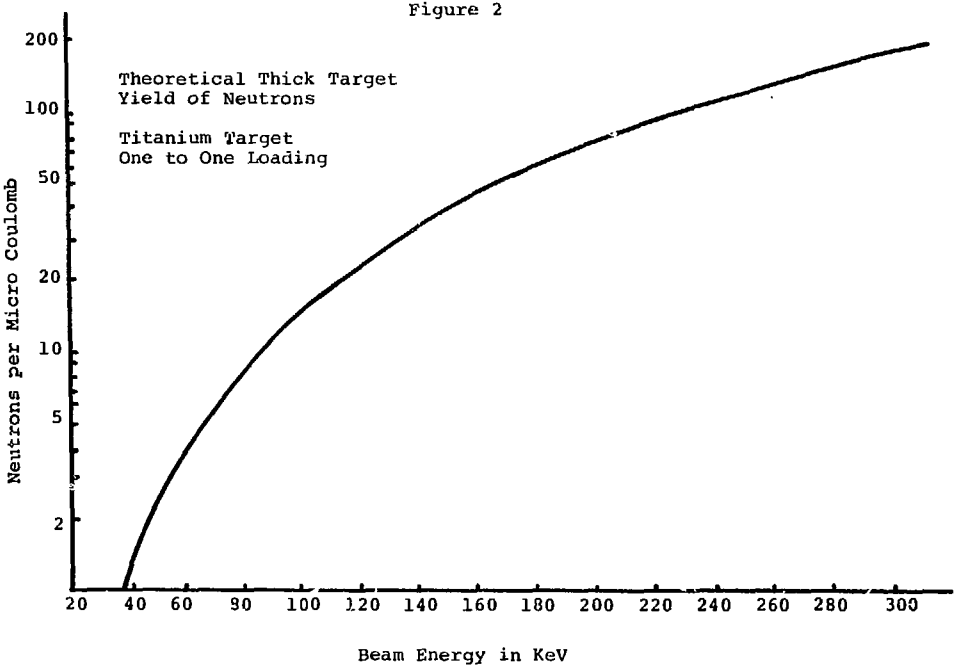


Figure 1

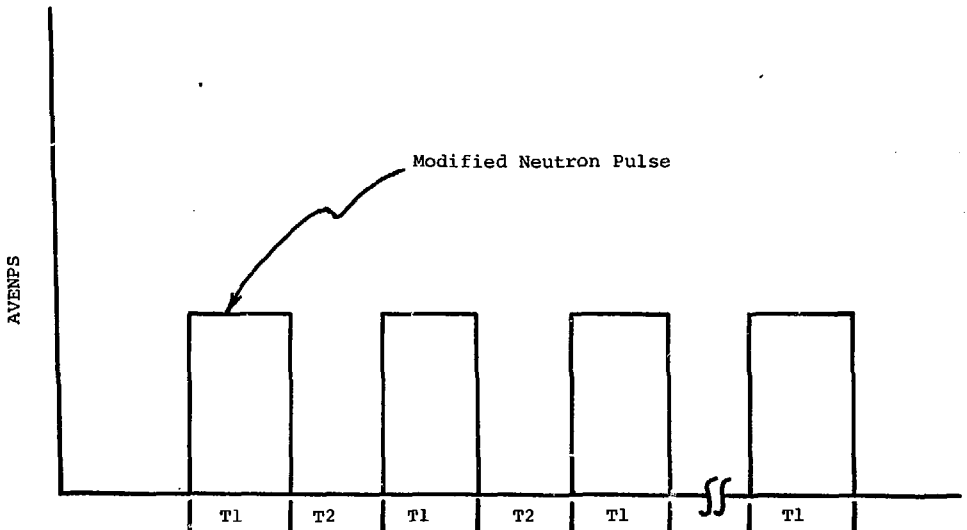
Figure 2


Theoretical Thick Target
Yield of Neutrons
Titanium Target
One to One Loading
Neutrons per Micro Coulomb
200
100
50
20
10
5
2
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
Beam Energy in KeV

is the subprogram which calculates INSNPS.

In modeling the beam pulses, INSNPS is multiplied by the fraction of the time the beam is on (the duty factor DF) to obtain the average number of neutrons produced per second during time T1, AVENPS. The short successive pulses are now represented by a single pulse (now referred to as a "test") with the same time length T1 but reduced in magnitude as shown in Figure 3. This type of pulse modeling is used throughout the project in connection with activation calculations. Typically, T1 is several hours. (T2, which represents an outage period as required by maintenance or overnight shutdowns, is also several hours in duration). The quantities, T1 and T2, and N, the total number of tests, are variables and must be specified by the user prior to execution of the program.

Figure 3



Derivation & Modification of the Activation Equation

The balance equation for the activation of target nuclides in the injector system is of the following form:

rate of change of a product nuclide = rate of production − rate of decay .

In differential form, this equation becomes

$$\frac{dB(t)}{dt} = \sigma_A \phi A - B(t)\lambda_B$$

where the variables are defined as below:

$B(t)$: number of atoms of the product nuclide B

λ_B: decay constant for the product nuclide B

σ_A: group activation cross section for target nuclide A

A: number of atoms in the target nuclide A

ϕ : group neutron flux

The solution is:

$$B(t) = \frac{\sigma_A \phi A}{\lambda_B}\left[1 - e^{-\lambda_B t}\right]$$

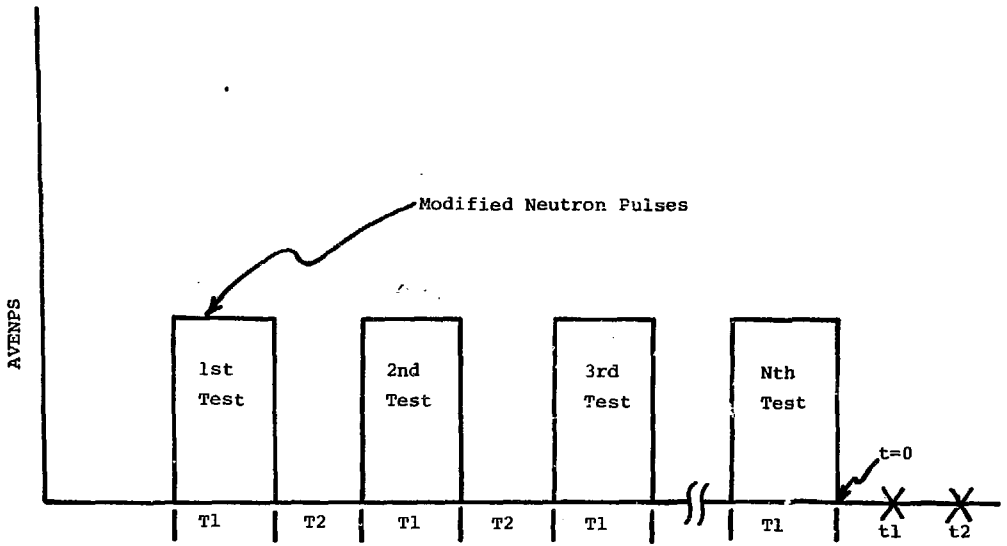
This is the basic differential equation that is used in

subsequent activation calculations. This form, however, must be algebraically modified to take into account pauses of length T2 in the injector test schedule as shown in Figure 4. T1 is the length of the test, T2 is the length of the pause, t=0 is the time at which shutdown occurs, and t1 and t2 are arbitrarily selected times after shutdown where an activity is required. N is the total number of tests. It is desired to calculate the amount of an activated nuclide at t=0 due to an arbitrary neutron testing history. The method of approach is to calculate the amount at t=0 due to each test prior to t=0. The total amount at t=0 will then be the sum of the contributions from each test. For some time t after shutdown, the amount of nuclide B will be given by:

$$B(t) = \frac{A\sigma_A\phi}{\lambda_B}\left[1 - e^{-\lambda_B \cdot T1}\right] \cdot \left[\sum_{i=1}^{N} e^{-\left\{\left[N-i\right]\left[T1+T2\right]\right\}\lambda_B}\right] e^{-\lambda_B t}$$

This expression, however, represents only 1 group of neutrons. To generalize the relation for up to n neutron groups, the calculation for B(t) above must be done for each group, that is:

Figure 4


Modified Neutron Pulses
AVENPS
1st Test
2nd Test
3rd Test
Nth Test
t=0
T1
T2
T1
T2
T1
T1
t1
t2

$$B(t) = \sum_{K=1}^{n} \frac{A\sigma_{A_K}\phi_K}{\lambda_B} \left[1-e^{-\lambda_B \cdot T1}\right] e^{-\lambda_B t} \cdot$$

$$\left[\sum_{i=1}^{N} e^{-\left\{\left[N-i\right]\left[T1+T2\right]\right\}\lambda_B}\right]$$

For the general case of the reaction A(n,b)B, the activity of B in Bq, t seconds after shutdown will be:

$$B(t)\,\lambda_B = \begin{matrix}\text{Activity of}\\ \text{Product Nuclide}\\ \text{B}\end{matrix} =$$

$$\sum_{K=1}^{n} A\sigma_{A_K}\phi_K \left[1 - e^{-\lambda_B \cdot T1}\right] e^{-\lambda_B t}$$

$$\left[\sum_{i=1}^{N} e^{-\left\{\left[N-i\right]\left[T1+T2\right]\right\}\lambda_B}\right] \qquad (1)$$

Since the amount of a target nuclide (A in this case) is entered into computer memory in terms of kilograms, the number of target atoms "A" is computed by multiplying the input mass by the quantity:

$$\frac{(1000\text{g/kg})\quad(6.023\text{X}10^{23}\ \text{atoms/mol})}{\text{atomic weight in g/mol}}$$

Equation 1 is the activation equation which is used in ACDOS1 for calculating total activities. Note that the equation is applicable to situations where the flux is constant as opposed to a "pulsing" nature by the disappearance of the second bracketed term when N=1.

Determination of the Weighting Flux & Flux Weighted Group Cross Sections

In order to average cross section data over the appropriate energy groups to acquire flux weighted group cross sections, a weighting flux is required. The flux weighted group cross section $\bar{\sigma}$, can then be evaluated from the expression:

$$\bar{\sigma} = \int_{\Delta E} \sigma(E)\, \phi(E)\, dE \Big/ \int_{\Delta E} \phi(E)\, dE$$

The problem was solved by assuming that the thermal region could be represented by a classical Maxwell-Boltzmann distribution with kT=0.025eV, intermediate energies by a 1/E function and fast groups by an exponential function, with all equations known except for arbitrary constants.

The decision to use this approach came in part from results of slowing-down theory, and, in addition, from results of the Monte Carlo calculations which produced the data for the sample problem. (The latter involves a neutral beam injector which is surrounded by thick concrete walls.) The group fluxes calculated[3] for this case are given in Table 1. To verify the above assumptions, column 3 (neutron/cm^2·eV·source neutron) was plotted as a function of the arithmetic average of the group boundaries found in column 2 for groups 1 through 19. See Figure 5.

NOTE: The convention will be adopted that Group 1 represents the highest energy group.

GP	Energy Interval (eV)	Neutrons/ $cm^2 \cdot eV \cdot$ Source Neutron	Neutrons/ $cm^2 \cdot$ Source Neutron
1	2.385E+6 to 2.307E+6	1.013E-10	7.901E-6
2	2.307E+6 to 1.827E+6	2.879E-13	1.382E-7
3	1.827E+6 to 1.108E+6	1.959E-13	8.833E-8
4	1.108E+6 to 5.502E+5	1.896E-13	1.058E-7
5	5.502E+5 to 1.576E+5	3.510E-13	1.378E-7
6	1.576E+5 to 1.111E+5	6.111E-13	2.842E-8
7	1.111E+5 to 5.248E+4	8.172E-13	4.790E-8
8	5.248E+4 to 2.479E+4	1.394E-12	3.860E-8
9	2.479E+4 to 2.188E+4	2.550E-12	7.420E-9
10	2.188E+4 to 1.033E+4	3.049E-12	3.522E-8
11	1.033E+4 to 3.355E+3	6.552E-12	4.570E-8
12	3.355E+3 to 1.234E+3	1.896E-11	4.021E-8
13	1.234E+3 to 5.829E+2	4.673E-11	3.043E-8
14	5.829E+2 to 1.013E+2	1.502E-10	7.234E-8
15	1.013E+2 to 2.902E+1	6.481E-10	4.684E-8
16	2.902E+1 to 1.068E+1	1.642E-9	3.011E-8
17	1.068E+1 to 3.059E+0	5.184E-9	3.951E-8
18	3.059E+0 to 1.125E+0	1.787E-8	3.456E-8
19	1.125E+0 to 4.140E-1	4.327E-8	3.076E-8
20	4.140E-1 to 1.000E-5	2.002E-6	8.288E-7

Table 1

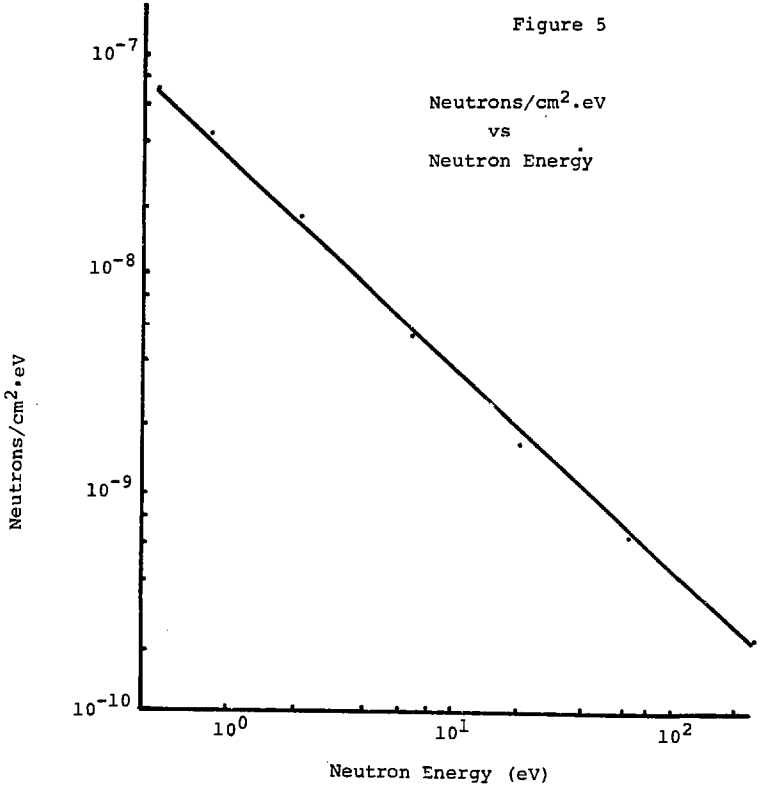


Figure 5
Neutrons/cm2.eV
vs
Neutron Energy
10-7
10-8
10-9
10-10
Neutrons/cm2.eV
100
101
102
Neutron Energy (eV)

Figure 5

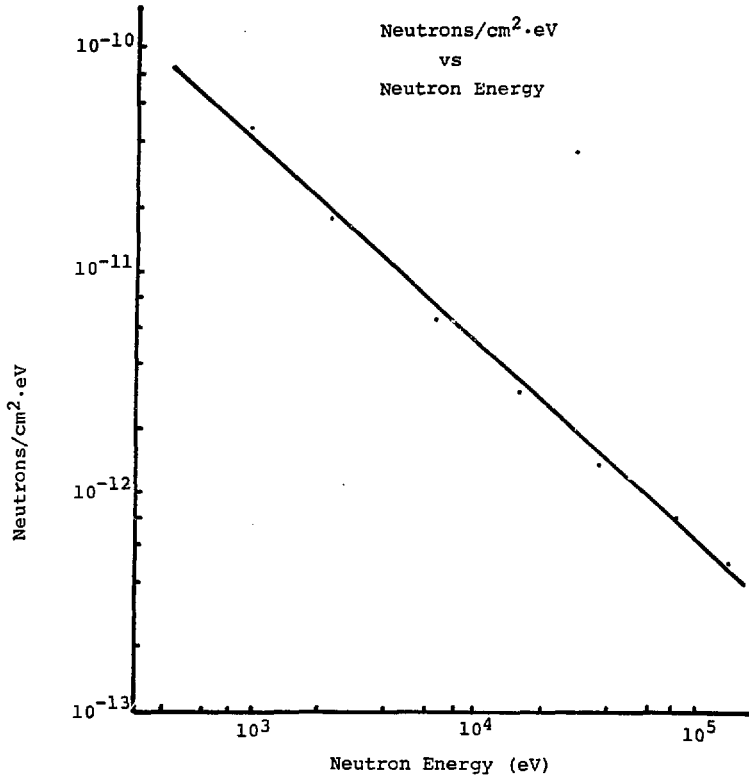
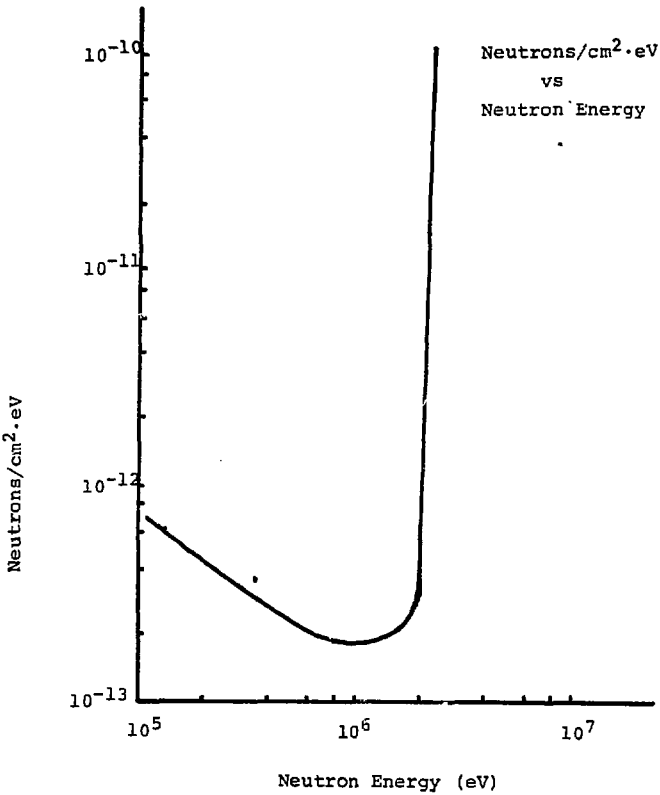

Neutrons/cm^2·eV
vs
Neutron Energy
10^{-10}
10^{-11}
10^{-12}
10^{-13}
Neutrons/cm^2·eV
10^3
10^4
10^5
Neutron Energy (eV)

Figure 5


Neutrons/cm²·eV
vs
Neutron Energy
10⁻¹⁰
10⁻¹¹
10⁻¹²
10⁻¹³
Neutrons/cm²·eV
10⁵
10⁶
10⁷
Neutron Energy (eV)

Inspection of the curve does suggest a 1/E type of behavior for groups 4 through 19 since a quantity that varies as a constant/E will plot as a straight line on log-log paper and will have a slope of -1. Calculation of the slope at various points along the line did reveal a slope close to -1. At higher energies (above $2X10^4$eV) the curve departs somewhat more from 1/E behavior and finally reaches a region just below fast energies where the curve is neither 1/E nor exponential. Even though this region just below the exponential part of the curve deviates from 1/E behavior, it is assumed, for ease of calculations, that it does behave as 1/E. Since neutrons are "born" with a narrow group of discrete energies in the injector as contrasted with a fission spectrum in a reactor, one would intuitively expect the fast region to appear more as a "spike" then a "spread out" distribution. The graph definitely illustrates this behavior and for this reason the assumption of an exponential form for the fast groups appears justified. As stated previously, the Maxwell-Boltzmann distribution is used for describing the thermal region. At this point, the forms of the equations for describing the weighting flux are known except for arbitrary constants. To evaluate the constants, each analytic expression for the weighting flux is integrated over the appropriate energy group and set equal to the numerical value of the corresponding group quantities in column 4 of Table 1, which are normalized

total group fluxes. For example, for the thermal region, group 20:

$$\int_{1.0X10^{-5}}^{4.14X10^{-1}} C\sqrt{E}\, e^{-\frac{E}{0.025}}\, dE = 8.288X10^{-7}$$

therefore,

$$C = 2.36 \times 10^{-4}$$

For an intermediate neutron group, group 17

$$\int_{3.059}^{10.68} C/E \quad dE = 3.951X10^{-8}$$

therefore,

$$C = 3.160 \times 10^{-8}$$

The determination of the constants associated with fast neutron groups, however, is somewhat more involved. Since the assumed form for the fast exponential is

$\phi(E) = Ae^{\frac{E1-E2}{B}}$ where E1 and E2 are knowns and E2 is the highest energy of that group, two equations are required to evaluate the two constants. One equation results from the usual integral condition that

$$(2) \quad \int_{E1}^{E2} Ae^{\frac{E1-E2}{B}} dE = \text{total flux for that group}$$

while the other condition results from a continuity of flux requirement at the boundary of the two groups. Equation 2 already represents one equation. A second equation resulting from the continuity requirement is:

$$C/E1 = Ae^{\frac{E1-E2}{B}}$$

where C is a known constant from the integral condition placed on the last 1/E group. (Analysis is always done starting with the lowest energy group). See Figure 6. With two equations and two unknowns, the constants can be determined by solving the resultant transcendental equations. If there is another fast group, the same method is applied. One equation results from the required integral condition on that group, and a second equation results from a continuity of flux requirement at the boundary with the preceding group. The result of this method is that the first exponential group is always matched to the last 1/E group, and any additional fast groups are always matched to the preceding fast groups. In the case of only two groups, one thermal and one fast, a match is required at the boundary between the groups to allow evaluation of the two constants in the assumed exponential function following the Maxwellian. For 1/E groups, only one equation is re-

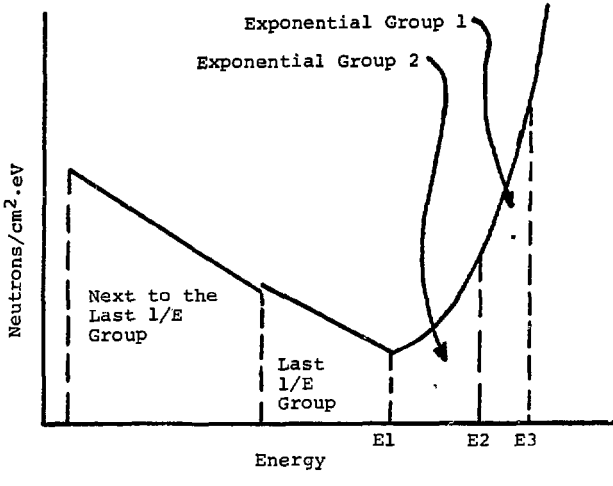


Figure 6

A continuity of flux requirement is used at energy E1 to obtain another equation required to evaluate the two unknown constants in the assumed exponential function for group 2. This equation is:

$$C/E1 = Ae^{\frac{E1-E2}{B}}$$

See appendix A for the forms and solutions to these transcendental equations.

quired as there is only one constant to be evaluated. Since 1/E constants are calculated solely from the integral condition placed on the equation for that group, discontinuities sometimes result in the flux at the group boundary between two adjacent 1/E groups or between the thermal and the first 1/E group. This is not a serious problem because the discontinuities are not significantly large. The important consideration is that the integral of the weighting flux function over the group be equal to the total flux for that group so that the correct averaging of the flux over that same interval is maintained. Table 2 shows the results of this approach to the weighting flux function determination. The integral of these functions over their proper energy range always results in the numerical value of the total flux for that group as it should. A plot of the data in Table 2 (Groups 1-19) is shown in Figure 7. As previously mentioned, slight flux discontinuities at boundaries between 1/E groups exists. Note that the discontinuities are greater in the region just below the two fast groups. This is not surprising since that region deviates somewhat from 1/E but was assumed to be 1/E in weighting flux calculations. Comparison of Figure 5 with Figure 7 shows a close resemblance as one would definitely want and expect.

