LA-NUREG-6685-MS

Informal Report

NRC-8

C.3

Methods for Calculating Group Cross Sections for Doubly Heterogeneous Thermal Reactor Systems

Issued: February 1977

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87545

An Affirmative Action/Equal Opportunity Employer

UNITED STATES
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

This work supported by the US Nuclear Regulatory Commission,
Division of Reactor Safety Research.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Price: Printed Copy $4.50 Microfiche $3.00

LA-NUREG-6685-MS
Informal Report
NRC-8

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87545

Methods for Calculating Group Cross Sections for Doubly Heterogeneous Thermal Reactor Systems

by

M. G. Stamatelatos
R. J. LaBauve

Manuscript completed: January 1977
Issued: February 1977

Prepared for the US Nuclear Regulatory Commission
Office of Nuclear Regulatory Research

METHODS FOR CALCULATING GROUP CROSS SECTIONS

FOR DOUBLY HETEROGENEOUS THERMAL REACTOR SYSTEMS

by

M. G. Stamatelatos and R. J. LaBauve

ABSTRACT

This report discusses methods used at LASL for calculating group cross sections for doubly heterogeneous HTGR systems of the General Atomic design. These cross sections have been used for the neutronic safety analysis calculations of such HTGR systems at various points in reactor lifetime (e.g., beginning-of-life, end-of-equilibrium cycle). They were also compared with supplied General Atomic cross sections generated with General Atomic codes. The overall agreement between the LASL and the GA cross sections has been satisfactory.

I. INTRODUCTION

Over approximately the past two and one-half years, the Los Alamos Scientific Laboratory has been engaged in reactor safety studies for High Temperature Gas-cooled Reactor (HTGR) systems of the General Atomic design. Discussed in this report is the methodology connected with a small part of this effort, namely the calculation of multigroup cross sections for use in neutronic calculations (e.g., effective multiplication factors, temperature coefficients, etc.). The initial effort has been directed towards using generally available computer codes with minimal effort in the direction of new methods development. Unfortunately, however, many specialized GA codes were kept proprietary and other widely available codes were not specialized enough to correctly treat special configurations like, for example, doubly heterogeneous HTGR systems. Therefore, at some point in the cross-section development, it was decided to intensify the development of methods to treat such system peculiarities. Therefore, as it will be seen in the following discussion, the final code system configuration used resembles little the initial configuration used for calculating homogeneous HTGR cross sections.

II. HOMOGENEOUS CROSS SECTIONS

In the initial stages of the cross-section generation process, a number of code systems were explored and these are discussed here mostly for the sake of "historic" completeness. Although these systems are quite different from the final system used, they are nevertheless valid options for generating homogenized-medium cross sections or cross sections for media with one allowed level of heterogeneity. Approximate ways of incorporating the effects of the second level of heterogeneity (fuel grains in a fuel rod) have been explored, as will be seen later, but the final system chosen has proved to be superior to the others in all respects including accuracy and flexibility.

The initial data flow system (including options) for generating homogeneous-medium few-group cross sections is shown in Fig. 1. The starting point has always been the basic Evaluated Nuclear Data Files (ENDF/B) cross sections (initially version III; later several version IV elements were included). The few-group neutron energy structure used in all the work described in this report has been a nine-group General Atomic structure (adopted for comparison with supplied GA cross sections) shown in Table I. The initial set of tempera-

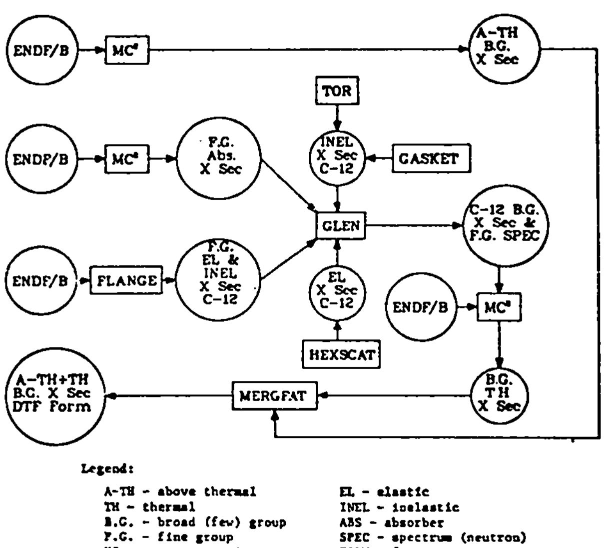


Fig. 1.
Initial data flow systems (several options are shown).

TABLE I

FEW-GROUP ENERGY STRUCTURE

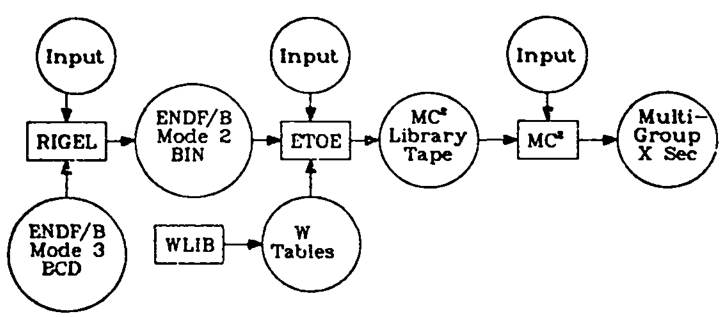
E_{max} = 10 MeV

Group No.	Lower Boundary (eV)
1	$1.83 \times 10^{+5}$
2	$9.61 \times 10^{+2}$
3	$1.76 \times 10^{+1}$
4	3.93
5	2.38
6	4.14×10^{-1}
7	1.00×10^{-1}
8	4.00×10^{-2}
9	5.00×10^{-4}

tures for which few-group cross sections were generated is: 300, 500, 800, 1200, 1700, 2300, and 3000 K. These were used for the beginning-of-life (BOL) composition. Later, several other temepratures (600, 1000, 1500, 2000, and 2600 K) were also included for a more accurate evaluation of the temperature coefficient at the end-of-equilibrium-cycle (EOEC) composition.