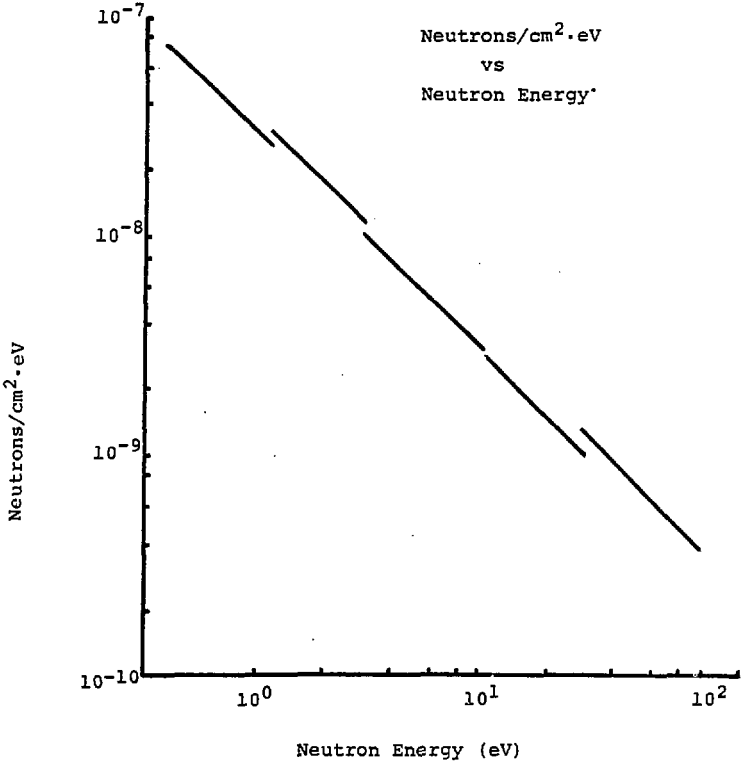
During the execution of ACDOS1, subroutine WTFLUX will determine the values of the constants associated with these

Table 2

Flux Densities for the Sample Problem

Group	Group Boundaries (eV)	Equation $\phi(E)$
20	1.0×10^{-5} to 0.414	2.36×10^{-4} Eexp $(-\frac{E}{0.025})$
19	0.414 to 1.125	$3.077\times10^{-8}E^{-1}$
18	1.125 to 3.059	$3.455\times10^{-8}E^{-1}$
17	3.059 to 10.68	$3.160\times10^{-8}E^{-1}$
16	10.68 to 29.02	$3.012\times10^{-8}E^{-1}$
15	29.02 to 101.3	$3.747\times10^{-8}E^{-1}$
14	101.3 to 582.9	$4.134\times10^{-8}E^{-1}$
13	582.9 to 1.234×10^{3}	$4.057\times10^{-8}E^{-1}$
12	1.234×10^{3} to 3.355×10^{3}	$4.020\times10^{-8}E^{-1}$
11	3.355×10^{3} to 1.033×10^{4}	$4.064\times10^{-8}E^{-1}$
10	1.033×10^{4} to 2.188×10^{4}	$4.693\times10^{-8}E^{-1}$
9	2.188×10^{4} to 2.479×10^{4}	$5.942\times10^{-8}E^{-1}$
8	2.479×10^{4} to 5.248×10^{4}	$5.147\times10^{-8}E^{-1}$
7	5.248×10^{4} to 1.111×10^{5}	$6.387\times10^{-8}E^{-1}$
6	1.111×10^{5} to 1.576×10^{5}	$8.129\times10^{-8}E^{-1}$
5	1.576×10^{5} to 5.502×10^{5}	$1.102\times10^{-7}E^{-1}$
4	5.502×10^{5} to 1.108×10^{6}	$1.511\times10^{-7}E^{-1}$
3	1.108×10^{6} to 1.827×10^{6}	$1.666\times10^{-7}E^{-1}$
2	1.827×10^{6} to 2.307×10^{6}	6.62×10^{-13}exp $(E-2.307\times10^{6}/2.42\times10^{5})$
1	2.307×10^{6} to 2.385×10^{6}	7.07×10^{-10}exp $(E-2.385\times10^{6}/11,184)$

Figure 7


Neutrons/cm^2·eV
vs
Neutron Energy
10^{-7}
10^{-8}
10^{-9}
10^{-10}
Neutrons/cm^2·eV
10^0
10^1
10^2
Neutron Energy (eV)

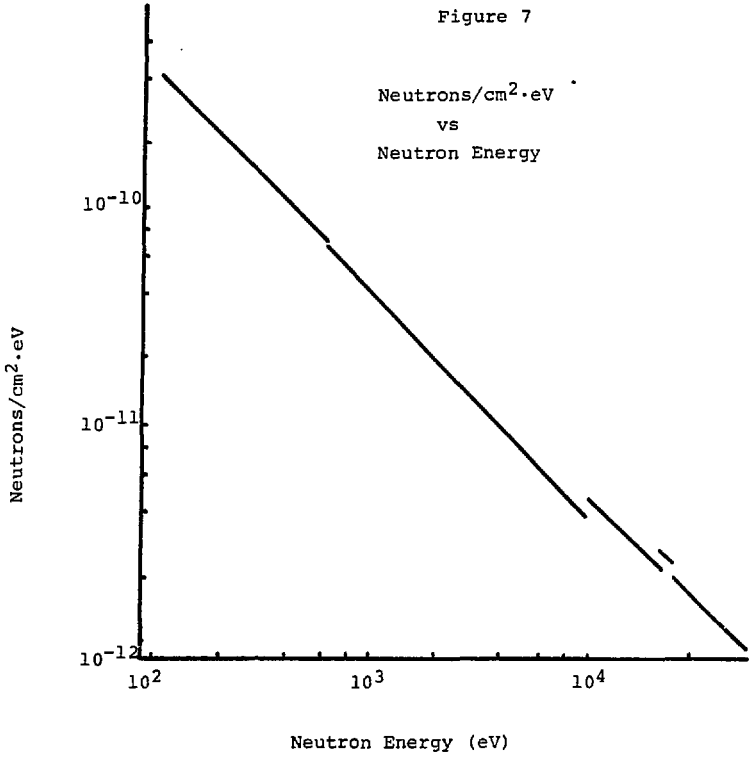


Figure 7
Neutrons/cm2·eV
vs
Neutron Energy
10-10
10-11
10-12
Neutrons/cm2·eV
102
103
104
Neutron Energy (eV)

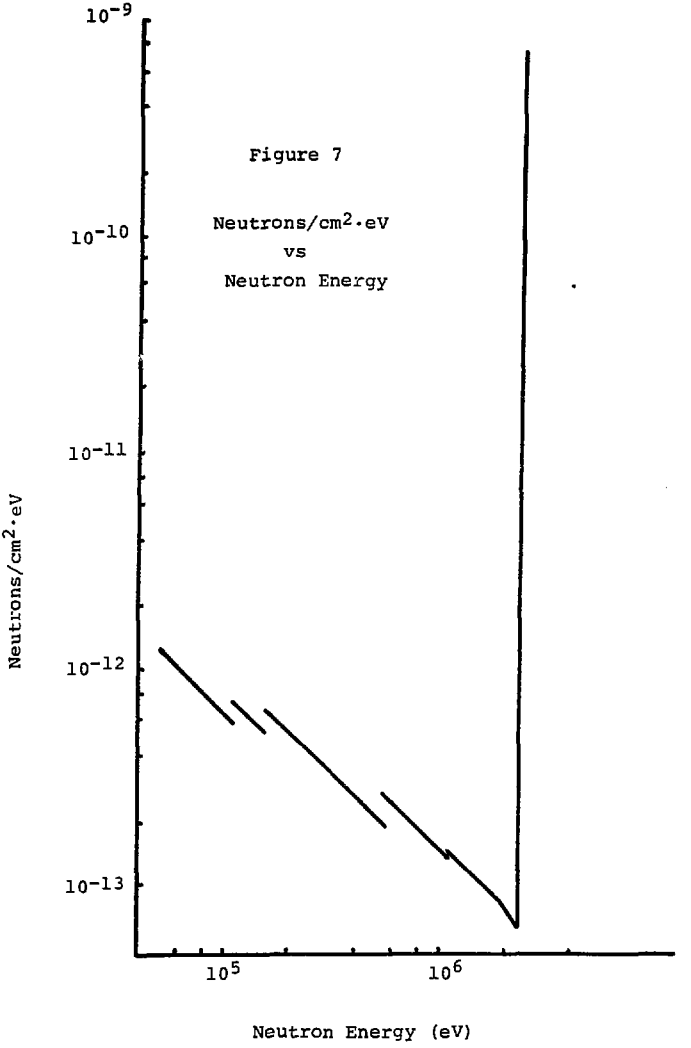


Figure 7
Neutrons/cm2·eV
vs
Neutron Energy
10-9
10-10
10-11
10-12
10-13
Neutrons/cm2·eV
10^5
10^6
Neutron Energy (eV)

assumed functions. The particular forms of the weighting functions chosen depend upon the number of neutron groups in any given activation problem as follows:

Number of Neutron Groups	Group Energy Designation	Form(s) of Weighting Function(s)
1	Thermal	Maxwellian
2	1 Thermal 1 Fast	Maxwellian Exponential
3	1 Thermal 1 Intermediate 1 Fast	Maxwellian 1/E Exponential
4 to 50	1 Thermal 1 or more Intermediate 1 or more Fast	Maxwellian 1/E Exponential

For four or more neutron groups, the user must specify the number of fast groups, NF, so that the correct apportionment of 1/E groups can be made.

With the weighting flux function known for each group, the flux weighted group cross section can be calculated from the expression:

$$\bar{\sigma} = \int_{\Delta E} \sigma(E)\,\phi(E)\,dE \Big/ \int_{\Delta E} \phi(E)\,dE$$

where ϕ (E) is the previously determined weighting flux function. During execution of the subroutine that calculates activation, ACTVAT, the cross section information for the target nuclide in question is read from the ACTLMFE library, which is a subset (without fission cross sections) of the ACTL library.[4] The data consist of two numbers, an energy in MeV and an associated cross section for that energy in barns. To calculate $\bar{\sigma}$, a subroutine, SMOOTH, first merges the group boundary values into the energy-cross-section pair data and performs a linear interpolation to calculate the cross section at the group boundaries.[5] The result is two arrays, one with energy values and the other with cross section values. There is always a one to one correspondence. See Figure 8 (less caption). GP(J) and GP(J+1) are arbitrary group boundaries which initially had no cross section value associated with them. The value is obtained by a linear interpolation using the first value on each side of a group boundary assuming that cross sections are linearly interpolable in energy. For two arbitrary cross sections, σ(I) and σ(I+1) whose values lie within the group boundaries GP(J) and GP(J+1), an analytical expression is found for the line connecting them, using the common two point formula:

$$Y-Y_1 = \frac{Y_2-Y_1}{X_2-X_1} \quad (X-X_1)$$

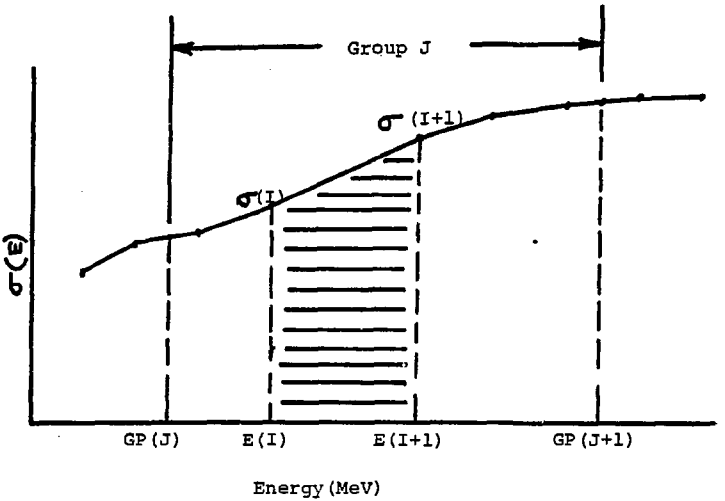


Figure 8

Straight lines are connected between two cross section points using the two point formula. The resulting equation for the line is used as the analytical expression for $\sigma(E)$ between E(I) and E(I+1). $\sigma(E)$ is used in the evaluation of

$$\Delta \bar{\sigma} = \int_{E(I)}^{E(I+1)} \sigma(E)\, \phi(E)\, dE \Bigg/ \int_{E(I)}^{E(I+1)} \phi(E)\, dE$$

where $\Delta \bar{\sigma}$ is an arbitrary fraction of the total flux weighted group cross section, $\bar{\sigma}$, for the group in question.

In this case y is $\sigma(E)$ and x is energy E in MeV. The equation takes the form:

$$\sigma(E) - \sigma(I) = \frac{\sigma(I+1) - \sigma(I)}{E(I+1) - E(I)} \left[E - E(I) \right]$$

so that $\sigma(E)$ for $\sigma(I) \leq E \leq \sigma(I+1)$ is

$$\sigma(E) = \left\{ \frac{\sigma(I+1) - \sigma(I)}{E(I+1) - E(I)} \left[E - E(I) \right] + \sigma(I) \right\} .$$

See Figure 8.

Now that an analytical expression exists for $\sigma(E)$ and $\phi(E)$, the product of these two functions is numerically integrated between E(I) and E(I+1) and the value kept in a running sum. This process is continued, point by point, until the upper boundary is reached. The particular form of $\phi(E)$ used depends upon the energy range in which the specific group boundaries lie. For example, in the thermal region where GP(J) = 1×10^{-5} eV and GP(J+1) = 0.414 eV, $\phi(E) = 2.360 \times 10^{-4} \sqrt{E} \exp(-(E/0.025)$ from Table 1, so that the expression for $\sigma(E)\,\phi(E)$ is:

$$\left\{ \frac{\sigma(I+1) - \sigma(I)}{E(I+1) - E(I)} \left[E - E(I) \right] + \sigma(I) \right\}$$

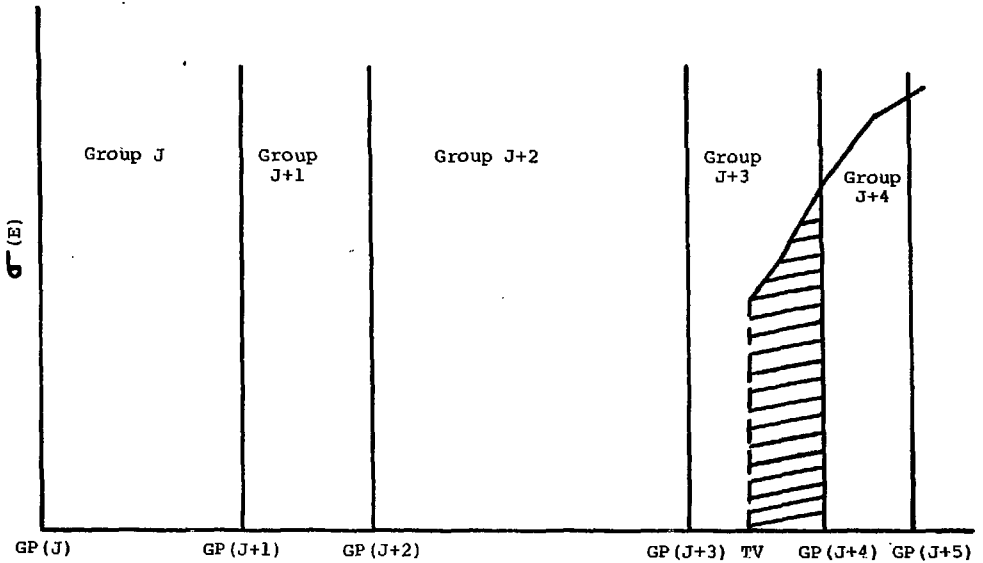
$$\left\{ 2.360 \times 10^{4} \sqrt{E}\, e^{-\frac{E}{0.025}} \right\}$$

It is this expression that is numerically integrated between each thermal E(I) and E(I+1). When all numerical

integrations are completed for a specific group, the running sum is divided by the total flux for that group. The quotient, $\bar{\sigma}$, is the desired quantity--the flux weighted group cross section for that group. This entire process is repeated until all energy groups have been addressed. Several test cases have been run to acquire results for comparison purposes with another averaging program which originated at LLNL.[5] The values obtained by this method are remarkably close to those obtained by the LLNL code for fast and intermediate neutron groups. For the thermal region, however, the above mentioned procedure produced better results than the LLNL code.

Two further comments are in order considering the method of averaging cross sections. First, for the case of nuclear reactions which are threshold oriented, a zero is substituted for each group cross section whose energy range is below the threshold energy. Calculations begin only when the energy at which a cross section value was measured equals or exceeds the threshold energy. Typically, that first cross section value is located somewhere within the group boundaries at which the calculations begin. The contribution to that particular group cross section comes solely from evaluation of the pertinent quantities for energies greater than or equal to the first cross section value and less than or equal to the first encountered group boundary. This is shown in Figure 9.

Figure 9


σ (E)
Group J
Group J+1
Group J+2
Group J+3
Group J+4
GP (J)
GP (J+1)
GP (J+2)
GP (J+3)
TV
GP (J+4)
GP (J+5)

In this example, the reaction has a threshold located between the arbitrary group boundaries, GP(J+3) and GP(J+4). The group cross section for group (J+3) results strictly from the evaluation of the pertinent quantities from the threshold value, Tv, to the first encountered group boundary, GP(J+4). Even if there is only one cross section value located in group (J+3), the calculation proceeds since there is a cross section value associated with GP(J+4) as a result of the initial merging of group boundary energies into the energy-cross-section data. In any case, however, zeros are substituted for group cross sections in group (J) through (J+2) in accordance with the above discussion.

The second comment concerns the calculation of the thermal group cross section. Since no provision is made for interpolating cross section values at the first group boundary, averaging calculations begin at the first encountered cross section value above the energy of the first group boundary as shown in Figure 10. As a result, a very small portion of the Maxwellian (generally below 1.15×10^{-10} eV) is not accounted for. However, this is unimportant since the contribution from this part of the Maxwellian is exceedingly small.

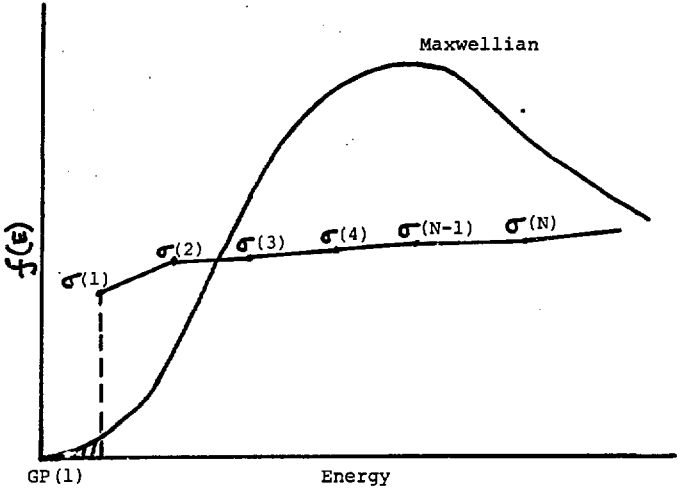


Figure 10

Program to Calculate Activities

The fourth major step in the solution of the problem required writing a standard FORTRAN program to calculate activities given the necessary information. The code is written in a structured format form using one executive program to call up a number of subroutines. Variably dimensioned arrays are used where possible to conserve on memory. The name and purpose of each subroutine is discussed below.

1) INPUT: for entering all "variable" input data into computer memory. Error traps are provided to check input data. All data is printed out for user verification.

2) ARAYIN: for entering all "array" data. Array data is printed out for user verification.

3) SOURCE: given the current, voltage, beam fraction, and duty factor, calculates the instantaneous and average number of source neutrons produced per second during a test. This subroutine is bypassed if the user wishes to use his own source term.

4) GPFLUX: uses the calculated values of the average number of source neutrons

produced per second during a test or the user supplied source term and calculates the average flux for each group by taking the product of the unity-normalized total group flux and source neutron term.

5) WTFLUX: determines the constants for the assumed weighting flux functions by applying integral and/or group boundary constraints to the pertinent equations. Also prints out the integrals of the weighting flux functions over their appropriate energy intervals for user verification.

6) AVRAGE: determines an analytical expression for $\sigma(E)$ over a specific ΔE and numerically integrates the product of $\sigma(E)$ and $\phi(E)$, the weighting function, over the group energy interval. It then takes the sum of the integrated products over the unity normalized total flux for the same group to calculate the flux weighted group cross section, $\bar{\sigma}$.

7) SMOOTH: first determines if the energy associated with the first energy-cross-section pair read from the ACTLMFE library

is below the greatest group boundary. If so, the routine then merges the group boundary energies into the energy-cross-section data read from the ACTLMFE library and linearly interpolates to find the values of the cross sections at the group boundaries. If not, the particular reaction is skipped and the next one considered.

8) POSITN: for positioning the file marker in the ACTLMFE library.

9) POSIT2: same as POSITN.

10) ACTVAT: takes the following parameters and calculates activities according to Equation 1.

a) number of tests

b) length of the tests

c) length of the pauses

d) number of kilograms of a particular nuclide

e) specific times after shutdown

f) average group fluxes

g) flux weighted group cross sections

The result of each activation calculation is printed out. This allows the user to determine what reaction is the most significant for a particular target nuclide.

Moreover, two running sums are maintained in order to print out the total activity from the activation of a particular target nuclide which is subject to many different reactions and, in addition, the total system activity when all target nuclides have been addressed. If, in the case of threshold reactions, the energy read from the first energy-cross-section pair is greater than the highest group boundary, the reaction is bypassed and interrogation continues. Only one pass through the entire ACTLMFE library is required since the nuclide data is arranged in order of increasing ZA (1000Z+A) as is the user specified target nuclide information. The accuracy of the calculations, of course, depend directly on the reliability of the cross section data and the flux calculations. At the termination of subroutine ACTVAT, all activation results will have been stored in memory for future dose rate calculations.

Program To Calculate Dose Rates

The fifth major and final step required writing a program to calculate dose rates as a function of geometry and time after shutdown. This subprogram, DOSRTE, accomplishes the desired calculation by performing essentially four major tasks:

1) interrogation of the decay library, LEVDEC, which is a subset of ENSL[6], for the energy and multiplicity associated with each applicable gamma ray produced by a specific nuclide,
2) determination of an effective particle flux that yields a unit absorbed dose in soft tissue,
3) calculation of the dose rate using the previously calculated activities, user specified geometry, applicable multiplicities, and appropriate effective particle fluxes, and
4) interrogation of the LEVDEC library, on a second pass, for radioactive daughters.