The above-thermal (10 MeV - 2.38 eV) cross sections of the system shown in Fig. 1 were generated with an operational LASL-modified version of MC^2-I[1] code that requires special library preparation, i.e., it does not directly operate on the ENDF/B cross-section files. The preparation of such an MC^2 input file is shown in the diagram of Fig. 2. The RIGEL[2] code is used to convert ENDF/B data in standard BCD format (Mode 3) to an alternate binary format (Mode 2). The ETOE[3] code prepares a library tape for MC^2 including "W-tables" that are supplied by the WLIB code. Since ETOE provides pointwise elastic-scattering cross sections for MC^2, temperature must be an input parameter to ETOE which means that a different MC^2 library tape must be prepared for each temperature. The various MC^2 libraries are then merged with an auxiliary code, MERMC2, not shown in Fig. 1. There are certain limitations connected with the MC^2 code, some of which have proved to be so hard to circumvent, unless considerable effort was put in modifying the code, that MC^2-I was removed from the final data flow system to be discussed later. First, because of storage limitations, fine-group cross sections for the entire energy range (10 MeV - 10^{-5} eV) cannot be generated in one pass, so that separate but slightly overlapping problems were run for the "high" (10 MeV - 0.414 eV) and "low" (2.38 - 5×10^{-4} eV) energy ranges. Second, the maximum energy value in MC^2-I is fixed (10 MeV) and one is also forced to use a fixed-lethargy grid in one of two available options, "all-fine" with Δu = 0.25 and "ultra-fine" with $\Delta u = 1/120$. Since the second option was found to be too time-consuming and costly without the benefit of considerable increase in

Fig. 2.
MC^2-I library preparation.



output cross-section quality, the "all-fine" option was chosen for generating both above-thermal and thermal fine-group cross sections in the GAM-I constant-lethargy structure of 0.25. The spectrum-weighting function specified for the derivation of fine-group cross sections was chosen to be 1/E for the above-thermal region and a "properly hardened Maxwellian" for the thermal region. The latter was calculated by the thermal code GLEN.[4]

The graphite cross sections in the thermal region were treated separately. Initially, the FLANGE[5] code was used to interpolate (both energy-wise and temperature-wise) preprocessed graphite thermal inelastic-scattering cross sections available in ENDF/B format (MAT 1065, MF 4 and 7). This process has proved costly and inefficient by comparison with directly calculating the $S(\alpha,\beta)$ data from codes like GASKET[6] or TOR.[7] The graphite coherent elastic cross section was calculated with a modified version of the HEXSCAT[8] code which now calculates Legendre elastic-scattering components up to the order 5. All fine-group thermal cross sections were collapsed with the GLEN code to the required few-group set. MC^2 was used to collapse the above-thermal fine-group cross sections to the corresponding few-group set. An auxiliary code MERGFAT (Appendix C) was used to merge the fast and thermal few-group cross sections in the proper format required by the DTF-IV[9] neutronics transport code.

Several modifications to MC^2-I were made. An important one was in the multigroup averaging method for the resolved-resonance capture cross sections. The MC^2-I method is given by the following equations:

$$\left(\bar{\sigma}^{bg}_{c_J}\right)_{resolved} = \frac{\sum\limits_{j \text{ in } J} \langle\sigma\rangle^{fg}_{j\ resolved}\, Q_j}{\sum\limits_{j \text{ in } J} Q_j} , \qquad (1)$$

where

$$Q_j = \int_{E_j}^{E_{j+1}} S^{fg}_j \frac{dE}{\Sigma_T(E)} , \qquad (2)$$

and

$$S_j^{fg} = \sum_{k\neq j} \Sigma_{inel,k}^{fg} \phi_k^{fg} \frac{<\Sigma_{inel}>_{k\to j}}{<\Sigma_{inel}>_k} + \sum_{k\neq j} \Sigma_{n,2n,k}^{fg} \phi_k^{fg} \frac{2<\Sigma_{n,2n}>_{k\to j}}{<\Sigma_{n,2n}>_k}$$

$$+ <\Sigma_{el}>_{j-1\to j} \phi_{j-1}^{fg} \quad , \tag{3}$$

where superscripts fg and bg indicate fine-group and broad-group, respectively. J and j are subscripts referring to broad-group and fine-group, respectively. This method of averaging has produced unsatisfactory results and, since it had not been shown to be valid for thermal reactor systems, it was replaced by the usual spectrum-weighting method used by almost all multigrouping codes. This change has resulted in much better MC^2-I cross sections.

Regarding other codes used, one of the most important changes was made in GLEN whose original version did not allow for energy-dependent scattering cross sections of nonmoderator materials. Although for most heavy absorbers it is possible to give the thermal-scattering cross sections in terms of an average energy-independent number, some resonance elements like ^{135}Xe or ^{149}Sm definitely require energy-dependent scattering cross sections. Therefore, modifications were made in the GLEN code to allow the option of including energy-dependent scattering cross sections together with the only previously available option of supplying a single energy-independent scattering cross section value for each nonmoderator material. The choice of options in the modified GLEN version is made by means of a flag, ISCAT.