Provisions are made in the program for keeping two running sums of dose rate values, one for a particular target nuclide and one for the system as a whole. The first allows the user to determine which of the particular target nuclides is contributing to the highest dose rate values. The second, of course, gives the dose rate for the entire system. After all parent and daughter

nuclides have been addressed, the results are printed out and the ACDOS1 object program terminates execution.

Prior to performing the calculations, parent ID numbers are rearranged in order of increasing ZA, 1000 Z+A, to match the format of nuclides listed in the library. The associated target ID numbers and parent activities and half-lifes are also rearranged at the same time to maintain the correct association of data. All rearrangement occurs automatically during the execution of DOSRTE and is a prerequisite for efficient interrogation of the library.

Since the same nuclides (parents or daughters) may result from two different target nuclides by different reactions, a temporary holding array is utilized to store the decay information when two or more successive parents have the same ID numbers. This allows repeated interrogation of the information stored in the holding array when the routine finds parents with the same ID number. The use of the array in conjunction with the initial rearrangement of parent ID numbers makes it possible to address all parents on one pass through the LEVDEC library. Dose rates due to each parent are calculated and then subsequently stored in memory with the correct target affiliation prior to addressing any potential radioactive daughters.

After calculating dose rates for applicable parents, radioactive daughters are considered. Since the

identity of any daughters is not known in advance, a list of such potential candidates (read from the LEVDEC library) is compiled and stored in memory with the proper parent ID number and parent activity affiliation. After all parents have been addressed on the first pass through LEVDEC, and the resulting dose rates calculated, a second pass is made to search the library for the previously compiled daughters. Since the same interrogation methodology is employed as with the parents, only one pass through the library is required.

After finding a particular nuclide in the decay library, parent or daughter, the first consideration prior to calculating any dose rates is the determination of an effective particle flux that yields a unit absorbed dose in soft tissue. For each applicable gamma ray energy read from the LEVDEC library, an effective particle flux is calculated. This calculation was accomplished by first fitting 13 curves to the data[7] shown in Table 3. Twelve of the curves are of the form ax^b and one of the form ae^{bx}. A large number of curves were chosen so as to accurately reproduce the data over such a wide energy range. (As a result, all coefficients of determination arising from the curve fitting process are in excess of 0.99.) Table 4 shows the results of the curve fittings. The appropriate equation for calculating the effective particle flux, FLUXE, is chosen according to the value

E (MeV)	Dose Rate of 2.5 mrad/h corresponds to ϕ_o QUANTA/cm^2·S
0.01	956
0.015	2,310
0.020	4,320
0.030	9,980
0.040	17,400
0.050	23,100
0.060	25,200
0.080	23,200
0.100	18,600
0.150	10,800
0.300	5,020
0.400	3,660
0.500	2,920
0.600	2,440
0.800	1,880
1.00	1,550
1.25	1,330
1.50	1,130
2.0	912
3.0	686
4.0	559
5.0	480
6.0	420
8.0	339
10.0	284

Table 3

Energy Range (MeV)	Particle Flux Equation
0.010 to 0.020	FLUXE=21495330.64*E** 2.1759447
0.020 to 0.040	FLUXE=11460233.22*E** 2.0136022
0.040 to 0.050	FLUXE=1036918.975*E** 1.2698653
0.050 to 0.060	FLUXE=96497.38032*E** 0.4772399
0.060 to 0.080	FLUXE=11225.12221*E** (-0.2874410)
0.080 to 0.150	FLUXE=55482.78326*EXP (E*(-10.9132069))
0.150 to 0.400	FLUXE=1330.569205*E** (-1.1036085)
0.400 to 0.600	FLUXE=1462.198966*E** (-1.0004513)
0.600 to 1.000	FLUXE=1546.997798*E** (-0.8891058)
1.000 to 1.500	FLUXE=1549.091320*E** (-0.7797676)
1.500 to 3.000	FLUXE=1508.205757*E** (-0.7188784)
3.000 to 6.000	FLUXE=1488.362190*E** (-0.7051622)
6.000 to 10.000	FLUXE=1656.852874*E** (-0.7649757)

Table 4

of E read from the decay library for a particular gamma ray.

The second consideration is that of geometry. Three choices are possible. The first is a point source approximation given by the following equation for the dose rate:

$$\text{DOSRAT} = \frac{S_o}{4\pi D^2} \quad .$$

S_o is the source strength divided by the flux-per-unit-dose-rate as given by the formula

$$S_o = \frac{\text{MULT} \times 2.5 \times \text{ACT}}{\text{FLUXE}}$$

where MULT is the multiplicity of the gamma ray, and FLUXE is the effective photon flux corresponding to a dose rate of 2.5 mrem/h in soft tissue. D is the radial distance in cm to the point where the dose rate is desired. The second choice is that of cylindrical-on-axis geometry subject to the condition that the cylinder be non-absorbing. The equation[9] that is used to calculate the dose rate as a function of the distance from one end is:

$$\text{DOSRAT} = \frac{S_V}{4}\left\{ (HT+D)\left[\ln\left(1+\frac{R^2}{(HT+D)^2}\right) + \frac{2R}{HT+D}\,\text{ARCTAN}\,\frac{HT+D}{R}\right] - D\left[\ln\left(1+\frac{R^2}{D^2}\right) + \frac{2R}{D}\,\text{ARCTAN}\,\frac{D}{R}\right]\right\}$$

where S_V is the ratio of S_o to the volume in cm^3, D is the distance in cm, and R and HT are the radius and height in cm. The third and final choice, is that of spherical geometry also subject to the condition that the sphere be non-absorbing. The equation[9] is:

$$\text{DOSRAT} = \frac{S_V}{4d}\left[2Rd - (d^2-R^2)\ \ln\left(\frac{R+d}{d-R}\right)\right]$$

where S_V is defined as above, R is the radius in cm, and d=D+R is the distance from the center of the sphere in cm.

In addition to the constraint that the cylinder and sphere be non-absorbing, the following assumptions apply to all dose rate calculations:

1) any potential radioactive daughters result only from β^- decay,
2) the daughter decays from the ground state,
3) the daughter radiations appear coincidently with those of the parents, i.e. have zero half lives,
4) the daughter of the daughter is stable, and
5) dose rates due to parents and daughters with half-lives less than one second can be neglected.

After all parents and radioactive daughters have been located and their contributions determined, the results are printed out starting with the lowest target ID number and progressing upward in ascending order. For each target nuclide, the dose rate is printed out for each reaction that occurs and then, for the target as a whole. This is, of course, just the sum of the dose rates due to all of the reactions generated by that target nuclide. Finally, after all target nuclides are addressed, the dose rates for the entire system are printed out and the ACDOS1 object program terminates execution.

Code Description

1. ACTLMFE - Description
2. LEVDEC - Description
3. Code Input Description for ACDOS1 Data Deck
4. Code Output Description for ACDOS1
5. Program Flowchart
6. Subroutine Level Chart
7. Description of ACDOS1 Tape Structure

ACTLMFE Description

The ACTLMFE library contains 20,177 lines. The format of the file is:

Record	Column		Format
1	1-6	ZA (1000Z+A)	I6
	7-13	Ignore	7X
	14-24	Target Mass (amu)	E11.4
	25-35	Ignore	11X
	36-46	Level of Target (MeV)	E11.4
	47	Ignore	1X
	48-58	Target Half-Life(s)	E11.4
2	1-2	Reaction ID Number	I2
	3-8	Ignore	6X
	9-20	Q Value (MeV)	E12.4
	21-32	ZA of Product	E12.4
	33-44	Level of Product (MeV)	E12.4
	45-56	Product Half-Life(s)	E12.4
3	1-3	Number of Energy-Cross-Section Pairs (NP)	I3
3 et seq	1-66	(EN(K), CS(K), K=1, NP) where EN is energy in MeV and CS is cross section in barns	6E11.4
After the last line of EN, CS pairs	72	Reaction Separator Sentinel	71X, I1

The above pattern is repeated for each reaction. The last reaction in ACTLMFE is for U^{240}, (N,γ).

LEVDEC Description

The LEVDEC library contains 62,221 lines. The format for the first record of each set is:

Parameter	Format
ZA(1000Z+A)	I6
Level (MeV)	E11.4
Parity	F4.1
Spin	F5.1
Half-Life(s)	E11.4
Number of Decay Modes to Follow (NDMODE)	I3

2nd et seq. records to NDMODE

Parameter	Format
Blank	38X
Mode of Decay	I2
ZA of the Daughter	I7
Level of Daughter	E11.4
Probability of Decay to that Level	E12.4

Two comments are in order concerning the LEVDEC library. First, of the twelve possible modes of decay shown on Table 5, only three are considered in dose rate calculations, γ, β^+, and β^-.

Decay Identifier	Mode of Decay
1	neutron
2	proton
3	deuteron
4	triton
5	He^3
6	α
7	γ
8	β^+
9	β^-
10	EC
18	unresolved EC+β^+
99	no decay-stable ground state

Table 5

During the interrogation of the LEVDEC library for a particular parent or radioactive daughter nuclide, subroutine DOSRTE will check the decay identifier. If the identifier is a 7, 8, or 9, the reaction is considered. If not, the reaction is skipped and interrogation continues. Secondly, when the identifier is 8, (which is β^+ decay) two annilihation γ's of 0.511 MeV each will automatically be included in the dose rate calculations with the appropriate multiplicities.

Code Input Description for ACDOS1 Data Deck

Card No.	Variable Description	Columns	Format
1	Any information to be printed out regarding problem	1-80	10(A8)
2	A, V, F, DF, T1, T2, N: current (amperes), voltage (KV), fraction of beam that is monatomic, duty factor, length of test (h), length of pause (h), number of tests. NOTE: If the user specifies his own neutron source term, A, V, F, and DF will not be needed. In this case:	1-75	(6(E10.3, 2X), I3)
2	T1, T2, N: length of test (h), length of pause (h), number of tests.	1-27	(2(E10.3, 2X), I3)
3	NOEGPS-number of energy groups	1-2	I2
4	NOPAS-number of points after shutdown	1-2	I2
5	R, D, HT, IGEOM: radius of sphere or cylinder (m), distance from point source or surface of sphere or cylinder (m), height of cylinder (m), geometry designation: 1-point source, 2-spherical, 3-cylindrical-on-axis.	1-37	(3(E10.3, 2X, I1)
6	NONUCL-number of nuclides, 30 maximum	1-2	I2
7	NF-number of fast neutron groups; use 0 if there are none	1-2	I2

8	(BFLUX(I), I=1, NOEGPS: unit normalized input fluxes	1-72	(6(E11.4, 1X))
9	(STAFS (I), I=1, NOPAS): specific time after shutdown (h)	1-72	(6(E10.3, 2X))
10 to as many as needed	ZNAME (I), IDNO (I), MASS (I): nuclide name, ID number-1000Z+A, mass of nuclide (kg). Maximum of 30 target nuclides per run	1-25	(A8, 1X, I5, 1X, E10.3)
from above to as many as needed	(GP (I), I=1, NG): group boundaries in MeV in ascending order	1-72	(6(E10.3, 2X))

Code Output Description for ACDOS1

Prior to the execution of activation calculations, all input information is printed out for user verification under the following heading: "THE FOLLOWING DATA HAS BEEN ENTERED INTO MEMORY." This information is:

AMPERES KILO-VOLTS BEAM FRACTION DUTY FACTOR
T1 T2 N

NUMBER OF ENERGY GROUPS-NOEGPS

NUMBER OF TARGET NUCLIDES-NONUCL

NUMBER OF POINTS AFTER SHUTDOWN-NOPAS

RADIUS R OF SPHERE OR CYLINDER (M)

HEIGHT HT OF CYLINDER (M)

DISTANCE D FROM POINT SOURCE OR SURFACE OF SPHERE OR CYLINDER (M)

NUMBER OF GROUP BOUNDARIES-NG

NUMBER OF FAST NEUTRON GROUPS-NF

SPECIFIC TIMES AFTER SHUTDOWN (H)

UNIT NORMALIZED FLUXES

GROUP BOUNDARIES (MEV)

NAME ID-NUMBER MASS(KG)

Quantities calculated by ACDOS1 are printed out under the heading: "THE FOLLOWING HAS BEEN CALCULATED BY ACDOS1." The quantities are:

INSTANTANEOUS NUMBER OF NEUTRONS PRODUCED PER SECOND

AVERAGE NUMBER OF NEUTRONS PRODUCED PER SECOND

INTEGRALS OF THE FITTING FUNCTION OVER THE GROUP INTERVALS

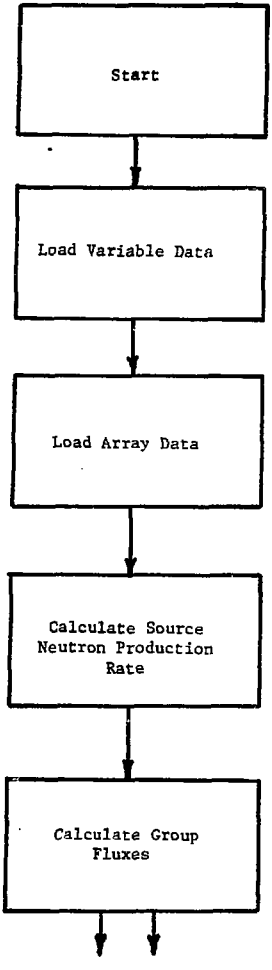
Calculated activities are printed out under the general heading: "THE FOLLOWING OUTPUT DATA ARE CALCULATED ACTIVITIES IN BQ." Three sub-headings are used to display the activation results. The first is: "TARGET PRODUCT ACT (T1) ACT(T2)...ACT(T12)." The second is: "TARGET ACT (T1) ACT(T2)...ACT(T12)." The third is: "SYSTEM ACT (T1) ACT(T2)...ACT(T12)." Above each of these three sub-headings, the times after shutdown in hours, are printed out to expedite review of the activation results. Calculated dose rates are printed out in exactly the same format as activation results. They appear under the general heading: "THE FOLLOWING OUTPUT DATA ARE CALCULATED DOSE RATES IN MRAD/H." The corresponding three sub-headings are:

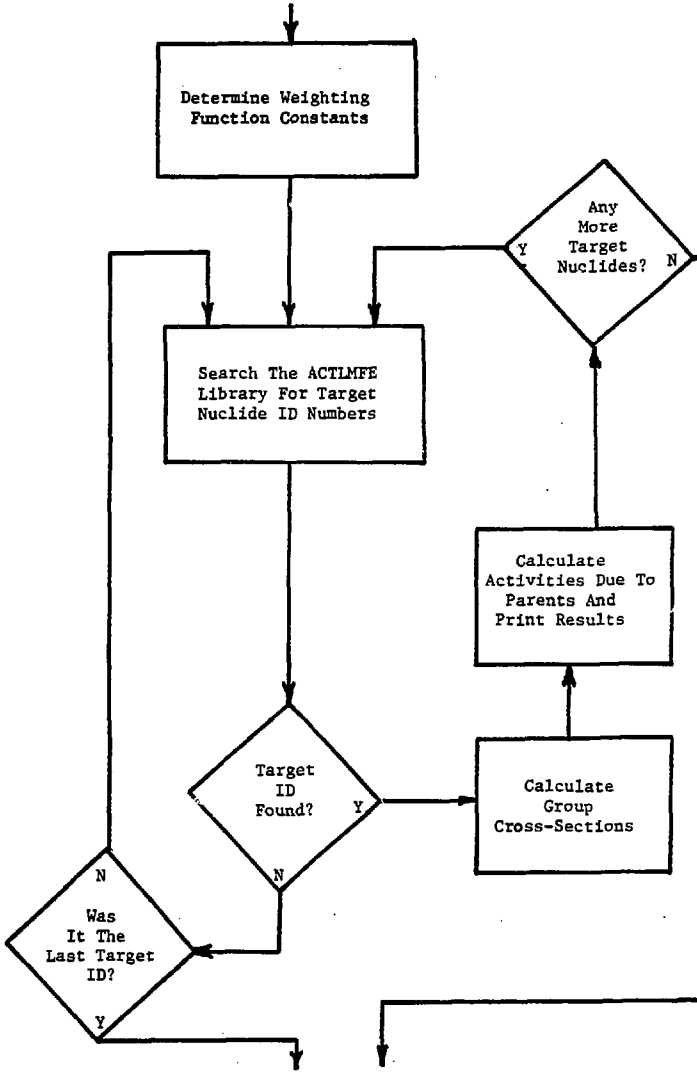
1) "TARGET PRODUCT DSR(T1) DSR(T2)...DSR(T12)"

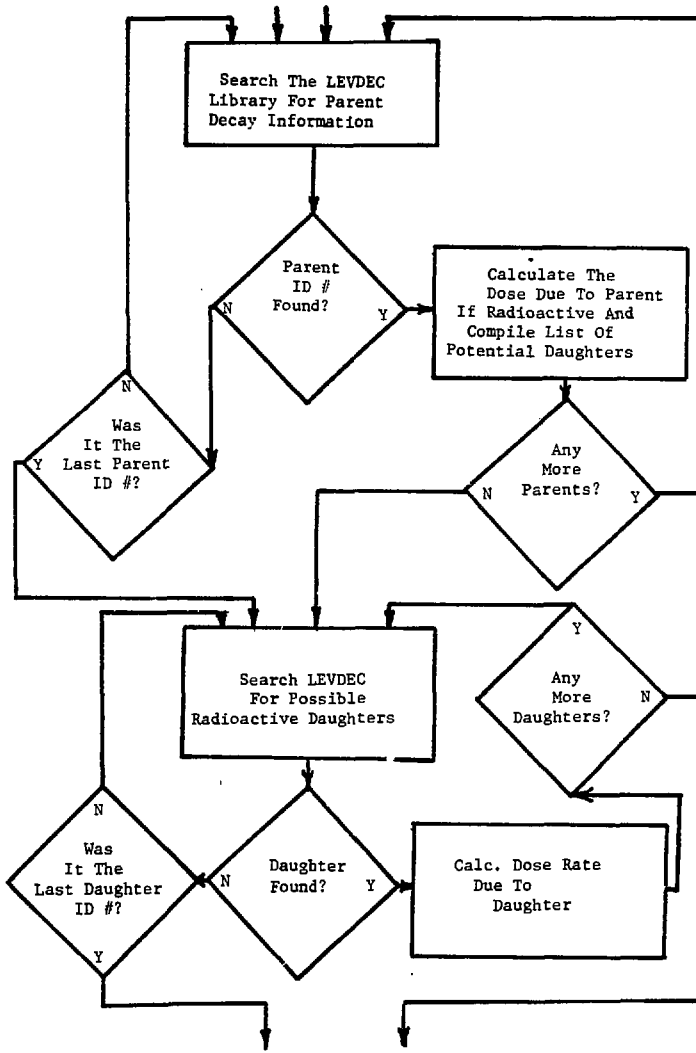
2) "TARGET DSR(T1) DSR(T2) DSR(T3)...DSR(T12)"

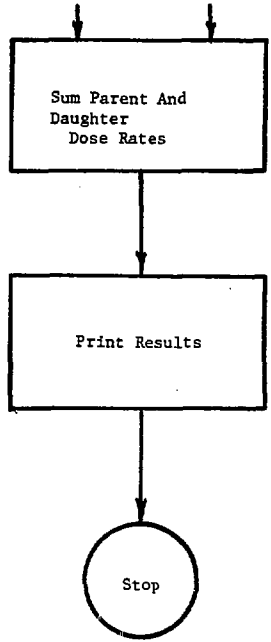
3) "SYSTEM DSR(T1) DSR(T2) DSR(T3)...DSR(T12)"

General Program Flow Chart*

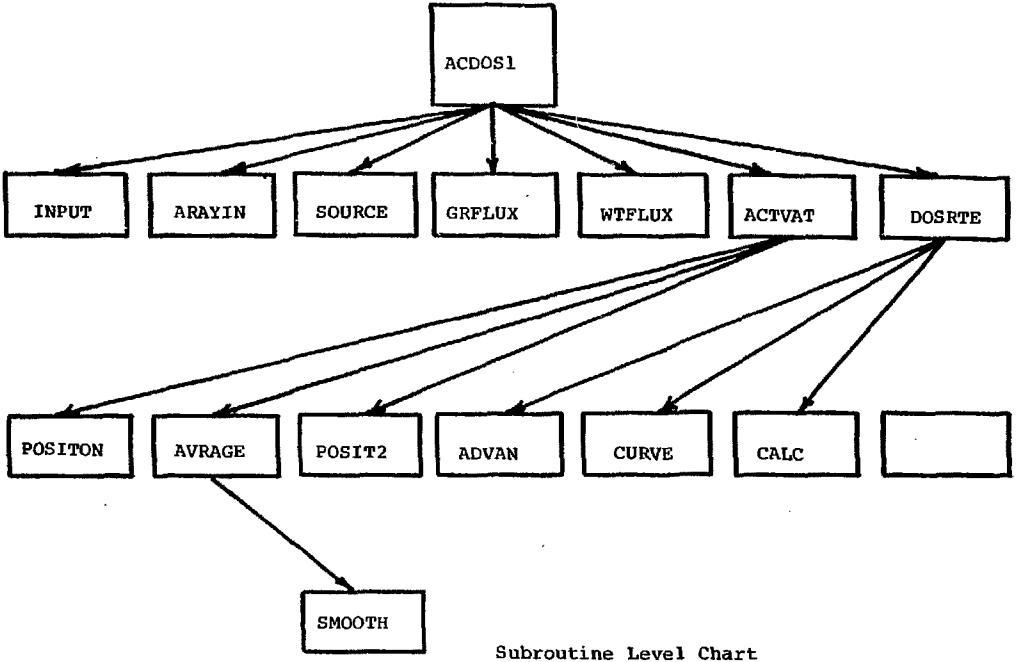



Determine Weighting Function Constants
Search The ACTLMFE Library For Target Nuclide ID Numbers
Target ID Found?
Y
N
Calculate Group Cross-Sections
Calculate Activities Due To Parents And Print Results
Any More Target Nuclides?
Y
N
Was It The Last Target ID?
N
Y


Search The LEVDEC Library For Parent Decay Information
Parent ID # Found?
N
Y
Calculate The Dose Due To Parent If Radioactive And Compile List Of Potential Daughters
Was It The Last Parent ID #?
N
Y
Any More Parents?
N
Y
Search LEVDEC For Possible Radioactive Daughters
Any More Daughters?
Y
N
Was It The Last Daughter ID #?
N
Y
Daughter Found?
N
Y
Calc. Dose Rate Due To Daughter



*Not in detail--only to show the general flow of logic.



Subroutine Level Chart

Description of the ACDOS1 Tape Structure

Tape ACDOS1 has 5 files:

File #	Name	Physical Records	Logical Records
1	ACDOS1 Source Listing (ANSI FORTRAN IV)		
2	ACTLMFE Library	449	20,177
3	LEVDEC 1	551	24,772
4	LEVDEC 2	544	24,476
5	LEVDEC 3	289	12,973

Logical records are 80 column card images.

Tape Characteristics

Type: 9 track, 1600 bpi

Blocking Factor: 45

Format: EBSCDIC

Sample Problem

Determine the dose rates resulting from activation of the copper target in the target section of the injector.

Composition-820lbs Cu

Chosen Geometry-Spherical

From the "Chart of the Nuclides", 9th Edition, 1966

Element	lbs	Isotope	Wt%	# of kg's
Cu	820	Cu^{63}	69.1	257.55
		Cu^{65}	30.9	115.17

Volume of a sphere = $\frac{4}{3}\pi R^3$. Therefore, the volume of copper is: $\frac{3.727 \times 10^5 gm}{8.96 gm/cm^3} = 4.160 \times 10^4\ cm^3$

$$R = \sqrt[3]{\frac{3 \cdot VOL}{4\pi}}$$

$$R = \sqrt[3]{\frac{3 \cdot 4.160 \times 10^4\ cm^3}{4\pi}}$$

$$R = \sqrt[3]{9931.27\ cm^3}$$

$$R = 0.215m$$

The following sample problem parameters apply:

A	current (amperes)	65
V	voltage (kilovolts)	170
F	beam fraction	1.0
DF	duty factor	0.10
T1	length of test (h)	8
T2	length of pause (h)	16
N	number of tests	7
NOEGPS	number of energy groups	20
NOPAS	number of points after shutdown	12
R	radius of sphere	0.215
D	distance from surface of sphere (m)	2
IGEOM	geometry designator (sphere)	2
NONUCL	number of nuclides	2
NF	number of fast neutron groups	2

GP(I), I=1, NG (MeV)	BFLUX (I), I=1, NOEGPS (neutrons/cm^2·sec/Source Neutron)
$1X10^{-11}$	$8.288X10^{-7}$
$4.14X10^{-7}$	$3.076X10^{-8}$
$1.125X10^{-6}$	$3.456X10^{-8}$
$3.059X10^{-6}$	$3.951X10^{-8}$
$1.068X10^{-5}$	$3.011X10^{-8}$
$2.902X10^{-5}$	$4.684X10^{-8}$
$1.013X10^{-4}$	$7.234X10^{-8}$
$5.829X10^{-4}$	$3.043X10^{-8}$
$1.234X10^{-3}$	$4.021X10^{-8}$
$3.355X10^{-3}$	$4.570X10^{-8}$
$1.033X10^{-2}$	$3.522X10^{-8}$
$2.188X10^{-2}$	$7.420X10^{-9}$
$2.479X10^{-2}$	$3.860X10^{-8}$
$5.248X10^{-2}$	$4.790X10^{-8}$
$1.111X10^{-1}$	$2.842X10^{-8}$
$1.576X10^{-1}$	$1.373X10^{-7}$
$5.502X10^{-1}$	$1.058X10^{-7}$
1.108	$8.333X10^{-8}$
1.827	$1.382X10^{-7}$
2.307	$7.901X10^{-6}$
2.385	

Results of Sample Problem

Inspection of the printout shows a dose rate of 2.67 mrad/h at time t=0 after shutdown. The distance from the surface of the sphere is 2m. Of this system dose rate, the majority of the contribution is coming from Copper-64 which was produced by the (n,γ) reaction on Copper-63. Due to the 12.9 hour half-life of Copper-64, one would need to exercise some caution for a period of several hours after shutdown if maintenance were to be performed near the beam dumps. (At 10 cm from the spherically modeled source, the corresponding dose rate is about 129 mrad/h due to Copper-64 alone.) It is interesting to note that the total activities of Copper-64 and Copper-66 only differ by a Factor 3, while the corresponding dose rates differ by a Factor of 7. This difference, of course, is due to the energy and decay probability considerations specific to the two different nuclides.

Although an activity was calculated for Nickel-63, there was no contribution to the dose rate. The long-lived state of Nickel-63 decays to Copper-63 by β^- emission. Since Copper-63 is formed in the ground state, no dose rate results.

Project Conclusions

The objective of this project was to determine the dose rates associated with neutron activation of neutral beam injector components. This objective has been accomplished by creating a Fortran IV program to calculate dose rates as a function of geometry, component composition and amount, and time after shutdown.

The work was successfully completed by performing five major tasks:

1) determination of a suitable mathematical model for the production of neutrons by the injector,
2) modification of the differential equation that describes the system undergoing activation (to reflect the pulse-pause nature of neutron production) and adapting the results to multigroup calculations,
3) development of a program to average microscopic cross-section data over the appropriate energy range (to calculate flux weighted group cross sections),
4) creation of a program to calculate activation given the injector testing history, pertinent beam parameters, component composition and amount, and desired times after shutdown, and finally,
5) development of a program to calculate dose rates given the previously calculated activities, user

specified geometry, and distance from the activated component.

The five tasks manifest themselves through 14 subroutine subprograms and one executive calling program. Together, they constitute the program, ACDOS1, which performs the desired dose rate calculations given the necessary information.

Although ACDOS1 is a complete program for calculating dose rates due to neutron activation, there are some areas where additional work could improve the code from the standpoint of completeness and accuracy. Coupling of a neutron transport code to the "front end" of the program would eliminate the need for the user to supply the input fluxes. With ACDOS1, the user must obtain the input fluxes from a separate calculation. This addition would make the program more self-contained computationally and could also reduce the input data required to run the program. Also, other geometry options could be added to increase the flexibility in modeling unusual source distributions that might arise in non-injector applications. Addition of these options could be made by incorporating the appropriate equation into subroutine CALC which is the subprogram that calculates the dose rates. (Any additional arguments would have to be passed in the calling statement.)

To produce more accurate dose rate calculations, three immediate improvements could be made to the program:

1) a more realistic treatment of radioactive daughters, 2) inclusion of self absorption in the geometrically modeled sources, and 3) a within-cavity geometry option. Currently, radioactive daughters are assumed to have a half life short compared to that of the parent. Although computationally expedient, this is not always realistic. The improvement would involve incorporating a more complex activation equation into the program; and in addition, some major changes in software logic. Inclusion of self absorption in the dose rate calculations would remove a major conservatism inherent in the present treatment of the calculations. Although modification of the dose rate equation would be a relatively simple matter, there would have to be some provision made for generating or acquiring γ ray attenuation coefficients since they vary strongly with composition and energy. An additional library containing such data would probably be the best solution. Another routine would also be necessary for interpolating γ attenuation coefficients between data points. Finally, a within-cavity geometry option could be utilized to include dose rates from activation of the concrete walls and/or floor surrounding the injector. Presently, there is no way to include any contribution to the dose rate from the walls of the injector housing.

At this time a second version of ACDOS1, ACDOS2, is being written and will incorporate many of the above men-

tioned improvements. These modifications will remove many of the approximations used in ACDOS1, and should yield more accurate dose rate calculations.

A final comment involves a paper[10] that was written at ORNL concerning the calculation of dose rates in the Tokomak Fusion Test Reactor Test Cell. Basically, it is a comparison of resultant dose rates in the test cell with and without a neutral beam injector present--the objective being to determine the contribution to the test cell dose rates due to the presence of the injector and its penetration. Although no code details are presented in the article, a comparison of the similarities, differences, and general methodologies used would offer a perspective from which to evaluate or "bench mark" the ACDOS1 code work since both projects have similar objectives but different origins.

A tabular comparison of the two projects is shown in Table 6.

Table 6

ORNL Project	ACDOS1 Project
Photon transport calculations are carried out using Monte Carlo methods.	No photon transport calculations are done. Photon flux at a dose point is determined from standard geometric equations. No self attenuation or buildup is taken into account in geometrically modeled sources.

ORNL Project	ACDOS1 Project
Concrete walls and ceiling of the test cell were considered.	Doesn't consider effects due to activation of the concrete. Considers only activation of injector components inside the injector shield.
Uses cylindrical geometry to calculate spatially averaged dose rates.	Choice of three different geometries. Dose rates are point values rather than spatial averages.
Considers contributions from parents, daughters, and grand-daughters.	Considers contributions from parents and daughters. Grand-daughters are assumed to be stable.
Allows for identification of those nuclides that are contributing to significant dose rates.	Allows for identification of those nuclides that are contributing to significant dose rates.
Arbitrary pulse sequences are allowed.	Arbitrary pulse sequences are allowed.
Calculates dose rates for a variety of times after shutdown.	Calculates dose rates for a variety of times after shutdown.
Neglects burnout of activated nuclei.	Neglects burnout of activated nuclei.
Activation photons produced inside the igloo which stream into the injector through the penetration were considered.	No provisions for including this contribution to dose rates.
Appears to be specific to neutral beam activation and dose rate calculations.	Can be used for non-injector activation and dose rate calculations.

Review of Table 6 illuminates the similarities and differences. Some of these differences will most likely be eliminated with the second version of ACDOS1, ACDOS2.

Inclusion of self attenuation and buildup in dose rate calculations along with dose contributions from concrete injector walls will probably be addressed. When these modifications are made to ACDOS1, calculated dose rates will probably be more in line with those resulting from a transport treatment of the γ photons. The ACDOS1 code was never intended to consider activation photons produced in an igloo and which stream into the injector through the penetration. This results from the nature of the design work being done at LBL and the forthcoming upgrade program which will require prolonged testing periods at higher beam currents and duty factors.

In general terms, the ORNL calculation is probably more accurate due to the photon transport approach to dose rate calculations. ACDOS1, however, appears to be more versatile. With upcoming improvements to ACDOS1 as mentioned above, improved accuracy coupled with the code's versatility should make it a useful tool for injector design studies and non-injector applications.

Acknowledgements

I wish to acknowledge the assistance given to me by the consulting staff of the Lawrence Berkeley Laboratory Computer Center during the development of several programs required to solve the problems associated with this project. Their help and guidance was greatly appreciated.

Also, I would like to thank Mr. Robert J. Howerton of LLNL who expended considerable time and effort in providing both data libraries, preliminary flux-weighted group cross sections, and advice and consultation throughout this endeavor. This work was supported in part by the Office of Fusion Energy, U. S. Department of Energy, under Contract No. W-7405-ENG-48.

Appendix A

Solutions to Solving Transcendental Equations

For problems involving two or more neutron groups, transcendental equations will result due to the assumed form of the fast group exponential used for representing the weighting flux function at higher energies. The exponential is:

$$Ae^{\frac{E-E1}{B}}$$

where A and B are constants to be determined by applying the appropriate boundary and integral constraints on the pertinent equations, and E1 is the highest energy in the particular fast group being evaluated--the upper boundary for that group. There are three cases where transcendental equations will result:

1) Matching the thermal flux to a fast flux
2) Matching a 1/E flux to a fast flux, and
3) Matching a fast flux to a fast flux

The match is always required at the boundary between the two groups.

Consider Case 1. It is required to match the Maxwellian to the fast group at the boundary between the two groups as shown in Figure 11. ϕ(E) is the fast weighting flux function and GP(J), J=1, 2, 3 are the group boundaries. In addition, it is required that the integral of ϕ (E) over group 2 be equal to the total flux as

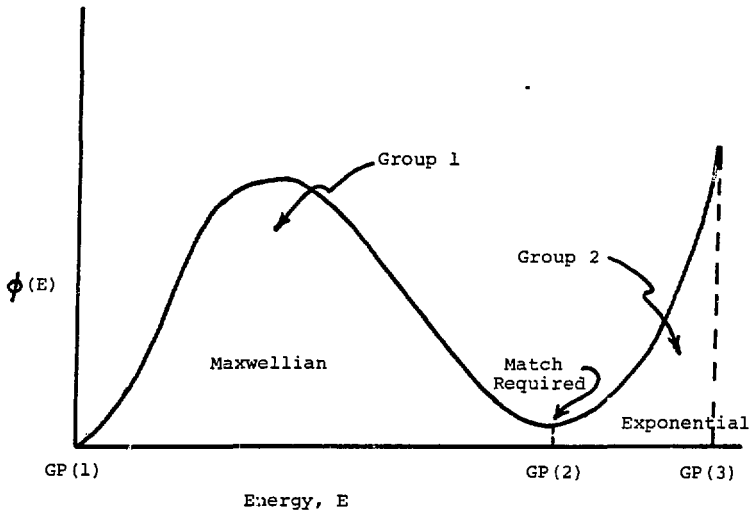

Group 1
Group 2
ϕ(E)
Maxwellian
Match
Required
Exponential
GP(1)
GP(2)
GP(3)
Energy, E

Figure 11

discussed on page . Since the constant in the Maxwellian ($C\sqrt{E}\exp(-(E/0.025))$ has already been evaluated, the flux at the boundary (FLUXBD) is known. It is:

```
FLUXBD=CONST(1)*SQRT(GP(2))*EXP(-(GP(2)/0.025))
```

where CONST(1) is the constant associated with the Maxwellian and GP is in eV. The first equation is:

$$\text{FLUXBD} = A\, e^{\frac{GP(2)-GP(3)}{B}} \qquad (1)$$

in accordance that the flux match at the group boundary GP(2). The second equation is:

$$\int_{GP(2)}^{GP(3)} Ae^{\frac{E-GP(3)}{B}}\, dE = \text{total flux for that group} = \text{BFLUX(NG)}$$

where NG=2 is the total number of groups and BFLUX is an array holding the total flux values. Evaluating this integral:

$$ABe^{\frac{(E-GP(3))}{B}}\Big|_{GP(2)}^{GP(3)} = \text{BFLUX(NG)}$$

$$AB-ABe^{\frac{GP(2)-GP(3)}{B}} = \text{BFLUX(NG)}$$

$$AB(1-e^{\frac{GP(2)-GP(3)}{B}}) = \text{BFLUX(NG)} \qquad (2)$$

From equation (1), $A = FLUXBD\ e^{-\left(\frac{GP(2)-GP(3)}{B}\right)}$

Equating this to equation (2), we get

$$FLUXBD\ e^{-\left(\frac{GP(2)-GP(3)}{B}\right)} \cdot B\left(1-e^{\left(\frac{GP(2)-GP(3)}{B}\right)}\right)=BFLUX(NG)$$

As a result,

$$B = \frac{BFLUX(NG)}{FLUXBD}\left\{e^{\frac{GP(3)-GP(2)}{B}}-1\right\}^{-1} \qquad (3)$$

This is the above mentioned transcendental equation. Once B is found, A can be determined by substituting B into equation (2). Subroutine WTFLUX, which evaluates the constants associated with the assumed weighting functions, used the Newton-Raphson Method to solve equation (3). The numerator in equation (3), BFLUX(NG)/FLUXBD, is always a known quantity and is of the same order of magnitude as the numerator in the exponential, GP(3)-GP(2). Therefore, BFLUX(NG)/FLUXBD is used as an initial guess to start the iterative routine which solves the equation.

The second case involves matching a 1/E flux to a

fast flux as illustrated in Figure 12. The same type of method is used as before. There are two requirements: (1) continuity of flux at the boundary, GP(NG), and (2) an integral constraint on the fast group. From the first requirement,

$$\text{FLUXBD} = \frac{\text{CONST(ICOUNT)}}{\text{GP(NG)}} = Ae^{\frac{\text{GP(NG)}-\text{GP(NG+1)}}{B}} \tag{4}$$

where CONST(ICOUNT) is the known constant associated with the previous 1/E group. ICOUNT is a neutron group counter. From equation (4), A can be solved in terms of B:

$$A = \frac{\text{CONST(ICOUNT)}}{\text{GP(NG)}}\, e^{-\left(\frac{\text{GP(NG)}-\text{GP(NG+1)}}{B}\right)} \tag{5}$$

From the second requirement:

$$\int_{\text{GP(NG)}}^{\text{GP(NG+1)}} Ae^{\frac{E-\text{GP(NG+1)}}{B}}\, dE = \text{total flux for the fast group} = \text{BFLUX(NG)}$$

$$AB - ABe^{\frac{\text{GP(NG)}-\text{GP(NG+1)}}{B}} = \text{BFLUX(NG)}$$

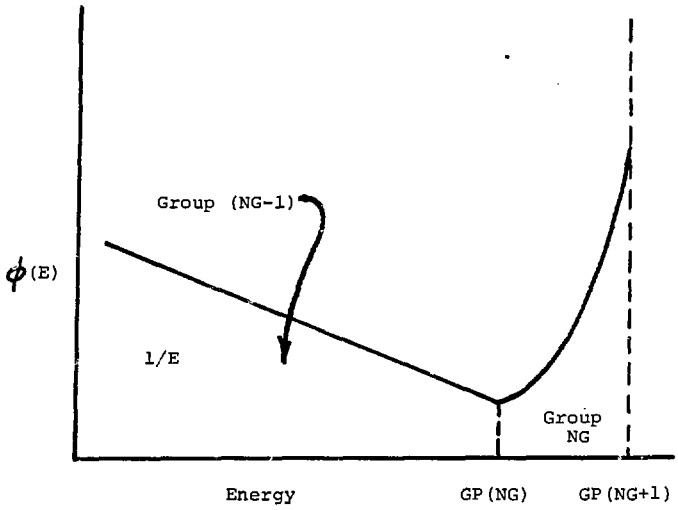


Figure 12

$$AB\left(1-e^{\frac{GP(NG)-GP(NG+1)}{B}}\right) = BFLUX(NG)$$

$$A = \frac{BFLUX(NG)}{B}\left\{1-e^{\frac{GP(NG)-GP(NG+1)}{B}}\right\}^{-1} \qquad (6)$$

equating equations (5) and (6)

$$B = \frac{GP(NG)\ \ BFLUX(NG)}{CONST(ICOUNT)}\left\{e^{\frac{GP(NG+1)-GP(NG)}{B}}-1\right\}^{-1} \qquad (7)$$

As before, the numerator in this equation is always known and is of the same order of magnitude as the exponential. Therefore, GP(NG)*BFLUX(NG)/CONST(ICOUNT) is used as an initial guess to start the Newton-Raphson iterative routine.

The third case involves matching a fast flux to a fast flux as shown in Figure 13. Again, the same method applies. The requirements yield two equations. From the continuity of flux requirement at GP(ICOUNT+1):

$$FLUXBD = CONST(ICOUNT) = Ae^{\frac{GP(ICP1)-GP(ICP2)}{B}}$$

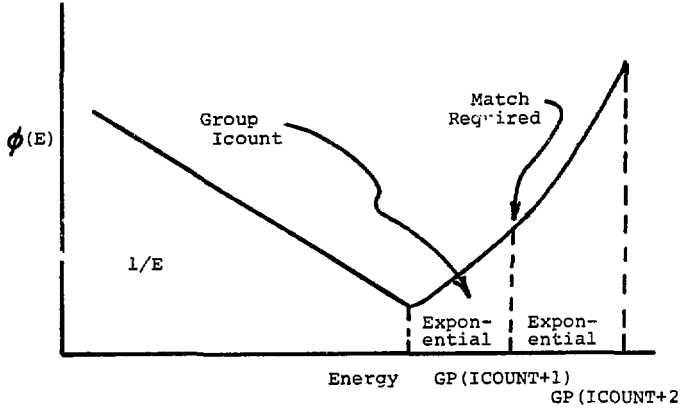

Match
Required
Group
Icount
ϕ(E)
1/E
Expon-
ential
Expon-
ential
Energy
GP(ICOUNT+1)
GP(ICOUNT+2)

Figure 13

where for brevity, ICP1 = ICOUNT+1 and ICP2 = ICOUNT+2. Note that the exponential associated with CONST(ICOUNT) equals 1 when evaluated at GP(ICP1). Therefore,

$$A = \text{CONST(ICOUNT)}\ e^{-\left(\frac{GP(ICP1)-GP(ICP2)}{B}\right)} \tag{8}$$

The integral constraint requires that

$$\int_{GP(ICP2)}^{GP(ICP1)} Ae^{\frac{E-GP(ICP2)}{B}}\, dE = \text{total flux for the fast group} = \text{BFLUX(ICP1)}$$

$$ABe^{\frac{E-GP(ICP2)}{B}}\Bigg|_{GP(ICP1)}^{GP(ICP2)} = \text{BFLUX(ICP1)}$$

$$AB\left(1-e^{\frac{GP(ICP1)-GP(ICP2)}{B}}\right) = \text{BFLUX(ICP1)}$$

$$A = \frac{\text{BFLUX(ICP1)}}{B}\left\{1-e^{\frac{GP(ICP1)-GP(ICP2)}{B}}\right\}^{-1} \tag{9}$$

Equating equation (8) and (9),

$$B = \frac{BFLUX(ICP1)}{CONST(ICOUNT)} \left\{ e^{\frac{GP(ICP2)-GP(ICP1)}{B}} - 1 \right\}^{-1}$$

As before, the numerator (BFLUX(ICP1)/CONST(ICOUNT) is used as an initial guess. A is then found from equation (8).

In all three cases, the methodology employed to solve the equations is the same. Since the numerators in the pertinent equations are all known quantities when they are required, the user does not have to enter an initial guess. Subroutine WTFLUX performs all of the calculations required to determine the constants associated with the assumed weighting functions. Furthermore, all of the calculated constants are stored in memory for future use in subroutine AVRAGE which calculates the flux weighted group cross sections. The A's are stored in array CONST(I) and the B's in array FCONST(I).

One word of caution is appropriate. It is entirely possible that converging problems may occur during the execution of the routine that solves the transcendental equations.

The Newton-Raphson Method

This method is used to solve the transcendental equations that result when two or more neutron groups are used. The interative expression is:

$$X_{i+1} = X_i - f(x_i)/f'(x_i)$$

where x_i is the initial guess.

The parameter B (see Appendix A) is always given by the following form:

$$B = \text{KNOWN QUANTITY}\left\{e^{\frac{GP(N+1)-GP(N)}{B}}-1\right\}^{-1}$$

where GP(N+1) is a group boundary higher in energy than GP(N). Therefore,

$$f(B) = \text{KNOWN QUANTITY}\left\{e^{\frac{GP(N+1)-GP(N)}{B}}-1\right\}^{-1} - B$$

and,

$$f'(B) = \frac{\text{KNOWN QUANTITY}\left(GP(N+1)-GP(N)\right) e^{\frac{GP(N+1)-GP(N)}{B}}}{B^2\left[e^{\frac{GP(N+1)-GP(N)}{B}}-1\right]^2} - 1$$

These two expressions are incorporated into subroutine WTFLUX for evaluating the constants associated with the exponential weighting functions. When the difference between B_{i+1} and B is less than 0.1, the iteration terminates.