The homogeneous cross sections produced by the scheme of Fig. 1, including all discussed modifications, for a beginning-of-life HTGR composition were found good* and the discrepancies between these cross sections and the supplied General Atomic (GA) cross sections produced with the proprietary MICROX[10] code were in the direction attributable to heterogeneity effects or to different initial basic data. The incorporation of double-heterogeneity effects by the MICROX method (for comparison with the GA cross sections) was found to essentially amount to reprogramming the MC^2-I code. Although alternate approximate methods of incorporating double heterogeneity effects in codes like MC^2-I were developed, as

*Comparisons were made with the MC^2-II code, courtesy of H. Henryson of ANL.

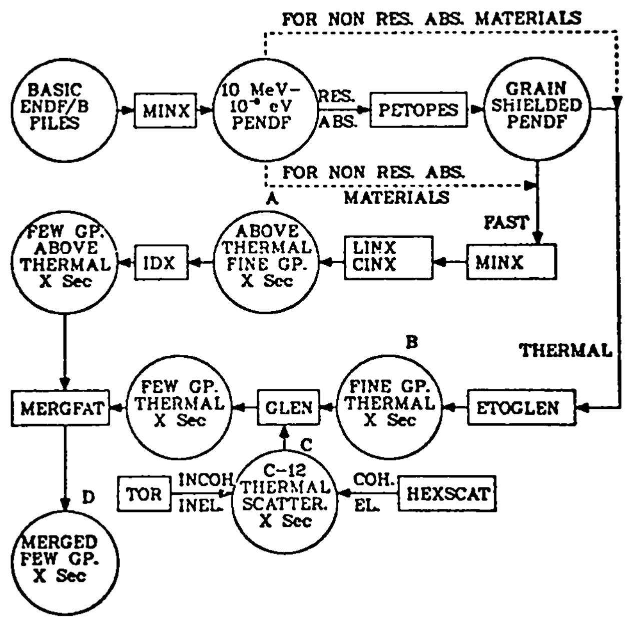


Fig. 3.
Final data flow system.

it will be discussed later, we have decided to adopt a totally new data flow system (Fig. 3), more modern and more flexible including the MINX[11] code which was developed at LASL.

III. CROSS SECTIONS FOR DOUBLY HETEROGENEOUS HTGR SYSTEMS

The latest version of the data flow system (Fig. 3) also starts from the basic ENDF/B file. The MINX code generates temperature-broadened pointwise cross sections in the ENDF/B format (PENDF) and further collapses them to the desired fine-group structure in the Bondarenko[12] energy-shielding formalism. For resonance absorber materials, the PENDF cross sections are space shielded over the entire energy range according to the Wälti formalism[13] adopted in the GA code MICROX to account for the grain heterogeneity in HTGR fuel rods. For this purpose, a special code, PETOPES (Appendix A), was written. The fast-group cross sections were then collapsed by the MINX code to a 69-group fine-group structure (68 equal-lethargy groups from 10 MeV to 0.414 eV, plus 1 dump group) and further collapsed by the 1DX[14] code to the desired broad-group structure (see Table I). Corrections for the second level of heterogeneity (fuel rods in the reactor core) were applied by the rational-approximation collision-probability method of Levine[15] in a modified version of the 1DX code that can handle the Bondarenko formalism provided by the MINX code.

The thermal portion of the PENDF cross sections was processed by a specially written code, ETOGLEN (Appendix B), and by the GLEN code. Since GLEN requires pointwise rather than groupwise cross sections, ETOGLEN was written to select a thermal fine-group structure in such a way as to best calculate resonance integrals by the GLEN method. GLEN calculates a properly hardened thermal neutron spectrum based on the input isotopic composition and collapses the fine-group (points) cross sections to the required few-group thermal structure. GLEN also accepts graphite elastic-scattering cross sections and scattering-law data as calculated, for example, by the HEXSCAT and TOR codes, respectively. MERGFAT (Appendix C) was used to merge the fast and thermal few-group cross sections in the required DTF-IV format.

The operation of the code system shown in Fig. 3 proceeds as follows:

1. Using the basic ENDF/B file as input, a pointwise ENDF/B file (PENDF) is prepared by the MINX code for each nuclide needed in the neutronic calculations. Nuclides prepared for the HTGR composition are shown in Table II. The data in the PENDF files are given at 0, 300, 950, and 3000 K.
2. If the cross sections of a nuclide are not to be grain shielded, the PENDF file is processed directly by the MINX code to give 69-group cross sections for input to the LINX-CINX codes.[16,17] The 69-group structure consists of the GAM-I group structure plus a dump group necessary to obtain the correct eigenvalue in 1DX. The weighting function used in MINX for generating the 69-group set is shown in Fig. 4. It is the composite result of calculations for a typical HTGR system made with the GLEN and MC^2 codes.
3. For those nuclides for which grain shielding is important, the PENDF files are used as input to the PETOPES code, which generates a grain-shielded PENDF file (PENDFS). This file is then used as input to MINX to generate multigroup cross sections as indicated in 2 above.
4. The LINX-CINX codes are used to combine multigroup data for all nuclides into a single data library used for input to the 1DX code. This is the file designated by "A" in Fig. 3. Note that file A contains temperature-dependent f-factors for Bondarenko treatment by 1DX.
5. The ETOGLEN code is used to retrieve cross-section thermal data (from 5×10^{-4} to 2.38 eV in the group structure of Table I) from the PENDF or PENDFS file for each nuclide and to supply pointwise cross sections for elastic scatter-

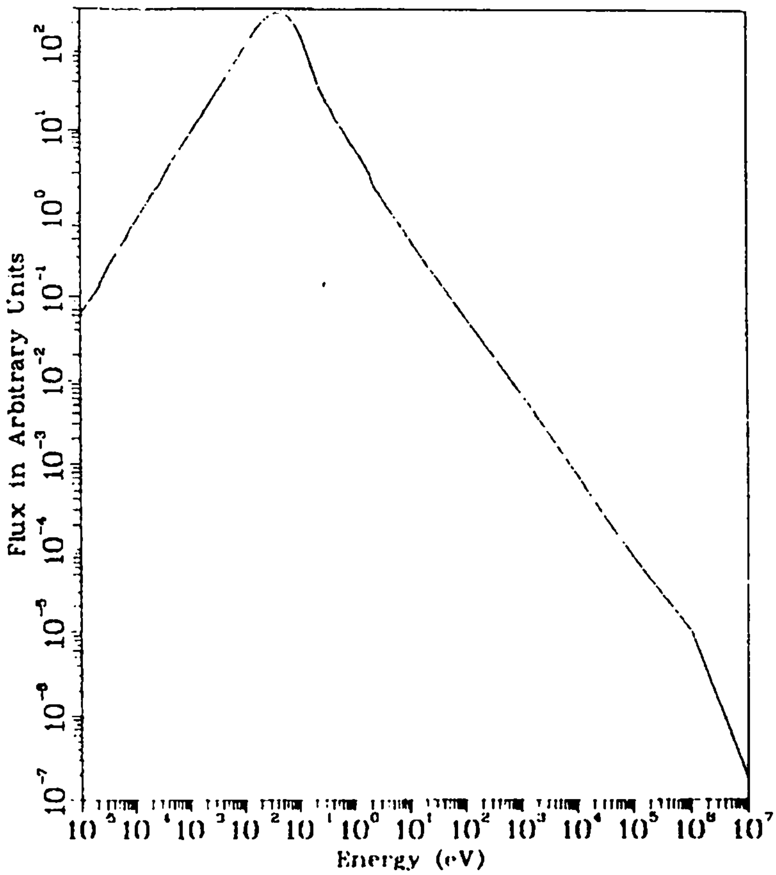


Fig. 4.
Typical HTGR spectrum used for MINX weight function.

ing, nu times fission, and absorption cross sections for the GLEN code. Data for all nuclides at several temperatures (300, 950, 3000 K for HTGR) are combined to form data file "B" in Fig. 3.

6. Data file "C" (graphite in the case of an HTGR) is made by combining the outputs of the TOR and HEXSCAT codes into a single file. For graphite, crystal-lattice parameters are input to HEXSCAT and a phonon distribution (Young-Koppel) is input to TOR. Library "C" contains data for each temperature of interest (see Table II for an HTGR).
7. The final broad-group cross sections for all nuclides at a single temperature, file "D" in Fig. 3, are created using the code MERGFAT to merge the outputs of GLEN and 1DX. This is usually done in a single run for

TABLE II

	Nuclide	MAT NO.	ENDF/B-VERSION	Region
1.	B-10	1155	III	Core
2.	C-12	1165	III	"
3.	O-16	1134	III	"
4.	Si-28	1194	III	"
5.	Xe-135	1294	IV	"
6.	Sm-149	1027	III	"
7.	Th-232	1117	III	"
8.	Pa-233	1119	III	"
9.	Pa-233	1297	IV	"
10.	U-233	1260	IV	"
11.	U-234	1043	III	"
12.	U-235	1157	III	"
13.	U-236	1163	III	"
14.	U-238	1158	III	"
15.	Pu-238	1050	III	"
16.	Pu-239	1264	IV	"
17.	Pu-240	1265	IV	"
18.	Pu-241	1266	IV	"
19.	Pu-242	1161	III	"
20.	B-10	1155	III	reflector
21.	C-12	1165	III	reflector

Cross sections for every nuclide in the above list are available for 12 temperatures: 300, 500, 600, 800, 1000, 1200, 1500, 1700, 2000, 2300, 2600, and 3000 Kelvin.

efficiency purposes. As cross sections at additional temperatures are generated, the data are added to the broad-group cross-section library by means of the UPDATE feature of the LASL CDC-7600 operating software.

IV. DOUBLE-HETEROGENEITY SPACE SHIELDING

Two methods of space shielding cross sections for a doubly heterogeneous reactor system are discussed here. The first method consists of the application of Wälti's[13] method of grain shielding to pointwise (PENDF) cross sections followed by the application of the Levine[15] formalism of "gross" (fuel-rod) space shielding to collapsed grain-shielded fine-group cross sections. The grain shielding was implemented in the PETOPES code and the gross heterogeneity correction was made in a modified 1DX code.

The second method of space shielding cross sections is a newly developed method based on rational approximations and collision probabilities which accounts for both levels of heterogeneity at the fine-group cross-section level. It, therefore, bypasses the time-consuming pointwise grain-shielding process and it serves as independent reference, since it produces results in close agreement with the first method.

A. First Method

1. Grain-Shielding Treatment. Wälti's grain-shielding method has been incorporated in the GA code MICROX and produces, according to Wälti's claims, results in close agreement with the detailed Nordheim integral method (NIT) used in the GAROL[18] and the GGC-5[19] codes.