Appendix C

```
      **PROGRAM ACDOS1 (INPUT,OUTPUT,TAPE5=INPUT,TAPE7,TAPE6=OUTPUT)**

      PROGRAM ACDCS1 (INPUT,OUTPUT,TAPE5=INPUT,TAPE7,TAPE6=OUTPUT)
C
C
C
C
C ACDOS1 IS A STANDARD FORTRAN 4 CODE DEVELOPED FOR CALCULATING DOSE RATE
C S FROM NEUTRON ACTIVATION OF NEUTRAL-BEAM INJECTORS.SUFFICIENT VERSATIL
C ITY HAS ALSO BEEN INCORPORATED INTO THE CODE TO MAKE IT APPLICABLE TO A
C WIDE VARIETY OF GENERAL ACTIVATION PROBLEMS DUE TO NEUTRONS OF ENERGY
C LESS THAN 20 MEV. FOR FURTHER INFORMATION CONCERNING THIS PROGRAM, SEE
C LSL REPORT NO. 12244 BY GREGORY S KENEY
C
C
C
C        VARIABLES AND ARRAYS ARE DEFINED AS BELOW
C
C        A                 CURRENT IN AMPERES
C        AC(I)             PARENT ACTIVITY
C        ACT               ACTIVITY DUE TO ACTIVATION OF A PARTICULAR TARG
C                          ET NUCLIDE BY A SPECIFIC NEUTRON REACTION
C        AVENPS            AVERAGE NUMBER OF NEUTRONS PRODUCED PER SECOND
C        BFLUX(I)          UNIT NORMALIZED FLUX
C        CF                NEUTRON YIELD CORRECTION FACTOR FOR VOLTAGES
C                          DIFFERENT FROM 150 KV
C        CONST(I)          CONSTANTS FOR WEIGHTING FUNCTIONS
C        CS(I)             CROSS SECTION VALUE
C        CS(I)             CROSS SECTION VALUE
C        C                 DISTANCE FROM POINT SOURCE OR SURFACE OF
C                          SPHERE OR CYLINDER
C        DF                DUTY-FACTOR
C        DIFF              DIFFERENCE BETWEEN THE LATEST REFINED GUESS AND
C                          THE PREVIOUS GUESS
C        DLEVL             LEVEL OF THE PARENT OR DAUGHTER
C        DLEVL(I)          LEVEL OF THE PARENT OR DAUGHTER
C        DOSDAU            DOSE RATE DUE TO A RADIOACTIVE DAUGHTER
C        DOSPRC(I)         DOSE RATE DUE TO A PARENT
C        DOSRAT            CALCULATED DOSERATE
C        DOSSUM(I)         DOSE RATE DUE TO A PARTICULAR TARGET NUCLIDE
C                          FOR UP TO 12 DIFFERENT TIMES
C        DPVAR             VALUE OF DEPENDENT VARIABLE IN NUMERICAL
C                          INTEGRATIONS
C        E                 ENERGY OF GAMMA RAY
C        EN(I)             ENERGY VALUE ASSOCIATED WITH A CROSS SECTION
C        E1(I)             ENERGY VALUE ASSOCIATED WITH A CROSS SECTION
C        F                 BEAM FRACTION
C        FCONST(I)         EXPONENTIAL CONSTANTS FOR FAST NEUTRON GROUPS
C        FLUXBD            WEIGHTING FLUX AT A GROUP BOUNDARY
C        FPRIME            EVALUATED VALUE OF THE DERIVATIVE OF THE
C                          TRANSCENDENTAL EQUATION FOR B
C        FXSUBI            EVALUATED VALUE OF THE TRANSCENDENTAL EQUATION
C                          FOR B
C        GP                GROUP BOUNDARIES
C        GPFLUX(I)         FEW GROUP FLUXES-UP TO 50
C        GPXSEC(I)         FEW GROUP CROSS SECTIONS-UP TO 50
C        GUESS1            INITIAL GUESS FOR SOLVING TRANSCENDENTAL
C                          EQUATIONS
C        H(I)              PARENT HALF LIFE
C        HT                HEIGHT OF CYLINDER(M)
```

```
      **PROGRAM ACDOS1 (INPUT,OUTPUT,TAPE5=INPUT,TAPE7,TAPE6=OUTPUT)**
C
C          IBASE        MARKS THE BEGINNING OF THE PARENT NUCLIDE LIST
C          IBEGIN       MARKS THE BEGINNING OF THE LIKE PARENTS NUCLIDE
C                       LIST
C          ICCUNT       NEUTRON GROUP COUNTER
C          ICP1         ICCUNT+1
C          ICP2         ICCUNT+2
C          IDAUG(I)     DAUGHTER ID NUMBER
C          IDAUGH       DAUGHTER ID READ FROM LEVDEC
C          IDEKMO       DECAY MODE IDENTIFIER
C          IDEKM(I)     DECAY MODE IDENTIFIER
C          IDMO(I)      5 DIGIT ID NUMBER OF TARGET NUCLIDE
C          IDUMB        DUMMY VARIABLE FOR ADVANCING DISK FILE RECORDS
C          IEND         MARKS THE END OF THE LIKE PARENT NUCLIDES IN
C                       THE LIST
C          IEPTH        COUNTER FOR INTERMEDIATE NEUTRON GROUPS
C          IGEOM        DESIGNATES PROBLEM GEOMETRY
C          IMODE        DECAY MODE COUNTER
C          INSNPS       INSTANTANEOUS NUMBER OF NEUTRONS PRODUCED PER
C                       SECOND
C          IORDER(I)    HOLDING ARRAY FOR PRODUCT NUCLIDE ID NUMBERS
C          IPROD        PRODUCT NUCLIDE COUNTER
C          ISTOP        MARKS THE END OF THE LIST OF ARRANGED PARENT
C                       NUCLIDES
C          ITAPE        READ VARIABLE FOR DISK FILE DATA
C          ITER         ITERATION COUNTER
C          MASS(I)      NUMBER OF KILOGRAMS OF A PARTICULAR NUCLIDE
C          MULT         MULTIPLICITY OF GAMMA RAY
C          N            NUMBER OF TESTS
C          NEPTH        NUMBER OF INTERMEDIATE NEUTRON GROUPS
C          NEI(I)       NUMBER OF EI CI PAIRS
C          NF           NUMBER OF FAST NEUTRON GROUPS
C          NG           NUMBER OF GROUP BOUNDARIES-UP TO 51
C          NGEGFS       NUMBER OF ENERGY GROUPS-UP TO 50
C          NONUCL       NUMBER OF TARGET NUCLIDES-UP TO 30 PER RUN
C          NOPAS        NUMBER OF POINTS AFTER SHUTDOWN-UP TO 12
C          NOTORD(I)    HOLDING ARRAY FOR PRODUCT NUCLIDE NUMBERS THAT
C                       HAVE NOT BEEN RE-ARRANGED IN ASCENDING ORDER
C          NOUT(I)      ID NUMBER OF DAUGHTER
C          NP           NUMBER OF ENERGY-CROSS-SECTION PAIRS
C          NP(I)        ID NUMBER OF PARENT
C          NTEMP1       TEMPORARY STORAGE FOR PARENT ID NUMBERS
C          NTEMP2       TEMPORARY STORAGE FOR TARGET NUCLIDE ID NUMBERS
C          NT(I)        ID NUMBER OF TARGET
C          OPTION       VARIABLE THAT DETERMINES WHETHER OR NOT AVENPS
C                       WILL BE CALCULATED BY ACDOS OR READ FROM A
C                       DATA CARD AS PRE-DETERMINED INPUT
C          PARSUM       PARTIAL SUM IN NUMERICAL INTEGRATIONS
C          PROB(I)      DECAY PROBABILITY
C          PRODHL(I,J)  PRODUCT NUCLIDE HALF-LIFE (S)
C          PRODNU(I,J)  5 DIGIT ID NUMBER OF PRODUCT NUCLIDE
C          R            RADIUS OF HOMOGENEOUS NON-ABSORBING SPHERE OR
C                       CYLINDER (M)
C          REALDUM      REAL DUMMY VARIABLE FOR ADVANCING DISK FILE
C                       RECORDS
C          STAFS(I)     SPECIFIC TIMES AFTER SHUTDOWN (H)
C          STOPHL(I)    HOLDING ARRAY FOR PRODUCT NUCLIDE ID NUMBERS
C          STOTACT(I,J) HOLDING ARRAY FOR PRODUCT NUCLIDE ACTIVITIES
C          SUMACT(I)    SUM OF ACTIVITIES PRODUCED FROM A PARTICULAR
```

```
      **PROGRAM ACDCS1 (INPUT,OUTPUT,TAPE5=INPUT,TAPE7,TAPE6=OUTPUT)**
C                                 TARGET NUCLIDE FOR UP TO 12 DIFFERENT TIMES
C          SUM1                   EXPONENTIAL EFFECT OF TESTS AND PAUSES ON THE
C                                 SHUTDOWN TIME ACTIVITY OF A SPECIFIC PRODUCT
C                                 NUCLIDE
C          SUMDEL                 RUNNING SUM OF AVERAGE CROSS SECTIONS
C          SUMINT                 RESULT OF NUMERICAL INTEGRATIONS
C          SYSACT(I)              SYSTEM ACTIVITY FOR UP TO 12 DIFFERENT TIMES
C          SYSDCSE(I)             SYSTEM DCSERATE FOR UP TO 12 DIFFERENT TIMES
C          TARMAS                 ATOMIC WEIGHT OF TARGET NUCLIDE (AMU)
C          TEMPS                  TEMPORARY STORAGE FOR DO LOOP SUMS
C          TIMACT(I,J,K)          INDUCED ACTIVITY AS A FUNCTION OF TARGET
C                                 NUCLIDE,REACTION, AND TIME
C          T1                     LENGTH OF TEST (H)
C          T2                     LENGTH OF PAUSE (H)
C          V                      VOLTAGE (KV)
C          WIDTH                  DELTA E USED IN NUMERICAL INTEGRATIONS
C          XSBIP1                 LATEST REFINED GUESS FOR B
C          XSUBI                  INDEPENDENT VARIABLE OF THE TRANSCENDENTAL
C                                 EQUATION FOR B
C          ZINCR                  INCREMENT IN ENERGY USED IN NUMERICAL
C                                 INTEGRATIONS
C          ZNAME                  VARIABLE FOR NUCLIDE NAMES
C          ZNUMER                 NUMERATOR IN THE TRANSCENDENTAL EQUATION FOR B
C          ZTEMP3                 TEMPORARY STORAGE FOR PARENT ACTIVITIES
C          ZTEMP4                 TEMPORARY STORAGE FOR PARENT HALF LIFES
C
C
C
C
      DIMENSION TIMACT(30,13,12),SYSACT(12),SUMACT(12),SYSDCS(12),PRODHL
     +(30,13),PRODNU(30,13),DCSSUM(12),ZNAME(30),STAFS(12),BFLUX(50),IDN
     +C(30),GPXSEC(50),MASS(30),GPFLUX(50),GP(51),CCNST(50),FCONST(50),D
     +OSNUC(13)
      COMMON A,V,F,DF,T1,T2,A,B,D,HT,OPTION,IGEOM
      REAL INSNPS,MASS
      CALL INPUT(NONUCL,NDEGPS,NOPAS,NG,NF)
      CALL ARAYIN(BFLUX,ZNAME,IDNO,MASS,STAFS,GP,NDEGPS,NONUCL,NOPAS,NG)
      IF(OPTION.EQ.1.CCCCCCC) GO TO 1
      AVERPS=OPTION
      GO TO 2
    1 CALL SOURCE(AVERPS)
    2 CALL GPFLUX(AVERPS,NDEGPS,GPFLUX,BFLUX)
      CALL WTFLUX(BFLUX,GP,CCNST,FCONST,NDEGPS,NF,NG)
      CALL ACTVAT(TIMACT,SYSACT,SUMACT,GPFLUX,PRODNU,PRODHL,GPXSEC,IDNO,
     +MASS,STAFS,GP,CCNST,FCONST,BFLUX,NONUCL,NDEGPS,NOPAS,NG,NF)
      CALL DCSRTE(TIMACT,PRODNU,NONUCL,NOPAS,DCSSUM,SYSDCS,IDNO,STAFS,PR
     +ODHL,DOSNUC)
      STOP
      END
```

```
      **SUBRCUTINE INPUT(NCNUCL,NCEGPS,NOPAS,NG,NF)**

      SUBRCUTINE INPUT(NCNUCL,NCEGPS,NOPAS,NG,NF)
C
C
CTHIS PROGRAM IS USED FOR ENTERING 'VARIABLE' DATA INTO MEMORY
C
C
C     ARGUMENTS IN-NONE
C     ARGUMENTS OUT-NCNUCL,NCEGPS,NOPAS,NG,NF
C
C
      COMMCN A,V,F,DF,T1,T2,N,R,D,HT,OPTION,IGEOM
      READ(5,80) Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10
   80 FORMAT(10(A8))
      READ(5,61) OPTION
   61 FORMAT(E14.7)
      IF(OPTION.NE.1.0000000E+00) GO TO 62
      READ(5,1) A,V,F,DF,T1,T2,N
    1 FORMAT(6(E10.3,2X),I3)
      IF(A.LE.0.0) GO TO 36
      IF(V.LT.38.0.OR.V.GT.300.0) GO TO 37
      IF(F.LE.0.0.OR.F.GT.1.0) GO TO 38
      IF(DF.LE.0.0.OR.DF.GT.1.0) GO TO 39
   64 IF(T1.LE.0.0) GO TO 40
      IF(T2.LT.0.0) GO TO 41
      DO 42 K=1,999
      IF(K.EQ.K) GO TO 43
   42 CONTINUE
      GO TO 44
   43 READ(5,2) NCEGPS
    2 FORMAT(I2)
      DO 45 K=1,50
      IF(NCEGPS.EQ.K) GO TO 46
   45 CONTINUE
      GO TO 9
   46 READ(5,2) NOPAS
      DO 47 K=1,12
      IF(NOPAS.EQ.K) GO TO 48
   47 CONTINUE
      GO TO 10
   48 READ (5,5) R,D,HT,IGEOM
    5 FORMAT(3(E10.3,2X),I1)
      IF(IGEOM.EQ.1) GO TO 70
      IF(IGEOM.EQ.2) GO TO 70
      IF(IGEOM.EQ.3) GO TO 70
      GO TO 67
   70 IF(R.LT.0.0.OR.D.LE.0.0) GO TO 49
      IF(HT.LT.0.0) GO TO 77
      READ(5,2) NONUCL
      DO 50 K=1,30
      IF(NCNUCL.EQ.K) GO TO 51
   50 CONTINUE
      GO TO 11
   51 READ(5,2) NF
      GO TO 19
   62 READ(5,63) T1,T2,N
   63 FORMAT(2(E10.3,2X),I3)
      GO TO 64
```

```
      **SUBROUTINE INPUT(NCNUCL,NOEGPS,NOPAS,NG,NF)**

    9 WRITE(6,12)
   12 FORMAT(1H ,110HPROGRAM 'ACDOS' ABORTED-NUMBER OF ENERGY GROUPS MUS
     +T BE INTEGER AND LESS THAN OR EQUAL TO 50-RECHECK THIS DATA)
      STOP
   13 WRITE(6,14)
   14 FORMAT(1H ,118HPROGRAM 'ACDOS' ABORTED-NUMBER OF POINTS AFTER SHUT
     +DOWN MUST BE INTEGER AND LESS THAN OR EQUAL TO 12-RECHECK THIS DAT
     +A)
      STOP
   11 WRITE(6,17)
   17 FORMAT(1H ,96HPROGRAM 'ACDOS' ABORTED-NUMBER OF NUCLIDES IN SYSTEM
     +MUST BE INTEGER AND LESS THAN OR EQUAL TO 30)
      STOP
   36 WRITE(6,52)
   52 FORMAT(1H ,80HPROGRAM 'ACDOS' ABORTED-CURRENT MUST BE GREATER THAN
     + 0 AMPERES-RECHECK THIS DATA)
      STOP
   37 WRITE(6,53)
   53 FORMAT(1H ,89HPROGRAM 'ACDOS' ABORTED-VOLTAGE MUST BE BETWEEN 38 A
     +ND 200 KV INCLUSIVE-RECHECK THIS DATA)
      STOP
   38 WRITE(6,54)
   54 FORMAT(1H ,107HPROGRAM 'ACDOS' ABORTED-BEAMFRACTION MUST BE GREATE
     +R THAN 0 AND LESS THAN OR EQUAL TO 1.0-RECHECK THIS DATA)
      STOP
   39 WRITE(6,55)
   55 FORMAT(1H ,107HPROGRAM 'ACDOS' ABORTED-DUTYFACTOR MUST BE GREATER
     +THAN 0.0 AND LESS THAN OR EQUAL TO 1.0-RECHECK THIS DATA)
      STOP
   40 WRITE(6,56)
   56 FORMAT(1H ,78HPROGRAM 'ACDOS' ABORTED-PULSE LENGTH MUST BE GREATER
     +THAN 0.0-RECHECK THIS DATA)
      STOP
   41 WRITE(6,57)
   57 FORMAT(1H ,90HPROGRAM 'ACDOS' ABORTED-PAUSE LENGTH MUST BE GREATER
     +THAN OR EQUAL TO 0.0-RECHECK THIS DATA)
      STOP
   44 WRITE(6,58)
   58 FORMAT(1H ,104HPROGRAM 'ACDOS' ABORTED-NUMBER OF PULSES MUST BE IN
     +TEGER AND LESS THAN OR EQUAL TO 999-RECHECK THIS DATA)
      STOP
   49 WRITE(6,60)
   60 FORMAT(1H ,104HPROGRAM 'ACDOS' ABORTED-R MUST BE GREATER THAN OR
     +EQUAL TO 0.0 AND C GREATER THAN 0.0-RECHECK THIS DATA)
      STOP
   77 WRITE(6,78)
   78 FORMAT(1H ,81HPROGRAM 'ACDOS' ABORTED-HT MUST BE GREATER THAN OR E
     +QUAL TO 0.0-RECHECK THIS DATA)
      STOP
   67 WRITE(6,68)
   68 FORMAT(1H ,64HPROGRAM 'ACDOS' ABORTED-IGECM MUST BE 1,2,OR 3-RECHE
     +CK THIS DATA)
      STOP
   19 WRITE(6,81) Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10
   81 FORMAT(1H1,10(A8))
      WRITE(6,20)
   20 FORMAT(///,1H ,47HTHE FOLLOWING DATA HAS BEEN ENTERED INTO MEMORY,
     +////)
```

```
      **SUBRCUTINE ZAPLT(ACAUCL,NOEGPS,NCPAS,NG,NF)**

      WRITE(6,21)
21    FCRMAT(1HO,85HAMPERES      KILO-VCLTS      BEAMERACTICN        OUTVEA
     +CTCR        T1              T2              N1
      IF(OPTICN.EQ.1.CCCCCOC) WRITE(6,13) A,V,F,DF,T1,T2,N
13    FCRMAT(1HO,E8.3,4X,E8.3,8X,E8.3,11X,E8.3,6X,E8.3,3X,E8.3,3X,I3,///
     +)
      IF(CPTICN.NE.1.CCCCCCC) WRITE(6,65) T1,T2,N
65    FCRMAT(1HC,55H*********      ********        ********        *****
     +****,6X,2(E8.3,2X),I3,///)
      WRITE(6,22) NCEGPS
22    FORMAT(1H ,35HNUMBER OF ENERGY GROUPS-NCEGPS      ,I2,/)
      WRITE(6,75) NCNUCL
75    FCRMAT(1H ,35HNUMBER CF TARGET NUCLICES-NCNUCL      ,I2,/)
      WRITE(6,23) NOPAS
23    FCRMAT(1H ,39HNUMBER OF POINTS AFTER SHUTDOWN-NOPAS      ,I2,/)
      WRITE(6,25) R
25    FORMAT(1H ,35HRADILS R OF SPHERE OR CYLINDER (M)      ,E10.3,/)
      WRITE(6,79) HT
79    FCRMAT(1H ,35HHEIGHT HT CF CYLINDER (M)      ,E10.3,/)
      WRITE(6,26) D
26    FCRMAT(1H ,68HDISTANCE D FRCM POINT SOURCE OR SURFACE OF SPHERE OR
     + CYLINDER (M)      ,E10.3,/)
      NG=NCEGPS+1
      WRITE(6,72) NG
72    FCRMAT(1H ,39HNUMBER CF GRCUPS BOUNDARIES-NG      ,I2,/)
      WRITE(6,76) NF
76    FCRMAT(1H ,39HNUMBER CF FAST NEUTRCN GROUPS-NF      ,I2,/)
      IF(CPTION.NE.1.CCCCCCC) WRITE(6,66) CPTICN
66    FCRMAT(1H ,46HAVERAGE NUMBER OF NEUTRONS PRODUCED PER SECCND,4X,E1
     +C.4,/)
      RETURN
      END
```

```
**SUBROUTINE ARAYIN(BFLUX,ZNAME,IDNO,MASS,STAFS,GP,NOEGPS,NONUCL,NOP**
      SUBROUTINE ARAYIN(BFLLX,ZNAME,IDNO,MASS,STAFS,GP,NOEGPS,NONUCL,NOP
     +AS,NG)
C
C
CTHIS PROGRAM IS USED FOR ENTERING 'ARRAY' DATA INTO MEMORY
C
C
C     ARGUMENTS IN-STAFS,NOEGPS,NONUCL,NOPAS,NG
C     ARGUMENTS OUT-BFLUX,ZNAME,IDNO,MASS,GP
C
C
      DIMENSION BFLUX(NOEGPS),ZNAME(NONUCL),IDNO(NONUCL),MASS(NONUCL),ST
     +AFS(NOPAS),GP(NG)
      REAL MASS
      READ(5,4)(BFLUX(I),I=1,NOEGPS)
      READ(5,4)(STAFS(I),I=1,NOPAS)
    4 FORMAT(6(E10.3,2X))
      DO 7 I=1,NONUCL
      READ(5,8) ZNAME(I),IDNO(I),MASS(I)
    8 FORMAT(A8,1X,I5,1X,E10.3)
    7 CONTINUE
      READ(5,4)(GP(I),I=1,NG)
      WRITE(6,24)
   24 FORMAT(/,1H ,33HSPECIFIC TIMES AFTER SHUTDOWN (H),/)
      DO 34 I=1,NOPAS
      WRITE(6,35) I,STAFS(I)
   35 FORMAT(1H ,4HTIME,1X,I2,10X,E10.3)
   34 CONTINUE
      WRITE(6,27)
   27 FORMAT(///,1H ,22HUNIT NORMALIZED FLUXES,/)
      DO 32 I=1,NOEGPS
      II=NOEGPS+1-I
      WRITE(6,28) I,BFLUX(II)
   28 FORMAT(1H ,5HGROUP,1X,I2,10X,E11.4)
   32 CONTINUE
      WRITE(6,73)
   73 FORMAT(///,1H ,22HGROUP BOUNDARIES (MEV),/)
      WRITE(6,74)(I,GP(NG+1-I),I=1,NG)
   74 FORMAT(1H ,8HBOUNDARY,1X,I3,5X,E11.4)
      WRITE(6,29)
   29 FORMAT(1H ,///,1H ,2X,4HNAME,4X,9HID-NUMBER,3X,8HMASS(KG),/)
      DO 30 I=1,NONUCL
      WRITE(6,31) ZNAME(I),IDNO(I),MASS(I)
   31 FORMAT(1X,A8,5X,I5,5X,E8.3)
   30 CONTINUE