In the Wälti procedure, the grain-shielded absorption cross section is given by

$$\sigma_i^{eff}(E) = \sigma_i(E) \frac{\Gamma(E)}{1 - r^3[1 - \Gamma(E)]} , \tag{4}$$

where

$\sigma_i(E)$ = unshielded energy-dependent cross section for the i-th heavy nuclide;

r = ratio of fuel-to-moderator radii in a two-concentric-sphere model (inner = fuel; outer = moderator) representing a uniform grain distribution in the fuel rod; and

$\Gamma(E)$ = self-shielding factor, i.e., the ratio of average neutron fluxes in the grain and in the moderator, $\bar{\Phi}_0/\bar{\Phi}_1$, where subscripts 0 and 1 refer to the grain and the surrounding moderator regions, respectively.

If, due to the presence of large amounts of moderator material, isotropic angular fluxes are assured for regions 0 and 1, the neutron balance equations for the two regions yield

$$\Gamma(E) = \frac{\bar{\Phi}_0}{\bar{\Phi}_1} = \frac{1 + \rho Q[1 + W\bar{\ell}_1(\Sigma_{a,1} + \Sigma_{out,1})]}{1 + \rho Q + W\,\bar{\ell}_0(\Sigma_{a,0} + \Sigma_{out,0})} , \tag{5}$$

where

$$\rho = \frac{\overline{\ell}_0}{\overline{\ell}_1} = \frac{V_0}{V_1} = \text{volume ratio of regions 0 and 1,}$$

Q = ratio of spatially averaged source densities in regions 0 and 1,

$$W = 1 + \overline{H}_0(\Sigma_{t,0}) + \overline{H}_1(\Sigma_{t,1}) \quad , \tag{6}$$

$\overline{\ell}_0, \overline{\ell}_1$ = mean chord lengths in regions 0 and 1, respectively;

$$\overline{\ell}_j = \frac{4Vj}{S_j} \quad , \; j = 0,1 \quad .$$

The first-collision "augment" for region j, $\overline{H}_j$ is given by

$$\overline{H}_j(\Sigma_{t,j}) = \frac{1 - \overline{P}_j}{\overline{\ell}_j \overline{P}_j \Sigma_{t,j}} \quad , \; j = 0,1 \quad , \tag{7}$$

and $\Sigma_{a,j}$, $\Sigma_{out,j}$, and $\Sigma_{t,j}$ are the macroscopic absorption, outscatter, and total group cross sections, respectively, for region j (0 or 1).

Augment $\overline{H}_1(\Sigma_t)$ can be approximated by $\overline{H}_1(0)$ which is given by the following expression

$$\begin{aligned}\overline{H}_1(0) = \left(\frac{\gamma}{r}\right)^2 \Bigg\{ & (1 - r^2)^2 \left(1 + \frac{1}{4} \ln \frac{1+r}{1-r}\right) - \frac{r}{2}\,(1 - r)^2 \\ & + \left(\frac{2}{3r}\right)^2 \left[(1 - r^2)^3 - 3(1 - r^3)^2 + 2(1 - r^3)(1 - r^2)^{3/2}\right]\Bigg\} \quad ,\end{aligned} \tag{8}$$

where

$$\gamma = \frac{3r^2}{4(1 - r^3)} \quad . \tag{9}$$

The escape probability function $\overline{P}_0$ is given by the expression of Case _et al._[20]

$$P_0(\Sigma_{t,0}) = \frac{3}{8X^3}\left[2X^2 - 1 + (1 + 2X)\exp(-2X)\right] \quad , \tag{10}$$

where

$$X = \frac{3}{4} \bar{\ell}_0 \, \Sigma_{t,0} \quad . \tag{11}$$

Source density ratio Q can be calculated from

$$Q = \frac{\xi_{0,pot} \Sigma_{S,0}^{pot}}{\xi_{1,pot} \Sigma_{S,1}^{pot}} \quad , \tag{12}$$

and the self-scattering cross section at the pointwise level is approximated by

$$\Sigma_{SS,j}(E) \simeq \frac{1 - \xi_j(E)}{\xi_1^{pot}} \Sigma_{S,j}(E) \quad , \quad j = 0, 1 \quad , \tag{13}$$

where the average logarithmic energy decrement $\xi_j(E)$ is given by

$$\xi_j(E) = \frac{\sum_i \xi_j^i \, \Sigma_{S,j}^i(E)}{\Sigma_{S,j}(E)} \quad , \; j = 0, 1 \quad , \tag{14}$$

i being the nuclide index.

The derivations of these equations and the justifications for the approximations made can be found in Wälti's paper.[13] The above summary of the theory has been included only for readers' convenience. The programming of the equations in the PETOPES code is discussed in Appendix A.

2. Fuel-Rod Heterogeneity Treatment. The escape probability from a regular array of fuel (absorber) lumps, each assumed to be homogeneous in composition, is given by the Nordheim expression

$$P^*_{esc} = P_{esc} \frac{1 - C}{1 - C(1 - \Sigma_F \bar{\ell}_F \, P_{esc})} \quad , \tag{15}$$

where

P_{esc} = escape probability from one lump,

C = Dancoff factor (Appendix D), and

$\bar{\ell}_F$ = fuel-rod mean chord length.

Equations for P_{esc} for different lump geometries have been derived by many investigators (e.g., see Refs. 20, 21, 22). Wigner[23] has proposed a "rational" approximation to P_{esc} which gives the correct value in the two limiting cases of very large and very small lumps. For better approximations between these two extreme limits, various Wigner-like approximations have been proposed. One such popular approximation is due to Levine[15] and is given by the following expression

$$P_{esc} = \frac{1}{1 + \frac{\Sigma_F \bar{\ell}_F}{A}} , \tag{16}$$

where A = Levine factor (fuel-rod-geometry dependent). Equation (16) preserves the convenient form of the Wigner rational expression at the two extreme limits and, in addition, it provides good values of P_{esc} for intermediate-size lumps. Incidentally, for A equal to unity, Eq. (16) reduces to Wigner's approximation.

For cylindrical rods, Otter[24] has found that the energy-independent value of 1.35 for A works quite well for a wide range of fuel-rod radii. When Eq. (16) is substituted into Eq. (15), the resulting expression for P^*_{esc} is

$$P^*_{esc} = \frac{1}{1 + \frac{\Sigma_F}{\Sigma_e}} , \tag{17}$$

where the effective cross section Σ_e is given by

$$\Sigma_e = \frac{A(1 - C)}{\bar{\ell}_F[1 + C(A - 1)]} . \tag{18}$$

The advantage of the rational form of Eq. (17) is the equivalence between the given heterogeneous system and a corresponding homogenized system for which the moderator cross section equals the moderator cross section in the fuel rod of the heterogeneous system plus the effective cross section Σ_e.[25,26] This implies that fuel-rod heterogeneity corrections to homogeneous cross sections can be made by adding Σ_e to the fuel-rod moderator cross section and treating the reactor system as homogeneous.

This formalism has been discussed in detail elsewhere[25,26,27] and has been included in a modified version of the 1DX code.

B. Second Method

The second method is in a way an extension of the fuel-rod heterogeneity correction and accounts for both levels of heterogeneity by means of collision probabilities and rational approximations.

From results of the first method, we have found that corrections associated with the "fine" (grain) heterogeneity in HTGR rods of the type under consideration (containing low-volume fractions of 200- to 500-μm-diam grains) is considerably smaller than the "gross" (fuel-rod) heterogeneity correction. Consequently, it would be possible to extend the rational-approximation collision-probability methods of the "gross" heterogeneity correction in order to account for both levels of heterogeneity. The method is briefly as follows.

Let us first define the following quantities:

P_E^* = neutron escape probability from the fuel in the reactor core,

P_e = escape probability from one grain for neutrons uniformly and isotropically produced in that homogeneous grain,

P_E = escape probability from a homogenized fuel rod for neutrons produced uniformly and isotropically in that fuel rod,

f_0 = volume fraction of the grains in one fuel rod,

P_F = probability that a neutron incident on a fuel rod collides in that fuel rod,

P_M = probability that a neutron leaving a fuel rod collides in the moderator outside that rod,

P_0 = probability that a neutron incident on a fuel grain collides in that grain,

P_1 = probability that a neutron leaving a fuel grain collides in the moderator outside it but inside the fuel rod in which the grain is,

P_E' = neutron escape probability from a fuel rod for neutrons produced in the grains of that fuel rod,

P_{ge} = probability that a neutron from the moderator outside any grain will escape from the fuel rod in which that grain is.

From these definitions, it immediately follows that

$$C = 1 - P_M \tag{19}$$

and

$$C_0 = 1 - P_1 \quad , \tag{20}$$

where

C = Dancoff factor of the regular array of fuel rods in the reactor core, and

C_0 = Dancoff factor of the grains in a fuel rod, i.e., the probability that a neutron leaving a grain will next collide with another grain of the same fuel rod.

From reciprocity theorems,[20] it also follows that

$$P_F = \Sigma_F \bar{\ell}_F P_E \tag{21}$$

and

$$P_0 = \Sigma_0 \bar{\ell}_0 P_e \quad , \tag{22}$$

where

Σ_0 = macroscopic fuel-grain cross section,

$\bar{\ell}_0 = \dfrac{4V_0}{S_0}$ = mean chord length of a grain of volume V_0 and surface area S_0; for a spherical grain of radius R, $\bar{\ell}_0 = (4/3)$ R.

The overall neutron escape probability is given by:

$$P_E^* = P_E'[P_M + (1 - P_M)(1 - P_F)P_M + \cdots\cdot] = P_E' \frac{P_M}{1 - (1 - P_M)(1 - P_F)} \quad , \tag{23}$$

or, combining Eqs. (19), (21), and (23), one obtains

$$P_E^* = P_E' \frac{1 - C}{1 - C(1 - \Sigma_F \bar{\ell}_F P_E)} \quad . \tag{24}$$

The rational approximations for P_E and P_e are

$$P_E = \frac{1}{1 + \frac{\Sigma_F \bar{\ell}_F}{A}} \tag{25}$$

and

$$P_e = \frac{1}{1 + \frac{\Sigma_0 \bar{\ell}_0}{a}} \quad , \tag{26}$$

where A is the rod-geometry-dependent Levine factor[15] with the recommended value[24] of 1.35 for cylindrical rods. Parameter "a" can be obtained by "rationalizing" Eq. (10) to give

$$P_e^{sph} \simeq \frac{1}{1 + \frac{9}{16} \Sigma_0 \bar{\ell}_0} \quad , \tag{27}$$

i.e., assigning the value of 16/9 to the Levine-like parameter "a."