      RETURN
      END
```

```
      **SUBROUTINE SOURCE(AVENPS)**

      SUBROUTINE SOURCE(AVENPS)
C
C
CTHIS PROGRAM CALCULATES THE SOURCE NEUTRON PRODUCTION RATE
C
C
C     ARGUMENTS IN-NONE
C     ARGUMENTS OUT-AVENPS
C
C
      COMMON A,V,F,CF,T1,T2,N,A,D,HT,CPTION,IGECM
      REAL INSNPS
      IF(V.GE.38.0.AND.V.LE.80.0) GO TO 3
      IF(V.GT.80.0.AND.V.LE.180.0) GO TO 4
      IF(V.GT.180.0.AND.V.LE.300.0) GO TO 5
    3 CF=(3.25917*V**2.83194)/38.5
      GO TO 7
    4 CF=(1.51725*V**2.48206)/38.5
      GO TO 7
    5 CF=(8.8466)*V**2.14122)/38.5
    7 INSNPS=(8.64E+4)*(A)*(1.0E+6)*F*(CF*1.0E-4)
      AVENPS=INSNPS*CF
      WRITE(6,8)
    8 FORMAT(///,1H ,43HTHE FOLLOWING HAS BEEN CALCULATED BY ACDCS1,//)
      WRITE(6,1) INSNPS
    1 FORMAT(//,1H ,52HINSTANTANEOUS NUMBER OF NEUTRONS PRODUCED PER SEC
     +OND,4X,E11.4)
      WRITE(6,2) AVENPS
    2 FORMAT(/,1H ,46HAVERAGE NUMBER OF NEUTRONS PRODUCED PER SECOND,4X,
     +E11.4)
      RETURN
      END
```

```
      **SUBROUTINE GRFLUX(AVENPS,NOEGPS,GPFLUX,BFLUX)**
      SUBROUTINE GRFLUX(AVENPS,NCEGPS,GPFLUX,BFLUX)
C
C
C THIS PROGRAM CALCULATES UP TO 50 GROUP FLUXES
C
C
C     ARGUMENTS IN-AVENPS,NCEGPS,BFLUX
C     ARGUMENTS OUT-GPFLUX
C
C
      DIMENSION GPFLUX(NCEGPS),BFLUX(NOEGPS)
      DO 1 I = 1,NCEGPS
      GPFLUX(I) = BFLUX(I)*AVENPS
    1 CONTINUE
      RETURN
      END
```

```
      **SUBROUTINE WTFLUX(BFLUX,GP,CONST,FCONST,NCEGPS,NF,NG)**

      SUBROUTINE WTFLUX(BFLUX,GP,CONST,FCONST,NCEGPS,NF,NG)
C
C
CTHIS PROGRAM DETERMINES THE CONSTANTS ASSOCIATED WITH THE ASSUMED
CWEIGHTING FUNCTIONS
C
C
C     ARGUMENTS IN-BFLUX,GP,NOEGPS,NF,NG
C     ARGUMENTS OUT-CONST,FCONST
C
C
      DIMENSION BFLUX(NCEGPS),GP(NG),CONST(NCEGPS),FCONST(NOEGPS)
      DO 106 JC=1,NG
      GP(JC)=GP(JC)*1.000E6
  106 CONTINUE
CCALCULATE THE THERMAL GROUP CONSTANT
      ICCUNT=1
      SUMINT=0.0
      WIDTH=(GP(2)-GP(1))/200.0
      ZINCR=WIDTH/2.0
      DPVAR=GP(1)+ZINCR
      DO 1 I=1,200
      PARSUM=SQRT(DPVAR)*EXP(-(DPVAR/0.025))*WIDTH
      SUMINT=SUMINT+PARSUM
      DPVAR=DPVAR+WIDTH
    1 CONTINUE
      CONST(ICOUNT)=BFLUX(ICCUNT)/SUMINT
      IF(NOEGPS.EQ.1) GO TO 1000
      IF(NOEGPS.EQ.2) GO TO 150
CCALCULATE THE 1/E GROUP CONSTANT(S)
      ICCUNT=ICCUNT+1
      IEPTH=0
    2 SUMINT=0.0
      WIDTH=(GP(ICCUNT+1)-GP(ICCUNT))/200.0
      ZINCR=WIDTH/2.0
      DPVAR=GP(ICCUNT)+ZINCR
      DO 3 I=1,200
      PARSUM=(1.0/DPVAR)*WIDTH
      SUMINT=SUMINT+PARSUM
      DPVAR=DPVAR+WIDTH
    3 CONTINUE
      CONST(ICCUNT)=BFLUX(ICCUNT)/SUMINT
      IEPTH=IEPTH+1
      NEPTH=NOEGPS-1-NF
      IF(IEPTH.EQ.NEPTH) GO TO 150
      ICCUNT=ICCUNT+1
      GO TO 2
  100 ICCUNT=ICCUNT+1
CTHE 1 GROUP FAST SPECTRUM WILL BE MATCHED TO THE MAXWELLIAN AT THE
CGROUP BOUNDARY
      FLUXBD=CONST(1)*SQRT(GP(2))*EXP(-(GP(2)/0.025))
      ZNUMER=GUESSI*BFLUX(NOEGPS)/FLUXBD
      ITER=1
    6 FXSUBI=ZNUMER/(EXP((GP(3)-GP(2))/GUESSI)-1)-GUESSI
      XSUBI=GUESSI
      FPRIME=ZNUMER*(GP(3)-GP(2))*EXP((GP(3)-GP(2))/GUESSI)/(GUESSI**2*(
     *EXP((GP(3)-GP(2))/GUESSI)-1)**2)-1
```

```
      **SUBROUTINE WTFLUX(BFLUX,GP,CONST,FCONST,NOEGPS,NF,NG)**
      XSBIP1=XSLBI-FXSLBI/FPRIME
      DIFF=ABS(XSBIP1-GUESSI)
      IF(DIFF.LT.0.1) GO TO 5
      ITER=ITER+1
      IF(ITER.GT.100) GO TO 75
      GLESSI=XSBIP1
      GO TO 6
   75 WRITE(6,76)
   76 FORMAT(1H ,114H****PROGRAM ABORTED****THE TRANSCENDENTAL EQUATION
     +USED TO MATCH THE THERMAL GROUP TO FAST GROUP IS NOT CONVERGING)
      WRITE(6,27)
      STOP
    5 B=XSBIP1
      A=BFLUX(NOEGPS)/(B*(1-EXP((GP(2)-GP(3))/B)))
      CONST(ICOUNT)=A
      FCONST(ICOUNT)=B
      GO TO 1000
  150 IF(NF.GT.1) GO TO 160
C THE 1 GROUP FAST SPECTRUM WILL BE MATCHED TO THE LAST 1/E GROUP
      ITER=1
      NGP1=NOEGPS+1
      ZNUMER=GUESSI=(BFLUX(NOEGPS)*GP(NOEGPS)/CONST(ICOUNT))
   51 FXSUBI=ZNUMER/(EXP((GP(NGP1)-GP(NOEGPS))/GUESSI)-1)-GUESSI
      XSUBI=GUESSI
      FPRIME=ZNUMER*(GP(NGP1)-GP(NOEGPS))*EXP((GP(NGP1)-GP(NOEGPS))/GUES
     +SI)/(GUESSI**2*(EXP((GP(NGP1)-GP(NOEGPS))/GUESSI)-1)**2)-1
      XSBIP1=XSUBI-FXSUBI/FPRIME
      DIFF=ABS(XSBIP1-GUESSI)
      IF(DIFF.LT.0.1) GO TO 92
      ITER=ITER+1
      IF(ITER.GT.100) GO TO 25
      GLESSI=XSBIP1
      GO TO 91
   92 B=XSBIP1
      A=BFLUX(NOEGPS)/(B*(1-EXP((GP(NOEGPS)-GP(NGP1))/B)))
      CONST(NOEGPS)=A
      FCONST(NOEGPS)=B
      GO TO 1000
  160 ICP1=ICOUNT+1
C THE FIRST OF 2 OR MORE FAST GROUPS WILL BE MATCHED TO THE LAST 1/E
GROUP
      ICP2=ICOUNT+2
      ITER=1
      ZNUMER=GUESSI=(BFLUX(ICP1)*GP(ICP1)/CONST(ICOUNT))
   20 FXSUBI=ZNUMER/(EXP((GP(ICP2)-GP(ICP1))/GUESSI)-1)-GUESSI
      XSUBI=GUESSI
      FPRIME=ZNUMER*(GP(ICP2)-GP(ICP1))*EXP((GP(ICP2)-GP(ICP1))/GUESSI)/
     +(GUESSI**2*(EXP((GP(ICP2)-GP(ICP1))/GUESSI)-1)**2)-1
      XSBIP1=XSUBI-FXSUBI/FPRIME
      DIFF=ABS(XSBIP1-GUESSI)
      IF(DIFF.LT.0.1) GO TO 30
      ITER=ITER+1
      IF(ITER.GT.100) GO TO 25
      GLESSI=XSBIP1
      GO TO 20
   25 WRITE(6,26)
   26 FORMAT(1H ,116H****PROGRAM ABORTED****THE TRANSCENDENTAL EQUATIONS
     **WHICH MATCH 1/E TO FAST OR FAST TO FAST GROUPS ARE NOT CONVERGING)
```

```
      ** SUBROUTINE WFLUX(BFLUX,GP,CCNST,FCCNST,NOEGPS,NF,NGI**

      WRITE(6,27)
   27 FORMAT(1H ,75HCST LIKELY PROBLEM IS UNREALISTIC OR UNUSAL FLUX
     +AND GROUP BOUNDARY DATA)
      STOP
   30 E=XSBIP1
      A=BFLUX(ICP1)/(B*(1-EXP((GP(ICP1)-GP(ICP2))/B)))
      ICCUNT=ICCUNT+1
      ICP1=ICCUNT+1
      ICP2=ICCUNT+2
      CCNST(ICCUNT)=A
      FCCNST(ICCUNT)=B
      IF(ICOUNT.EQ.NOEGPS) GO TO 1000
      ZNUMER=BFLUX(ICP1)/CCNST(ICOUNT)
      ITER=1
      GO TO 20
 1000 WRITE(6,11)
   11 FORMAT(//,1H ,63H      INTEGRALS OF THE FITTING FUNCTION OVER THE G
     +ROUP INTERVALS)
      WRITE(6,12)
   12 FORMAT(1H ,74HGROUP  LOWER BOUNDARY UPPER BOUNDARY REQUIRED INTEGR
     +AL CALCULATED INTEGRAL,/)
CCALCULATE THE THERMAL INTEGRAL
      ICCUNT=1
      SUMINT=0.0
      WIDTH=(GP(2)-GP(1))/200.0
      ZINCR=WIDTH/2.0
      DPVAR=GP(1)+ZINCR
      DO 21 I=1,200
      PARSUM=CONST(1)*SQRT(DPVAR)*EXP(-(DPVAR/0.025))*WIDTH
      SUMINT=SUMINT+PARSUM
      DPVAR=DPVAR+WIDTH
   21 CONTINUE
      IF(ICCUNT.EQ.NOEGPS) GO TO 109
      IF(ICCUNT.EQ.2) GO TO 46
      JJ=NG-ICOUNT
      WRITE(6,22) JJ,GP(ICCUNT),GP(ICCUNT+1),BFLUX(ICOUNT),SUMINT
   22 FORMAT(1H ,2X,I2,5X,E10.3,4X,E10.3,5X,E10.3,9X,E10.3)
      ICOUNT=ICCUNT+1
CCALCULATE THE 1/E INTEGRAL(S)
   45 SUMINT=0.0
      WIDTH=(GP(ICCUNT+1)-GP(ICOUNT))/200.0
      ZINCR=WIDTH/2.0
      DPVAR=GP(ICOUNT)+ZINCR
      DO 23 I=1,200
      PARSUM=(CCNST(ICCUNT)/DPVAR)*WIDTH
      SUMINT=SUMINT+PARSUM
      DPVAR=DPVAR+WIDTH
   23 CONTINUE
      JJ=NG-ICOUNT
      WRITE(6,22) JJ,GP(ICCUNT),GP(ICOUNT+1),BFLUX(ICOUNT),SUMINT
      ICOUNT=ICCUNT+1
      IF(ICOUNT.LE.NEPTH+1) GO TO 45
      GO TO 16
   46 JJ=2
      WRITE(6,22) JJ,GP(ICCUNT),GP(ICOUNT+1),BFLUX(ICOUNT),SUMINT
      IF(NG.EQ.2) ICCUNT=ICCUNT+1
   16 SUMINT=0.0
CCALCULATE THE FAST INTEGRAL(S)
```

```
      **SUBROUTINE WFLL(BFLUX,GP,CCNST,FCCNST,NCEGPS,NF,NG)**
      WIDTH=(GP(ICCUNT+1)-GP(ICOUNT))/200.0
      ZINCR=WIDTH/2.0
      DPVAR=GP(ICOUNT)+ZINCR
      DO 24 I=1,200
      PARSUM=CCNST(ICCUNT)*EXP((DPVAR-GP(ICCUNT+1))/FCONST(ICOUNT))*WIDT
     +H
      SUMINT=SUMINT+PARSUM
      DPVAR=DPVAR+WIDTH
   24 CONTINUE
      JJ=NG-ICCUNT
      WRITE(6,22) JJ,GP(ICCUNT),GP(ICCUNT+1),BFLUX(ICCUNT),SUMINT
      ICCUNT=ICCUNT+1
      IF(ICCUNT.LE.NCEGPS) GO TO 16
      GO TO 15
  109 JJ=1
      WRITE(6,22) JJ,GP(ICCUNT),GP(ICCUNT+1),BFLUX(ICCUNT),SUMINT
   15 DO 105 JC=1,NG
      GP(JC)=GP(JC)/1.000E6
  105 CONTINUE
      RETURN
      END
```

```
      **SUBROUTINE ACTVAT(TIMACT,SYSACT,SUMACT,GPFLUX,PRCDNU,PRCDHL,GPXSEC**
      SUBROUTINE ACTVAT(TIMACT,SYSACT,SUMACT,GPFLUX,PRCDNU,PRODHL,GPXSEC
     +,IDNO,MASS,STAES,GP,CONST,FCONST,BFLUX,NONUCL,NCEGPS,NOPAS,NG,NF)
C
C
CTHIS PROGRAM CALCULATES ACTIVITIES DUE TO NEUTRON INDUCED REACTIONS
C
C
C     ARGUMENTS IN-GPFLUX,GPXSEC,IDNO,MASS,STAES,GP,CONST,FCONST,BFLUX
C     ARGUMENTS IN-NONUCL,NCEGPS,NOPAS,NG,NF
C     ARGUMENTS OUT-TIMACT,SYSACT,PRCDNU,PRCDHL
C
C
      DIMENSION TIMACT(NONUCL,13,NOPAS),SYSACT(NOPAS),SUMACT(NOPAS),GPFL
     +UX(NCEGPS),PRCDNU(NONUCL,13),PRODHL(NONUCL,13),GP(NG),GPXSEC(NOEGP
     +S),CCNST(NCEGPS),FCCNST(NCEGPS),BFLUX(NCEGPS),IDNO(NONUCL),MASS(NO
     +NUCL),STAES(NOPAS)
      CCMMCN A,V,F,CF,T1,T2,N,R,D,HT,OPTION,IGEOM
      REAL MASS
      WRITE(6,1C2)
  1C2 FCRMAT(///,1H ,57HTHE FCLLOWING OUTPUT DATA ARE CALCULATED ACTIVIT
     +IES IN BQ)
      I=1
      J=0
C INITIALIZE FLAGS AND ARRAYS
      CO 40 M=1,NCNUCL
      DC 40 MM=1,13
      PRCDNU(M,MM)=0.0
      FRCDHL(M,MM)=0.0
   40 CCNTINUE
      OC 41 M=1,NCNUCL
      CC 41 MM=1,13
      DC 41 MMM=1,NCPAS
      TIMACT(M,MM,MMM)=0.0
   41 CCNTINUE
      OC 22 JN=1,NCPAS
      SYSACT(JN)=SUMACT(JN)=0.0
   22 CCNTINUE
      KFLAG=LFLAG=C
CPRINT HEADINGS
      WRITE(6,37)(STAES(K),K=1,NCPAS)
   37 FCRMAT(/,1H ,4X,7HTIME(H),4X,10(E8.3,1X),2X,2(E8.3,2X))
      WRITE(6,1)
    1 FCRMAT(1H ,125HTARGET PRODUCT  ACT(T1)  ACT(T2)  ACT(T3)  ACT(T4)
     + ACT(T5)  ACT(T6)  ACT(T7)  ACT(T8)  ACT(T9)  ACT(T10)  ACT(T11)
     +ACT(T12))
   24 READ(T,25) ITAPE,TARMAS
   25 FCRMAT(I6,7X,E11.4)
      IF(ITAPE.EQ.IDNO(I)) GO TO 7
      IF(ITAPE.GT.IDNO(I)) GO TO 44
      CALL PCSITCN
      GC TO 24
CTHE ABCVE STATEMENTS SEARCH THE TAPE FOR A MATCHING TARGET ID NUMBER
   44 WRITE(6,45) IDNO(I)
   45 FORMAT(////,1H ,31HATTENTION-TARGET NUCLIDE NUMBER,I6,88HDOES NOT
     +EXIST IN CATA LIBRARY-THEREFORE-NO CONTRIBUTION FROM THIS NUCLIDE
```

```
      **SUBROUTINE ACTVAT(TIMACT,SYSACT,SUMACT,GPFLUX,PRCDNU,PRCDHL,GPXSEC**
     +IN DCSE CALCS,/////)
      IF(ITAPE.EQ.IDNO(I+1)) GO TO 46
      I=I+1
      CALL PCSITCN
CPRINT HEADINGS
      WRITE(6,37)(STAFS(K),K=1,NOPAS)
      WRITE(6,1)
      GO TO 24
   46 I=I+1
CPRINT HEADINGS
      WRITE(6,37)(STAFS(K),K=1,NOPAS)
      WRITE(6,1)
    7 J=J+1
      READ(7,8) PRCDNU(I,J),PRCDHL(I,J)
      READ(7,99) AP
   99 FORMAT(I3)
      IF(ITAPE.EQ.IDAC(NONUCL)) LFLAG=1
      IF(PRCDHL(I,J).GT.0.1000E+50) GO TO 50
    8 FORMAT (20X,E12.4,12X,E12.4)
CCALL AVRAGE TO CALCULATE GROUP CROSS SECTIONS
      CALL AVRAGE(GP,SEC,CCNST,FCCNST,BFLUX,GP,NCEGPS,NG,NF,NP)
      IF(GPXSEC(1).GT.0.5CCE+49) GO TO 103
CINITIALIZE SUM1A AND SUM2A
      SUM1A=SUM2A=0.0
CBEGIN ACTIVATION CALCULATIONS
      DO 11 JL=1,N
      TEMPS=EXP(-((0.6...)*((N-JL)*((T1*3600.0)+(T2*3600.0)))/PRCDHL(I,J)
     +))
      SUM1A=SUM1A+TEMPS
   11 CONTINUE
      DO 12 L=1,NOEGPS
      TEMPS=SUM1A*(1.0E-24)*GPXSEC(L)*GPFLUX(L)
      SUM2A=SUM2A+TEMPS
   12 CONTINUE
CCALCULATE ACTIVITY AT T=0
      ACT=(MASS(I)*1000.0/TARMAS)*6.023E+23*SUM2A*(1-EXP(-0.693*3600.0*T
     +1/PRCDHL(I,J)))
CCALCULATE ACTIVITY AS A FUNCTION OF TIME
      DO 13 K=1,NOPAS
      TIMACT(I,J,K)=ACT*EXP(-(STAFS(K)*0.693*3600.0/PRCDHL(I,J)))
CSUM ACTIVITIES
      SUMACT(K)=SUMACT(K)+TIMACT(I,J,K)
   13 CONTINUE
CPRINT ACTIVITIES
  103 WRITE(6,20) IDNO(I),IFIX(PRCDNU(I,J)),(TIMACT(I,J,K),K=1,NOPAS)
   20 FORMAT(1X,1X,I5,2X,I5,2X,10(E8.3,1X),2X,2(E8.3,2X))
CBEGIN INTERROGATION OF THE ACTLNFE LIBRARY
      READ(7,25) ITAPE,TARMAS
      IF(ECF(T).NE.0) KFLAG=1
      IF(KFLAG.EQ.1) GO TO 27
      IF(LFLAG.EQ.1.AND.ITAPE.NE.IDNO(NONUCL)) GO TO 27
      IF(ITAPE.EQ.IDAC(NONUCL)) LFLAG=1
      IF(ITAPE.NE.IDAC(I)) GO TO 27
      GO TO 7
CPRINT HEADINGS
   27 WRITE(6,37)(STAFS(K),K=1,NOPAS)
```

```
      **SUBROUTINE ACTVAT(IMACT,SYSACT,SUMACT,GPFLUX,PRODNU,PRODHL,GPXSEC**
      WRITE(6,14)
   14 FORMAT(1H ,125H     TARGET        ACT(T1)   ACT(T2)   ACT(T3)   ACT(T4)
     + ACT(T5)   ACT(T6)   ACT(T7)   ACT(T8)   ACT(T9)   ACT(T10)   ACT(T11)
     +ACT(T12))
CPRINT SUM OF ACTIVITIES DUE TO A PARTICULAR TARGET NUCLIDE
      WRITE(6,15) IDNC(I),(SUMACT(K),K=1,NCPAS)
   15 FORMAT(1X,5X,I5,5X,10(E8.3,1X),2X,2(E8.3,2X))
      DO 10 K=1,NCPAS
      SYSACT(K)=SYSACT(K)+SUMACT(K)
   10 CONTINUE
      IF(ITAPE.EQ.IDNO(I+1)) GO TO 32
      IF(LFLAG.EQ.1) GO TO 16
   32 DO 23 JN=1,NCPAS
      SUMACT(JN)=0.0
   23 CONTINUE
      IF(ITAPE.EQ.IDNC(I+1)) GO TO 31
      IF(ITAPE.GT.IDNO(I+1)) GO TO 47
      I=I+1
      CALL PCSITCN
CPRINT HEADINGS
      WRITE(6,37)(STAFS(K),K=1,NCPAS)
      WRITE(6,1)
      J=0
      GO TO 24
   47 WRITE(6,45) IDNC(I+1)
      CALL PCSITCN
      I=I+2
      IF(ITAPE.GT.IDNO(I)) WRITE(6,45) IDNC(I)
      IF(ITAPE.GT.IDNC(I)) I=I+1
CPRINT HEADINGS
      WRITE(6,37)(STAFS(K),K=1,NOPAS)
      WRITE(6,1)
      J=0
      GO TO 24
CPRINT HEADINGS
   31 WRITE(6,37)(STAFS(K),K=1,NCPAS)
      WRITE(6,1)
      I=I+1
      J=0
      GO TO 7
CPRINT HEADINGS
   16 WRITE(6,37)(STAFS(K),K=1,NOPAS)
      WRITE(6,21)
   21 FORMAT(1H ,125H     SYSTEM        ACT(T1)   ACT(T2)   ACT(T3)   ACT(T4)
     + ACT(T5)   ACT(T6)   ACT(T7)   ACT(T8)   ACT(T9)   ACT(T10)   ACT(T11)
     +ACT(T12))
CPRINT SYSTEM ACTIVITY
      WRITE(6,51)(SYSACT(K),K=1,NOPAS)
   51 FORMAT(1X,15X,10(E8.3,1X),2X,2(E8.3,2X))
      IF(KFLAG.EQ.0) GO TO 42
      RETURN
   42 CALL PCSITCN
   43 READ(7,25) ITAFE,TARMAS
      IF(EOF(7).NE.0) RETURN
```

```
      **SUBROUTINE  ACTVATIT IMACT,SYSACT,SUMACT,GPFLUX,PRODNU,PRODHL,GPXSEC**
      CALL  POSITON
      GO TO 43
   50 CALL  POSITZ(NP)
      GO TO 103
      END
```

```
      **SUBROUTINE AVRAGE(GPXSEC,CONST,FCONST,BFLUX,GP,NOEGPS,NG,NF,NP)**
      SUBROUTINE AVRAGE(GPXSEC,CONST,FCONST,BFLUX,GP,NOEGPS,NG,NF,NP)
C
C
C THIS PROGRAM CALCULATES FLUX WEIGHTED GROUP CROSS SECTIONS BY AVERAGING
C MICROSCOPIC CROSS SECTION DATA
C
C
C     ARGUMENTS IN-CONST,FCONST,BFLUX,GP,NOEGPS,NG,NF,NP
C     ARGUMENTS OUT-GPXSEC
C
C
      DIMENSION GPXSEC(NOEGPS),CONST(NOEGPS),FCONST(NOEGPS),BFLUX(NOEGPS
     +),GP(NG),CS(150),EN(150),CS1(201),E1(201)
C INITIALIZE ARRAY GPXSEC
      DO 41 JK=1,NOEGPS
      GPXSEC(JK)=0.0
   41 CONTINUE
C LOAD ENERGY AND CROSS SECTION DATA FROM THE ACTINFE LIBRARY
      READ(7,3)(EN(I),CS(I),I=1,NP)
    3 FORMAT(6(E11.4))
C DETERMINE IF THE ENERGY OF THE FIRST ENERGY-CROSS SECTION PAIR READ
C FROM THE ACTINFE LIBRARY IS BELOW THE GREATEST GROUP BOUNDARY
      IF(EN(1)-GP(NG)) 105,102,102
  105 CALL SMOOTH(GE,NG,EN,CS,NP,E1,CS1,NE1)
      I=1
      J=1
      SUMINT=0.0
C DETERMINE IF THE ENERGY AT WHICH THE CROSS SECTION WAS MEASURED IS
C BELOW THE NEXT GROUP BOUNDARY
   53 IF(E1(I).LT.GP(J+1)) GO TO 51
      GPXSEC(J)=0.0
      J=J+1
      GO TO 50
   52 IF(E1(I).LT.GP(J+1)) GO TO 51
C CALCULATE THE GROUP CROSS SECTION
      GPXSEC(J)=SUMINT/BFLUX(J)
      IF(J.EQ.NOEGPS) GO TO 106
      J=J+1
      SUMINT=0.0
      GO TO 52
   51 IF(J.EQ.1) GO TO 1000
      IF(NOEGPS.GE.3.AND.J.GT.1.AND.J.LE.(NOEGPS-NF)) GO TO 2000
      GO TO 3000
 1000 SUMDEL=0.0
C AVERAGE MICROSCOPIC CROSS SECTION DATA OVER THERMAL ENERGIES
      WIDTH=(E1(I+1)-E1(I))/200.0
      ZINCR=WIDTH/2.0
      DPVAR=E1(I)+ZINCR
      DO 1001 KK=1,200
      PARSUM=(CONST(J)*SQRT(DPVAR*1.0E6)*EXP(-(DPVAR*1.0E6/0.025))*((CS1
     +(I+1)-CS1(I))*(DPVAR-E1(I))/(E1(I+1)-E1(I))+CS1(I)))*WIDTH*1.0E6
      SUMDEL=SUMDEL+PARSUM
      DPVAR=DPVAR+WIDTH
 1001 CONTINUE
      SUMINT=SUMINT+SUMDEL
      I=I+1
      GO TO 52
```

```
      **SUBROUTINE AVRAGE(GPXSEC,CONST,FCONST,BFLUX,GP,NCEGPS,NG,NF,NPI**
 2CCC SUMDEL=0.C
CAVERAGE MICROSCOPIC CROSS SECTION DATA OVER INTERMEDIATE ENERGIES
      WIDTH=(E(I+1)-E(I))/200.0
      ZINCR=WIDTH/2.C
      DPVAR=E(I)+ZINCR
      DO 2001 KK=1,200
      PARSUM=(CONST(J)/(DPVAR*1.0E6))*WIDTH*1.0E6*((CS1(I+1)-CS1(I))*(DP
     +VAR-E(I))/(E(I+1)-E(I))+CS1(I))
      SUMDEL=SUMDEL+PARSUM
      DPVAR=DPVAR+WIDTH
 2001 CONTINUE
      SUM(I)=SUM(I)+SUMDEL
      .S.)3
      GO TO 32