We can evaluate P_E' from the series: .

$$P_E' = P_e[P_1 P_{ge} + (1 - P_1)(1 - P_0) P_1 P_{ge} + \cdots] = P_e \frac{P_1 P_{ge}}{1 - (1 - P_1)(1 - P_0)} \quad , \tag{28}$$

which, after combining Eqs. (20), (26), (22), and (28), yields

$$P_E' = \frac{P_{ge}}{1 + \Sigma_0 \bar{\ell}_0 \left[\frac{1}{a} + \frac{C_0}{1 - C_0}\right]} \quad . \tag{29}$$

If we now treat the grains-in-the-fuel-rod configuration as a perturbation of the homogeneous rod model, we can replace Eq. (29) by the approximate expression

$$P_E' \approx \frac{P_E}{1 + \Sigma_F \bar{\ell}_0 \left[\frac{1}{a} + \frac{C_0}{1 - C_0}\right]} \quad . \tag{30}$$

Equations (24), (25), and (30) can be combined to give:

$$P_E^* = \frac{1}{\left[1 + \Sigma_F \bar{\ell}_0\left(\frac{1}{a} + \frac{C_0}{1 - C_0}\right)\right]\left[1 + \Sigma_F \bar{\ell}_F\left(\frac{1}{A} + \frac{C}{1 - C}\right)\right]} \quad , \tag{31}$$

which after neglecting second-order terms yields

$$P_E^* = \frac{1}{1 + \Sigma_F \bar{\ell}_F\left(\frac{1}{A^*} + \frac{C}{1 - C}\right)} \quad , \tag{32}$$

where

$$\frac{1}{A^*} = \frac{1}{A} + \frac{\bar{\ell}_0}{\bar{\ell}_F}\left(\frac{1}{a} + \frac{C_0}{1 - C_0}\right) \quad . \tag{33}$$

Equation (32) preserves the rational form of Eq. (16) and corrects for both levels of heterogeneity provided that the Levine parameter A is replaced by the new grain-dependent parameter A^* given by Eq. (33). Equation (33) can be written as

$$P_E^* = \frac{1}{1 + \frac{\sigma_F}{\sigma_{eff}}} \quad , \tag{34}$$

where

$$\sigma_{eff} = \frac{1}{N_F \bar{\ell}_F\left(\frac{1}{A} + \frac{C}{1 - C}\right)} \quad . \tag{35}$$

N_F = absorber atomic density in the fuel rod. All the σ's are microscopic cross sections per absorber atom. The new quantity σ_{eff} can then replace Σ_e/N_F of Eq. (18) in the single-heterogeneity correction discussed in Sec. IV.A.2 to yield double-heterogeneity corrections.

This method can be easily incorporated in codes like MC^2-I or 1DX without need of pointwise cross sections as required by the first double-heterogeneity shielding method discussed in Sec. IV.A.1.

A similar space shielding method was developed earlier and is discussed in Ref. 28. The grain Dancoff factor calculation necessary for Eq. (33) is derived in Refs. 28 and 29 and is given by:

$$C_0 = \frac{\Sigma_g}{\Sigma_f}\left\{1 - \left[1 + \frac{\Sigma_f \bar{\ell}_F}{m+1}\right]^{-(m+1)}\right\} \quad , \tag{36}$$

where

$$\Sigma_g = n\bar{\sigma}_0 \quad , \tag{37}$$

$$\Sigma_f = \Sigma_g + \Sigma_{mod} \quad , \tag{38}$$

$$\Sigma_{mod} = N_1\sigma_1 \quad , \tag{39}$$

and

N_1 = atomic density of fuel-rod moderator outside the grains,

σ_1 = fuel-rod moderator microscopic cross section,

$n = f_0/V_0$ = number of grains per unit volume of the fuel rod,

$\bar{\sigma}_0 = \frac{S_0}{4}$ = average "geometric" cross section of the grains,

$m = 3.58$.

If scattering effects in the fuel grains are considered, parameter "a" should be replaced[28,29] by group parameter a^*:

$$a^* = \frac{a}{1 - q} \quad , \tag{40}$$

where q is the ratio of the self-scattering cross section to the total cross section in a particular group. Scattering effects in fuel grains are generally of relatively small importance for the HTGR rods under consideration.

C. Comparisons and Discussion

The above double-heterogeneity space-shielding methods were used for generating above-thermal few-group ^{232}Th, ^{235}U, and ^{233}U cross sections for a 3000-MW(th) HTGR system with fuel rods containing 500-and 200-μm-diameter ThO_2 and UC_2 grains, respectively, in a graphite matrix. The most affected in the above-thermal region is the ^{232}Th absorption cross section of group 3 (in the group structure of Table I), which incorporates all resolved resonances of Thorium. Table III shows a comparison of the group-3 absorption cross sections at 3 temperatures (300, 800, and 1200 K) as calculated by the first method (Sec. IV. A), by the second method (Sec. IV. B), and by the GA code MICROX (GA results supplied to LASL on magnetic tape). A non-grain-shielded absorption cross section (NGSX) is also included for comparison. The grain-shielding effect is seen to be of the order of 4-5% by comparison with the fuel-rod shielding effect, which was seen to be ~25%. In the thermal region, the space shielding of the ^{233}U and ^{235}U absorption cross sections (^{232}Th is not important in the thermal region) was seen to be considerably less important.