 3CCC SUMDEL=0.0
CAVERAGE MICROSCOPIC CROSS SECTION DATA OVER FAST ENERGIES
      WIDTH=(E(I+1)-E(I))/200.0
      ZINCR=WIDTH/2.C
      DPVAR=E(I)+ZINCR
      DO 3001 KK=1,200
      PARSUM=CONST(J)*EXP((DPVAR-GP(J+1))*1.0E6/FCONST(J))*WIDTH*1.0E6*(
     +(CS1(I+1)-CS1(I))*(DPVAR-E(I))/(E(I+1)-E(I))+CS1(I))
      SUMDEL=SUMDEL+PARSUM
      DPVAR=DPVAR+WIDTH
 3001 CONTINUE
      SUMINT=SUMINT+SUMDEL
      I=I+1
      GO TO 32
  1C2 GPXSEC(I)=1.DE+50
  106 READ(7,51 IDUMB
    5 FORMAT(71X,I1)
      RETURN
      END
```

```
      **SUBROUTINE SMOOTH(X1,NX1,X2,Y2,NX2,X3,Y3,NX3)**

      SUBROUTINE SMOOTH(X1,NX1,X2,Y2,NX2,X3,Y3,NX3)
C
C
CTHIS PROGRAM CALCULATES CROSS SECTIONS AT GROUP BOUNDARIES BY LINEAR
CINTERPOLATION
C
C
C     ARGUMENTS IN-GE,NG,EN,CS,AP
C     ARGUMENTS OUT-E1,CS1,NE1
C
C
      DIMENSION X1(1),X2(1),Y2(1),X3(1),Y3(1)
      K=1
   12 IF(X2(K)-X1(1)) 20,11,13
   20 K=K+1
      GO TO 12
   11 X3(1)=X2(K)
      Y3(1)=Y2(K)
      K=K+1
      L=2
      I1=2
      GO TO 14
   13 I1=2
      IF(X2(K).LT.X1(I1)) GO TO 11
   24 I1=I1+1
      IF(X2(K).LT.X1(I1)) GO TO 25
      GO TO 24
   25 X3(1)=X2(K)
      Y3(1)=Y2(K)
      K=K+1
      L=2
   14 DO 1 I=I1,NX1
    5 IF(X2(K)-X1(I)) 2,3,4
    2 X3(L)=X2(K)
      Y3(L)=Y2(K)
      L=L+1
      K=K+1
      IF(K-NX2) 5,5,10
    3 X3(L)=X2(K)
      Y3(L)=Y2(K)
      L=L+1
      K=K+1
      IF(K-NX2) 1,1,10
    4 X3(L)=X1(I)
      Y3(L)=Y3(L-1)+((Y2(K)-Y3(L-1))/(X2(K)-X3(L-1)))*(X3(L)-X3(L-1))
      L=L+1
    1 CONTINUE
   10 NX3=L-1
      RETURN
      END
```

```
      **SUBROUTINE ADVAN(NDMODE)**

      SUBROUTINE ADVAN(NDMODE)
C
C
C THIS PROGRAM IS USED FOR POSITIONING THE FILE MARKER IN THE LEVDEC
CLIBRARY
C
C
C     ARGUMENTS IN-NDMODE
C     ARGUMENTS OUT-NONE
C
C
      DO 305 JB=1,NDMODE
      READ(7,202) IDUMB
  202 FORMAT(38X,I2)
  305 CONTINUE
      RETURN
      END


      **SUBROUTINE POSITON**

      SUBROUTINE POSITON
C
C
C THIS PROGRAM IS USED FOR POSITIONING THE FILE MARKER IN THE ACTLEMFE
CLIBRARY
C
C
C     ARGUMENTS IN-NONE
C     ARGUMENTS OUT-NONE
C
C
      READ(7,30) NP
   30 FORMAT(/,I3)
      NP1=INT(FLOAT(NP)/3.000+0.800)
      DO 52 ICOUNT=1,NP1
      READ(7,53) REALDUM
   53 FORMAT(E11.4)
   52 CONTINUE
      READ(7,5) IDUMB
    5 FORMAT(71X,I1)
      RETURN
      END
```

```
      **SUBROUTINE POSIT2(NP)**

      SUBROUTINE POSIT2(NP)
C
C
C THIS PROGRAM IS USED FOR POSITIONING THE FILE MARKER IN THE ACTLEMFE
C LIBRARY
C
C
C     ARGUMENTS IN-NP
C     ARGUMENTS OUT-NONE
C
C
      NP1=INT(FLOAT(NP)/3.000+0.800)
      DO 52 ICOUNT=1,NP1
      READ(7,53) REALDUM
   53 FORMAT(E11.4)
   52 CONTINUE
      READ(7,51) IDUME
    5 FORMAT(71X,I1)
      RETURN
      END
```

```
      **SUBROUTINE DCSRTE(TIMACT,PRCDNU,NONUCL,NOPAS,DOSSUM,SYSDOS,IDNO,ST**

      SUBROUTINE DOSRTE(TIMACT,PRCDNU,NONUCL,NOPAS,DOSSUM,SYSDCS,IDNO,ST
     +AFS,PRCDHL,DCSNUC)
C
C
CTHIS PRCGRAM CALCULATES DCSE RATES AS A FUNCTION OF COMPONENT COMPOSIT-
CION,GECMETRY,AND TIME AFTER SHUTDOWN
C
C
C     ARGUMENTS IN-TIMACT,PRCDNU,NONUCL,NOPAS,DCSSUM,SYSDCS,IDNO,STAFS
C     ARGUMENTS IN-PRCDHL,DCSNUC
C     ARGUMENTS OUT-NONE
C
C
      DIMENSICN TIMACT(NONUCL,13,NOPAS),PRCDNU(NONUCL,13),DOSSUM(NOPAS),
     +SYSCDS(NOPAS),IDNO(NONUCL),STAFS(NOPAS),PRODHL(NONUCL,13),DCSNUC(N
     +CPAS),CCSCAU(390),NT(390),NP(390),AC(390),H(390),ND(390),DOSPRD(39
     +0),IDEKM(30),IDALG(30),DLEVL(30),PRCB(30)
      COMMCN A,V,E,CE,T1,T2,N,R,D,HT,CPTION,IGEOM
      REAL MULT
      WRITE(6,100)
  100 FCRMAT(///,1H ,61HTHE FCLLOWING OUTPUT DATA ARE CALCULATED DOSE RA
     +TES IN MRAD/H,/)
CINITIALIZE ARRAYS
      DC 1 M=1,NCPAS
      SYSDCS(M)=C.0
      DCSNUC(M)=0.0
      DCSSUM(M)=C.C
    1 CCNTINUE
CINITIALIZE ARRAYS
      DC 101 I=1,39C
      NT(I)=NP(I)=ND(I)=C
      AC(I)=DCSPRC(I)=H(I)=DCSCAU(I)=0.0
  101 CONTINUE
CINITIALIZE ARRAYS
      CC 2 I=1,NCNUCL
      DC 2 J=1,13
      K=(I-1)*13+J
      NP(K)=IFIX(PRCDNU(I,J))
      H(K)=PRODHL(I,J)
      AC(K)=TIMACT(I,J,1)
      NT(K)=IDNO(I)
    2 CONTINUE
CARRANGE TARGET ID NUMBERS,PARENT ACTIVITIES,AND PARENT HALF-LIFES IN
CORDER CF INCREASING PARENT ID NUMBER PRICR TC SEARCHING LEVDEC FOR
CPARENT DECAY INFORMATION
      NC=NCNUCL*13-1
      DC 3 J=1,ND
      NTEMP1=NP(J+1)
      NTEMP2=NT(J+1)
      ZTEMP3=AC(J+1)
      ZTEMP4=H(J+1)
      DC 4 K=1,J
      I=J+1-K
      IF(NTEMP1.GE.NP(I)) GO TO 5
      NP(I+1)=NP(I)
      NT(I+1)=NT(I)
      AC(I+1)=AC(I)
```

```
      **SUBROUTINE DOSRTE(TIMACT,PRODNU,NONUCL,NOPAS,DOSSUM,SYSDCS, IDNO,ST**
      H(I+1)=H(I)
    4 CONTINUE
      I=0
    5 AP(I+1)=ATEMP1
      NT(I+1)=ATEMP2
      AC(I+1)=ZTEMP3
      H(I+1)=ZTEMP4
    3 CONTINUE
CSEARCH THE NP ARRAY FOR THE FIRST NON-ZERO PARENT ID NUMBER
      L=1
    7 IF(NP(L).NE.0) GO TO 205
      L=L+1
      GO TO 7
CMARK THE BEGINNING OF THE PARENT NUCLIDE LIST-THIS LIST IS IN ASCENDIN
CG ORDER
  205 IBASE=L
CBEGIN INTERROGATION OF THE LEVDEC LIBRARY FOR A PARTICULAR PARENT ID
CNUMBER
  200 READ(7,16) ITAPE,HALFLF,NOMODE
   16 FORMAT(I6,20X,E11.4,I3)
      IF(ECF(7).NE.0) GO TO 200
      IF(ITAPE.EQ.AP(L)) GO TO 17
      IF(ITAPE.GT.NP(L)) GO TO 280
      CALL ADVANCE(NOMODE)
      GO TO 200
  280 CALL ADVANCE(NOMODE)
   28 WRITE(6,29) AP(L)
   29 FORMAT(/////,1H ,32HATTENTION-PRODUCT NUCLIDE NUMBER,I6,90HDOES NO
     +T EXIST IN DECAY LIBRARY-THEREFORE-NO CONTRIBUTION FROM THIS NUCL
     +IDE IN DOSE CALCS)
      WRITE(6,37) NT(L)
   37 FORMAT(1H ,19HTHE TARGET ATOM WAS,I5,/////)
      L=L+1
CDISCONTINUE IF THE LAST PARENT NUCLIDE HAS BEEN ADDRESSED
      IF(NP(L).EQ.0) GO TO 813
      IF(ITAPE.GT.AP(L)) GO TO 28
      GO TO 200
CCHECK TO SEE IF NP(L) EQUALS NP(L+1)-THIS OCCURS WHEN THE SAME PARENTS
CARE PRODUCED FROM DIFFERENT TARGET NUCLIDES. IN THIS CASE, THE DECAY
CINFORMATION IS STORED IN VARIOUS HOLDING ARRAYS AND IS INTERROGATED AS
CMANY TIMES AS THERE ARE ADDITIONAL LIKE PARENTS. THIS NEGATES HAVING
CTO RE-READ LEVDEC FOR EACH LIKE PARENT.
   17 IF(NP(L).EQ.NP(L+1)) GO TO 800
CCOMPARE HALF LIFES
  313 IF(H(L)/HALFLF.LT.0.8.OR.H(L)/HALFLF.GT.1.2) GO TO 19
      IF(HALFLF.EQ.1.00E+50.OR.AC(L).EQ.0.0) GO TO 19
      INODE=0
  302 INODE=INODE+1
CBEGIN INTERROGATION OF THE LEVDEC LIBRARY
      READ(7,300) IDEKMO,IDAUGH,DLEVEL,MULT
  300 FORMAT(38X,I2,I7,E11.4,E12.4)
CIF DECAY IS BY BETA MINUS,STORE DAUGHTER ID NUMBER
      IF(IDEKMO.EQ.9) ND(L)=IDAUGH
      IF(IDEKMO.EQ.7.OR.IDEKMO.EQ.9) GO TO 301
      IF(IDEKMO.EQ.8) GO TO 310
  303 IF(INODE.EQ.NOMODE) GO TO 304
      GO TO 302
  301 IF(DLEVEL.EQ.0.0) GO TO 303
```

```
      **SUBROUTINE DCSRTE(TIMACT,PRCONU,NONUCL,NOPAS,DOSSUM,SYSDOS,IDNO,ST**
      CALL CURVE(DLEVEL,FLUXE)
      CALL CALC(AC(L),DCSPRC(L),MULT,FLUXE)
      GC TC 303
  210 IF(CLEVEL.EQ.0.C) GO TO 311
      CALL CURVE(DLEVEL,FLUXE)
      CALL CALC(AC(L),DCSPRO(L),MULT,FLUXE)
CIN ANY CASE ADD IN TWO C.511 MEV ANNILIHATION GAMMAS
  311 CC 312 IBETA=1,2
      CALL CURVE(C.511,FLUXE)
      CALL CALC(AC(L),DOSPRO(L),MULT,FLUXE)
  312 CONTINUE
      GC TC 303
   19 CALL ACVAN(NCMCDE)
CREAD THE NEXT LEVCEC NUCLIDE
  204 READ(7,16) ITAPE,HALFLF,NDMODE
      IF(ECF(7).NE.0) GO TO 304
      IF(ITAPE.EC.NP(L)) GC TC 313
      IF(ITAPE.EQ.NP(L+1)) GO TO 21
CIISCCNTINUE IF THE LAST NUCLIDE HAS BEEN ADDRESSED
      IF(NP(L+1).EC.C) GC TC 813
      CALL ACVAN(NCMCDE)
      L=L+1
      GC TO 2CC
   21 L=L+1
      GC TC 17
CLOAD DECAY INFCRMATICN INTC THE HCLDING ARRAYS
  800 CC 801 IL=1,NDMCDE
      READ(7,30C) IDEKM(IL),IDAUG(IL),DLEVL(IL),PROB(IL)
  801 CONTINUE
CMARK THE BEGINNING CF LIKE PARENT NUCLIDES IN THE LIST
      IBEGIN=L
  8C2 L=L+1
      IF(NP(L).EC.NP(L+1)) GO TO 802
CMARK THE END OF LIKE PARENT NUCLIDES IN THE LIST
      IENC=L
  81C DC 805 IP=IBEGIN,IEND
CDETERMINE IF A PAREAT NUCLIDE WILL BE CONSIDERED FCR DOSE RATE CALCS
      IF(H(IP)/HALFLF.GT.C.8.AND.H(IP)/HALFLF.LT.1.2.AND.HALFLF.NE.1.0E+
     *50.AND.AC(IP).NE.0.0) GC TO 804
      GC TO 805
  804 CC 820 IL=1,NCMCDE
CBEGIN INTERRCGATICN CF THE HCLDING ARRAYS
      IF(IDEKM(IL).EQ.9) NDT(IP)=IDAUG(IL)
      IF(IDEKM(IL).EC.7.OR.IDEKM(IL).EQ.9) GO TO 501
      IF(IDEKM(IL).EC.8) GC TC 510
      GO TO 820
  5C1 IF(DLEVL(IL).EC.0.0) GO TC 820
      CALL CURVE(DLEVL(IL),FLUXE)
      CALL CALC(AC(IP),DCSPRO(IP),PROB(IL),FLUXE)
      GC TC 820
  51C IF(DLEVL(IL).EC.0.C) GC TC 511
      CALL CURVE(DLEVL(IL),FLUXE)
      CALL CALC(AC(IP),DCSPRO(IP),PROB(IL),FLUXE)
  511 DO 821 IBETA=1,2
      CALL CURVE(0.511,FLUXE)
      CALL CALC(AC(IP),CCSPRO(IP),PROB(IL),FLUXE)
  821 CONTINUE
  820 CCNTINUE
```

```
      **SUBROUTINE DOSRTE(TIMACT,PRODNU,NONUCL,NOPAS,DOSSUM,SYSDOS,IDNO,ST**
  805 CONTINUE
CDETERMINE IF THE NEXT LEVDEC NUCLIDE IS THE SAME AS THE PREVIOUS ONE
  806 READ(7,16) ITAPE,HALFLF,NDMODE
      IF(EOF(7).NE.0) GO TO 808
      IF(ITAPE.EQ.NP(IBEGIN)) GO TO 808
      GO TO 807
  808 DO 809 IL=1,NDMODE
      READ(7,300) IDEKM(IL),IDAUG(IL),OLEVL(IL),PROB(IL)
  809 CONTINUE
      GO TO 810
CDETERMINE IF THE NEXT LEVDEC NUCLIDE IS EQUAL TO THE NEXT PARENT ID
CNUMBER FOLLOWING THE LIST OF LIKE PARENT NUCLIDES
  807 IF(ITAPE.EQ.NP(IEND+1)) GO TO 811
      CALL ADVAN(NDMODE)
      L=IEND+1
      IF(ITAPE.GT.NP(IEND+1)) GO TO 812
      GO TO 200
  811 L=IEND+1
      GO TO 17
  812 WRITE(6,29) NP(L)
      WRITE(6,37) NT(L)
      L=L+1
      IF(ITAPE.GT.NP(L)) GO TO 812
      GO TO 200
  813 REWIND 7
CMOVE THE FILE MARKER THROUGH THE ACTINIDE LIBRARY AND POSITION AT THE
CBEGINNING OF LEVDEC
   49 READ(7,47) ITAPE
   47 FORMAT(I6)
      IF(EOF(7).NE.0) GO TO 700
      CALL POSITON
      GO TO 49
CMARK THE END OF THE LIST OF ARRANGED PARENT NUCLIDES
  700 ISTOP=L
      L=IBASE
CBEGIN INTERROGATION OF THE LEVDEC LIBRARY FOR POSSIBLE RADIOACTIVE
CDAUGHTERS-READ UNTIL THE ID NUMBER IS MATCHED
  701 IF(ND(L).NE.0) GO TO 702
      L=L+1
      IF(L.EQ.(ISTOP+1) GO TO 913
      GO TO 701
  702 READ(7,16) ITAPE,HALFLF,NDMODE
      IF(EOF(7).NE.0) GO TO 702
      IF(ITAPE.EQ.ND(L)) GO TO 703
      IF(ITAPE.GT.ND(L)) GO TO 926
      CALL ADVAN(NDMODE)
      GO TO 702
  926 L=L+1
      IF(L.EQ.(ISTOP+1) GO TO 913
      IF(ITAPE.EQ.ND(L)) GO TO 703
      CALL ADVAN(NDMODE)
      GO TO 701
CCHECK THE HALF LIFE OF THE DAUGHTER
  703 IF(HALFLF.GT.1.0E+48) GO TO 704
      GO TO 705
  704 CALL ADVAN(NDMODE)
  904 L=L+1
      IF(L.EQ.(ISTOP+1) GO TO 913
```

```
      **SUBROUTINE DCSRTE(TIMACT,PRCDNU,NCNUCL,NCPAS,DOSSUM,SYSDOS,IDNO,ST**
      GO TO 701
  705 IMODE=0
  902 IMODE=IMODE+1
CBEGIN INTERFCGATION OF THE LEVDEC LIBRARY FOR DAUGHTER DECAY INFO
      READ(7,707) IDEKMC,DLEVEL,MULT
  707 FORMAT(38X,I2,7X,E11.4,E12.4)
      IF(IDEKMC.EQ.7.OR.IDEKMO.EQ.9) GO TO 901
      IF(IDEKMC.EQ.8) GO TO 910
  902 IF(IMODE.EQ.NDMODE) GO TO 904
      GO TO 902
  901 IF(DLEVEL.EQ.0.0) GO TO 903
      CALL CURVE(DLEVEL,FLUXE)
      CALL CALC(AC(L),DCSDAU(L),MULT,FLUXE)
      GO TO 903
  910 IF(DLEVEL.EQ.0.0) GO TO 911
      CALL CURVE(DLEVEL,FLUXE)
      CALL CALC(AC(L),DOSDAL(L),MULT,FLUXE)
C1K ANY CASE ADD IN TWO 0.511 MEV ANNIHILIATION GAMMAS
  911 DO 912 IBETA=1,2
      CALL CURVE(0.511,FLUXE)
      CALL CALC(AC(L),DCSDAU(L),MULT,FLUXE)
  912 CONTINUE
      GO TO 903
  913 REWIND 7
      NC=NCNUCL*13
      DO 815 I=1,NONUCL
CPRINT HEADINGS
      WRITE(6,8)(STAFS(K),K=1,NCPAS)
    8 FORMAT(/,1H ,4X,7HTIME(H),4X,10(E8.3,1X),2X,2(E8.3,2X))
      WRITE(6,400)
  400 FORMAT(1H ,125HTARGET PRODUCT  DSR(T1)  DSR(T2)  DSR(T3)  DSR(T4)
     + DSR(T5)  DSR(T6)  DSR(T7)  DSR(T8)  DSR(T9)  DSR(T10)  DSR(T11)
     +DSR(T12))
      DO 818 L=IBASE,NC
      IF(NT(L).EQ.IDNO(I)) GO TO 817
      GO TO 818
CSUM THE DCSE RATES DUE TO PARENTS AND DAUGHTERS
  817 DO 819 K=1,NCPAS
      DCSNUC(K)=DCSPRO(L)*EXP(-(STAFS(K)*0.693*3600.0/H(L)))+DOSDAU(L)*E
     +XP(-(STAFS(K)*0.693*3600.0/H(L)))
      DCSSUM(K)=DCSSUM(K)+DCSNUC(K)
      SYSDCS(K)=SYSDCS(K)+DCSNUC(K)
  819 CONTINUE
CPRINT DCSE RATES
      WRITE(6,20) IDNO(I),NP(L),(DOSNUC(K),K=1,NCPAS)
   20 FORMAT(1X,1X,I5,2X,I5,2X,10(E8.3,1X),2X,2(E8.3,2X))
  818 CONTINUE
CPRINT HEADINGS
      WRITE(6,8)(STAFS(K),K=1,NCPAS)
      WRITE(6,10)
   10 FORMAT(1H ,125H    TARGET        DSR(T1)  DSR(T2)  DSR(T3)  DSR(T4)
     + DSR(T5)  DSR(T6)  DSR(T7)  DSR(T8)  DSR(T9)  DSR(T10)  DSR(T11)
     +DSR(T12))
      WRITE(6,23) IDNO(I),(DCSSUM(K),K=1,NCPAS)
```

```
      **SUBROUTINE CCSRTE(TIMACT,PRODNU,NONUCL,NOPAS,DOSSUM,SYSDOS,IDNO,ST**
   23 FORMAT(1X,5X,I5,5X,10(E8.3,1X),2X,2(E8.3,2X))
      DO 822 K=1,NOPAS
      CCSSUM(K)=0.0
  822 CONTINUE
  815 CONTINUE
      WRITE(6,81)(STAFS(K),K=1,NOPAS)
CPRINT SYSTEM HEADINGS
      WRITE(6,26)
   26 FORMAT(1H ,125H     SYSTEM        DSR(T1)   DSR(T2)   DSR(T3)   DSR(T4)
     + DSR(T5)   DSR(T6)   DSR(T7)   DSR(T8)   DSR(T9)   DSR(T10)   DSR(T11)
     +DSR(T12))
CPRINT SYSTEM CCSE
      WRITE(6,51)(SYSDOS(K),K=1,NOPAS)
   51 FORMAT(1X,15X,10(E8.3,1X),2X,2(E8.3,2X))
      RETURN
      END
```

```
      **SUBROUTINE CALC(ACT,DSPRD,MULT,FLUXE)**

      SUBROUTINE CALC(ACT,DSPRD,MULT,FLUXE)
C
C
CTHIS PROGRAM CALCULATES DOSERATE FOR 3 DIFFERENT GEOMETRIES
C
C
C     ARGUMENTS IN-ACT,MULT,FLUXE
C     ARGUMENTS OUT-DSPRD
C
C
      COMMON A,V,F,DF,T1,T2,N,R,D,HT,OPTION,IGEOM
      REAL MULT
      GO TO(31,32,33) IGEOM
   31 DOSRAT=(ACT*MULT*2.5/(4.0*3.1416*FLUXE*(D*100.0)**2))          1
      DSPRD=DSPRD+DOSRAT
      RETURN
CTHE ABOVE DOSERATE CALCULATION IS FOR A POINT SOURCE
   32 VOL=4*3.1416*((R*100.0)**3)/3
      S=(ACT*MULT*2.5)/(VOL*FLUXE*4.0)
      DOSRAT=S*(2*R*(R+D)*10000.0-D*(2*R+D)*10000.0*(ALOG((2*R+D)/D)))/(
     +(R+D)*100.0)
      DSPRD=DSPRD+DOSRAT
      RETURN
CTHE ABOVE DOSERATE CALCULATION IS FOR A HOMOGENEOUS NON-ABSORBING
CSPHERE
   33 VOL=3.1416*((R*100.0)**2)*HT*100.0
      S=(ACT*MULT*2.5)/(VOL*FLUXE*4.0)
      DOSRAT=S*((HT+D)*100.0*(ALOG((1+(R**2)/((HT+D)**2))+2*R*ATAN((HT+D)
     +/R)/(HT+D))-D*100.0*(ALOG(1+(R**2)/(D**2))+2*R*ATAN(D/R)/D))
      DSPRD=DSPRD+DOSRAT
      RETURN
CTHE ABOVE CALCULATION IS FOR A HOMOGENEOUS NON-ABSORBING CYLINDER WHERE
CTHE OBSERVER FACES THE AXIS
      END
```

```
      **SUBROUTINE CURVE(E,FLUXE)**

      SUBROUTINE CURVE(E,FLUXE)
C
C
CGIVEN THE ENERGY OF THE GAMMA RAY,THIS PROGRAM CALCULATES THE PARTICLE
CFLUX(QUANTA/CM**2*SEC) THAT CORRESPONDS TO A DOSE RATE OF 2.5 MR/HR --
CFLUXE
C
C
C     ARGUMENTS IN-DLEVEL
C     ARGUMENTS OUT-FLUXE
C
C
      IF(E.GE.0.010.AND.E.LE.0.020) GO TO 1
      IF(E.GT.0.020.AND.E.LE.0.040) GO TO 2
      IF(E.GT.0.040.AND.E.LE.0.050) GO TO 3
      IF(E.GT.0.050.AND.E.LE.0.060) GO TO 4
      IF(E.GT.0.060.AND.E.LE.0.080) GO TO 5
      IF(E.GT.0.080.AND.E.LE.0.150) GO TO 6
      IF(E.GT.0.150.AND.E.LE.0.400) GO TO 7
      IF(E.GT.0.400.AND.E.LE.0.600) GO TO 8
      IF(E.GT.0.600.AND.E.LE.1.000) GO TO 9
      IF(E.GT.1.000.AND.E.LE.1.500) GO TO 10
      IF(E.GT.1.500.AND.E.LE.3.000) GO TO 11
      IF(E.GT.3.000.AND.E.LE.6.000) GO TO 12
      IF(E.GT.6.000.AND.E.LE.10.000) GO TO 13
      FLUXE=1.0E+99
      RETURN
    1 FLUXE=21495330.64*E**2.1759447
      RETURN
    2 FLUXE=11460233.22*E**2.0136022
      RETURN
    3 FLUXE=1036515.575*E**1.2698653
      RETURN
    4 FLUXE=96497.38032*E**0.4772399
      RETURN
    5 FLUXE=11225.12221*E**(-0.2874410)
      RETURN
    6 FLUXE=554E2.78326*EXP(E*(-10.91320651))
      RETURN
    7 FLUXE=1330.569205*E**(-1.1036085)
      RETURN
    8 FLUXE=1462.198566*E**(-1.0004513)
      RETURN
    9 FLUXE=1546.557758*E**(-0.8891058)
      RETURN
   10 FLUXE=1545.05132C*E**(-0.7797676)
      RETURN
   11 FLUXE=1508.205757*E**(-0.7188784)
      RETURN
   12 FLUXE=1488.362150*E**(-0.7051622)
      RETURN
   13 FLUXE=1656.852E74*E**(-0.7649757)
      RETURN
      END
```

DETERMINATION OF THE DOSE RATE FROM NEUTRON ACTIVATION OF THE COPPER BEAM DUMP

THE FOLLOWING DATA HAS BEEN ENTERED INTO MEMORY

AMPERES	KILO-VOLTS	BEAMREACTION	DUTYFACTOR	T1	T2	N
.650E+02	.170E+03	.100E+01	.100E+00	.800E+01	.160E+02	7

NUMBER OF ENERGY GROUPS-NCEGPS 20

NUMBER OF TARGET NUCLIDES-NONUCL 2

NUMBER OF POINTS AFTER SHUTDOWN-NOPAS 12

RADIUS R OF SPHERE OR CYLINDER (M) .215E+00

HEIGHT HT OF CYLINDER (M) 0.

DISTANCE D FROM POINT SOURCE OR SURFACE OF SPHERE OR CYLINDER (M) .200E+01

NUMBER OF GROUPS BOUNDARIES-NG 21

NUMBER OF FAST NEUTRON GROUPS-NF 2

SPECIFIC TIMES AFTER SHUTDOWN (H)

TIME 1	0.
TIME 2	.1CCE+C1
TIME 3	.2CCE+C1
TIME 4	.300E+01
TIME 5	.4CCE+01
TIME 6	.5C0E+C1
TIME 7	.600E+01
TIME 8	.7CCE+C1
TIME 9	.8C0E+01
TIME 10	.900E+01
TIME 11	.1CCE+C2
TIME 12	.200E+02

UNIT NORMALIZED FLUXES

GROUP 1	.7591E-C5
GROUP 2	.1382E-C6
GROUP 3	.8333E-C7
GROUP 4	.1058E-06
GROUP 5	.1378E-C6
GROUP 6	.2842E-C7
GROUP 7	.4790E-07
GROUP 8	.3860E-C7
GROUP 9	.7420E-C8
GROUP 10	.3522E-07
GROUP 11	.457CE-C7
GROUP 12	.4021E-C7
GROUP 13	.3043E-C7
GROUP 14	.7234E-C7
GROUP 15	.4684E-C7
GROUP 16	.3C11E-C7
GROUP 17	.3551E-C7
GROUP 18	.3456E-07
GROUP 19	.3C76E-C7
GROUP 20	.8288E-C6

GROUP BOUNDARIES (MEV)		
BOUNDARY	1	.2385E+01
BOUNDARY	2	.2307E+01
BOUNDARY	3	.1827E+01
BOUNDARY	4	.1108E+01
BOUNDARY	5	.5502E+00
BOUNDARY	6	.1576E+00
BOUNDARY	7	.1111E+00
BOUNDARY	8	.5248E-01
BOUNDARY	9	.2479E-01
BOUNDARY	10	.2198E-01
BOUNDARY	11	.1033E-01
BOUNDARY	12	.3355E-02
BOUNDARY	13	.1234E-02
BOUNDARY	14	.5829E-03
BOUNDARY	15	.1013E-03
BOUNDARY	16	.2902E-04
BOUNDARY	17	.1068E-04
BOUNDARY	18	.3059E-05
BOUNDARY	19	.1125E-05
BOUNDARY	20	.4140E-06
BOUNDARY	21	.1000E-10

NAME	ID-NUMBER	MASS(KG)
COPPER63	29063	.258E+03
COPPER65	29065	.115E+03

THE FOLLOWING HAS BEEN CALCULATED BY ACDOS1

INSTANTANEOUS NUMBER OF NEUTRONS PRODUCED PER SECOND .7645E+13

AVERAGE NUMBER OF NEUTRONS PRODUCED PER SECOND .7645E+12

INTEGRALS OF THE FITTING FUNCTION OVER THE GROUP INTERVALS

GROUP	LOWER BOUNDARY	UPPER BOUNDARY	REQUIRED INTEGRAL	CALCULATED INTEGRAL
20	.100E-04	.414E+00	.829E-06	.829E-06
19	.414E+00	.112E+01	.308E-07	.308E-07
18	.113E+01	.306E+01	.346E-07	.346E-07
17	.306E+01	.107E+02	.395E-07	.395E-07
16	.107E+02	.290E+02	.301E-07	.301E-07
15	.290E+02	.101E+03	.468E-07	.468E-07

14	.101E+03	.583E+03	.723E-07	.723E-07
13	.583E+03	.123E+04	.304E-07	.304E-07
12	.123E+04	.336E+04	.402E-07	.402E-07
11	.336E+04	.103E+05	.457E-07	.457E-07
10	.103E+05	.219E+05	.352E-07	.352E-07
9	.219E+05	.248E+05	.742E-08	.742E-08
8	.248E+05	.525E+05	.386E-07	.386E-07
7	.525E+05	.111E+06	.479E-07	.479E-07
6	.111E+06	.158E+06	.284E-07	.284E-07
5	.158E+06	.550E+06	.138E-06	.138E-06
4	.550E+06	.111E+07	.106E-06	.106E-06
3	.111E+07	.183E+07	.833E-07	.833E-07
2	.183E+07	.231E+07	.138E-06	.138E-06
1	.231E+07	.239E+07	.790E-05	.790E-05

THE FOLLOWING OUTPUT DATA ARE CALCULATED ACTIVITIES IN BQ

TIME(H)		.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
TARGET	PRODUCT	ACT(T1)	ACT(T2)	ACT(T3)	ACT(T4)	ACT(T5)	ACT(T6)	ACT(T7)	ACT(T8)
29063	29062	.0	.0	.0	.0	.0	.0	.0	.0
29063	28062	.0	.0	.0	.0	.0	.0	.0	.0
29063	27059	.0	.0	.0	.0	.0	.0	.0	.0
29063	28063	.176E+04	.176E+04	.176E+04	.176E+04	.176E+04	.176E+04	.176E+04	.176E+04
29063	27060	.0	.0	.0	.0	.0	.0	.0	.0
29063	27060	.0	.0	.0	.0	.0	.0	.0	.0
29063	29064	.431E+10	.409E+10	.387E+10	.367E+10	.347E+10	.329E+10	.312E+10	.295E+10

TIME(H)	.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
TARGET	ACT(T1)	ACT(T2)	ACT(T3)	ACT(T4)	ACT(T5)	ACT(T6)	ACT(T7)	ACT(T8)
29063	.431E+10	.409E+10	.387E+10	.367E+10	.347E+10	.329E+10	.312E+10	.295E+10

.800E+01	.900E+01	.100E+02	.200E+02
ACT(T9)	ACT(T10)	ACT(T11)	ACT(T12)
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.176E+04	.176E+04	.176E+04	.176E+04
.0	.0	.0	.0
.0	.0	.0	.0
.280E+10	.265E+10	.251E+10	.146E+10

.800E+01	.900E+01	.100E+02	.200E+02
ACT(T9)	ACT(T10)	ACT(T11)	ACT(T12)
.280E+10	.265E+10	.251E+10	.146E+10

TIME(H)		.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
TARGET	PRODUCT	ACT(T1)	ACT(T2)	ACT(T3)	ACT(T4)	ACT(T5)	ACT(T6)	ACT(T7)	ACT(T8)
29065	29064	.0	.0	.0	.0	.0	.0	.0	.0
29065	29063	.0	.0	.0	.0	.0	.0	.0	.0
29065	28064	.0	.0	.0	.0	.0	.0	.0	.0
29065	27061	.0	.0	.0	.0	.0	.0	.0	.0
29065	28065	.0	.0	.0	.0	.0	.0	.0	.0
29065	27062	.0	.0	.0	.0	.0	.0	.0	.0
29065	27062	.0	.0	.0	.0	.0	.0	.0	.0
29065	25066	.148E+10	.426E+06	.123E+03	.353E-01	.102E-04	.293E-08	.842E-12	.242E-15

TIME(H)	.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
TARGET	ACT(T1)	ACT(T2)	ACT(T3)	ACT(T4)	ACT(T5)	ACT(T6)	ACT(T7)	ACT(T8)
29065	.148E+10	.426E+06	.123E+03	.353E-01	.102E-04	.293E-08	.842E-12	.242E-15

TIME(H)	.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
SYSTEM	ACT(T1)	ACT(T2)	ACT(T3)	ACT(T4)	ACT(T5)	ACT(T6)	ACT(T7)	ACT(T8)
	.579E+10	.409E+10	.387E+10	.367E+10	.347E+10	.329E+10	.312E+10	.295E+10

.800E+01	.900E+01	.100E+02	.200E+02
ACT(T9)	ACT(T10)	ACT(T11)	ACT(T12)
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.698E-19	.201E-22	.578E-26	.226E-61

.800E+01	.900E+01	.100E+02	.200E+02
ACT(T9)	ACT(T10)	ACT(T11)	ACT(T12)
.698E-19	.201E-22	.578E-26	.226E-61

.800E+01	.900E+01	.100E+02	.200E+02
ACT(T9)	ACT(T10)	ACT(T11)	ACT(T12)
.280E+10	.265E+10	.251E+10	.146E+10

THE FOLLOWING OUTPUT DATA ARE CALCULATED DOSE RATES IN MRAD/H

TIME(H)		.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
TARGET	PRODUCT	DSR(T1)	DSR(T2)	DSR(T3)	DSR(T4)	DSR(T5)	DSR(T6)	DSR(T7)	DSR(T8)
29063	27059	.0	.0	.0	.0	.0	.0	.0	.0
29063	27060	.0	.0	.0	.0	.0	.0	.0	.0
29063	27060	.0	.0	.0	.0	.0	.0	.0	.0
29063	28062	.0	.0	.0	.0	.0	.0	.0	.0
29063	28063	.0	.0	.0	.0	.0	.0	.0	.0
29063	29062	.0	.0	.0	.0	.0	.0	.0	.0
29063	29064	.234E+01	.221E+01	.210E+01	.199E+01	.188E+01	.178E+01	.169E+01	.160E+01

TIME(H)	.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
TARGET	DSR(T1)	DSR(T2)	DSR(T3)	DSR(T4)	DSR(T5)	DSR(T6)	DSR(T7)	DSR(T8)
29063	.234E+01	.221E+01	.210E+01	.199E+01	.188E+01	.178E+01	.169E+01	.160E+01

.800E+01	.900E+01	.100E+02	.200E+02
DSR(T9)	DSR(T10)	DSR(T11)	DSR(T12)
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.151E+01	.143E+01	.136E+01	.791E+00

.800E+01	.900E+01	.100E+02	.200E+02
DSR(T9)	DSR(T10)	DSR(T11)	DSR(T12)
.151E+01	.143E+01	.136E+01	.791E+00

TIME(H)		.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
TARGET	PRODUCT	DSR(T1)	DSR(T2)	DSR(T3)	DSR(T4)	DSR(T5)	DSR(T6)	DSR(T7)	DSR(T8)
29065	27061	.0	.0	.0	.0	.0	.0	.0	.0
29065	27062	.0	.0	.0	.0	.0	.0	.0	.0
29065	27062	.0	.0	.0	.0	.0	.0	.0	.0
29065	28064	.0	.0	.0	.0	.0	.0	.0	.0
29065	28065	.0	.0	.0	.0	.0	.0	.0	.0
29065	25063	.0	.0	.0	.0	.0	.0	.0	.0
29065	29064	.0	.0	.0	.0	.0	.0	.0	.0
29065	29066	.332E+00	.556E-04	.275E-07	.792E-11	.228E-14	.656E-18	.189E-21	.544E-25

TIME(H)	.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
TARGET	DSR(T1)	DSR(T2)	DSR(T3)	DSR(T4)	DSR(T5)	DSR(T6)	DSR(T7)	DSR(T8)
29065	.332E+00	.556E-04	.275E-07	.792E-11	.228E-14	.656E-18	.189E-21	.544E-25

TIME(H)	.0	.100E+01	.200E+01	.300E+01	.400E+01	.500E+01	.600E+01	.700E+01
SYSTEM	DSR(T1)	DSR(T2)	DSR(T3)	DSR(T4)	DSR(T5)	DSR(T6)	DSR(T7)	DSR(T8)
	.267E+01	.221E+01	.210E+01	.199E+01	.188E+01	.178E+01	.165E+01	.160E+01

.800E+01	.900E+01	.100E+02	.200E+02
DSR(T9)	DSR(T10)	DSR(T11)	DSR(T12)
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.0	.0	.0	.0
.157E-28	.451E-32	.130E-35	.508E-71

.800E+01	.900E+01	.100E+02	.200E+02
DSR(T9)	DSR(T10)	DSR(T11)	DSR(T12)
.157E-28	.451E-32	.130E-35	.508E-71

.800E+01	.900E+01	.100E+02	.200E+02
DSR(T9)	DSR(T10)	DSR(T11)	DSR(T12)
.151E+01	.143E+01	.136E+01	.791E+00

References

1. M. S. Singh, "D+D Reaction Neutron Yields from the Implantation of Incident D^+ in Metals", Lawrence Livermore National Laboratory Report UCRL-77067, July 31, 1975.

2. L. Ruby, private communication dated September 14, 1980.

3. J. McCaslin, "Calculated Energy Distribution of Neutrons Inside the TFTR (Testcell) Shield - Application to Activation Analysis, Memo to L. Ruby dated March 14, 1980.

4. M. A. Gardner and R. J. Howerton, "ACTL: Evaluated Neutron Activation Library", Lawrence Livermore National Laboratory Report UCRL-50400, Vol. 18, October 17, 1978.

5. R. J. Howerton, "Flux Weighted Group Cross Sections Based on the ACTLMFE Data File", Lawrence Livermore National Laboratory Report UCID-18749, July 1980.

6. R. J. Howerton, "ENSL and CDRL: Evaluated Nuclear Structure Library", Lawrence Livermore National Report UCRL-50400, Vol. 23, February 2, 1981.

7. R. G. Jaeger, E. D. Blizard, A. B. Chilton, M. Grotenhuis, A. Honig, TH. A. Jaeger, H. H. Eisenlogr, Engineering Compendium On Radiation Shielding, Vol. 1, Springer-Verlag, New York Inc. 1968

8. L. Ruby, "Radioactivation of a TFTR-Type Beam Dump During Testing", Memo to R. V. Pyle dated July 3, 1979.

9. B. T. Price, C. C. Horton, "Radiation Shielding", International Series of Monographs on Nuclear Energy, Pergamon Press, 1957.

10. R. G. Alsmiller, Jr., J. Barish, R. T. Santoro, R. A. Lillie, J. M. Barnes, M. M. H. Ragheb, "Dose Rates From Induced Activity in the Tokamak Fusion Test Reactor Test Cell", Nucl. Technol. 48, 187 (1980).