TABLE III

RESOLVED-RESONANCE-GROUP ABSORPTION CROSS SECTION IN ^{232}Th (b)

Temperature (K)	1st Method	2nd Method	MICROX	NGSX
300	6.58	6.72	6.76	6.95
800	7.82	8.03	8.12	8.28
1200	8.42	8.65	8.78	8.90

APPENDIX A

PETOPES PROGRAM

The purpose of the PETOPES program is to change a PENDF tape to a PENDF shielded tape; that is, to produce a pointwise tape in the ENDF/B format containing grain-shielded cross sections from a pointwise ENDF/B tape originally produced by the MINX[11] code. The shielded data can then be used as input to the MINX code to obtain multigroup grain-shielded cross sections.

The grain-shielding technique used in PETOPES is that suggested by Wälti.[13] Although the theory is discussed in detail in the text, the formulas used in the Wälti treatment are repeated here in a notation mnemonically compatible with that used in the code. Grain shielding may be accounted for by noting that the effective resonant material (e.g., thorium in the HTGR) cross section is given by

$$\sigma_{Th}^{eff} = \sigma_{Th} \frac{V_f}{V_c} \frac{\Gamma(E)}{1 + \frac{V_p}{V_c}\Gamma(E)} \quad , \tag{A-1}$$

where σ_{Th} is the unshielded cross section, V_f, V_p, and V_c are the relative volumes of fuel, particle, and moderator regions, respectively, and $\Gamma(E)$ is the energy-dependent disadvantage factor for the particle relative to the remainder of the fuel element. $\Gamma(E)$ depends on the energy-dependent total and scattering cross sections of the resonant material and on other parameters which are insensitive to energy. $\Gamma(E)$ is given by Wälti as

$$\Gamma(E) = \frac{1 + \frac{V_p}{V_c} Q(1 + \tau_{x,c} W)}{1 + \frac{V_p}{V_c} Q + \tau_{x,p} W} \quad , \tag{A-2}$$

where p refers to the particle region, c refers to the moderator region, ξ's are

$$\tau_{x,j} = \tau_{t,j}\left[1 - \left(1 - \frac{\xi_j}{\xi_c^{pot}}\right)\frac{\Sigma_{sj}}{\Sigma_{tj}}\right] \qquad j = p,c \quad , \tag{A-3}$$

the logarithmic slowing-down decrements for each region, and Σ_s and Σ_t are macroscopic scattering and total cross sections, respectively, for the resonant material in each region. Note that for region c the potential scattering cross section is used to evaluate ξ, so that this quantity is energy independent in the moderator region.

$$\tau_{t,j} = \frac{4V_j}{S_j} \Sigma_{t,j} \quad , \quad j = p,c \quad , \tag{A-4}$$

where S refers to the surface areas of the regions.

$$W = 1 + \tilde{H}_0(\tau_{t,p}) + \tilde{H}_1(\tau_{t,c}) \quad . \tag{A-5}$$

$$\tilde{H}_0(\tau_{t,p}) = \frac{1 - \tilde{P}_0(\tau_{t,p})}{\tau_{t,p}\tilde{P}_0(\tau_{t,p})} \quad . \tag{A-6}$$

$$\tilde{P}_0(\tau_{t,p}) = \frac{3}{8X^3} \left[2X^2 - 1 + (1 + 2X)e^{-2X}\right] \quad , \quad X = \frac{3}{4} \tau_{t,p} \quad . \tag{A-7}$$

$$\begin{aligned} \tilde{H}_1(\tau_{t,c}) = \left(\frac{\gamma}{r}\right)^2 \Bigg\{ &(1 - r^2)^2 \left(1 + \frac{1}{4} \ln \frac{1 + r}{1 - r}\right) - \frac{r}{2} (1 - r)^2 \\ &+ \left(\frac{2}{3r}\right)^2 \left[(1 - r^2)^3 - 3(1 - r^3)^2 + 2(1 - r^3)(1 - r^2)^{3/2}\right] \Bigg\} . \end{aligned} \tag{A-8}$$

$$r = R_0/R_1 \quad , \tag{A-9}$$

where R_0 and R_1 are outer radii of regions p and c, respectively.

$$\gamma = \frac{3r^2}{4(1 - r^3)} \quad . \tag{A-10}$$

$$Q = \frac{\xi_p^{pot} \ \Sigma_p^{pot}}{\xi_c^{pot} \ \Sigma_c^{pot}} \quad . \tag{A-11}$$

Also, the cross-section weighted logarithmic decrements for the mixtures in each region are given by

$$\xi_j = \frac{\sum_k \xi_k N_k^j \sigma_{sk}}{\sum_k N_k^j \sigma_{sk}} \quad , \tag{A-12}$$

where the N_k are the concentration and σ_{sk} the scattering cross sections for isotopic constituents of the regions.

The basic input to the PETOPES code is a PENDF file output by the MINX code. This file usually consists of the cross-section data for a particular nuclide (e.g., ^{232}Th) given for several temperatures. The object of the PETOPES code is to calculate a grain-shielding factor (Eq. A-1) at each energy point in the PENDF file, multiply this factor by the cross section at the given energy, and prepare a new file of the grain-shielded cross sections. This is done for every temperature on the tape. If there is more than one nuclide in a mixture contributing to the grain shielding, a preparatory routine, DBLSHLD, is called which prepares a cross-section file used in calculating the shielding factors according to the formula:

$$\sigma_{eff} = \sum_{i=1}^{n} N_i \sigma_i \quad , \tag{A-13}$$

where σ_{eff} is the effective cross section for calculating the self-shielding factor at a particular energy point; n the number of nuclides in the mixture contributing to the self-shielding; N_i the fraction of the i-th nuclide in the mixture, and σ_i the cross section of the i-th nuclide at the energy point in question.

In the data input to the PETOPES code, only the cross-section data for the material for which grain-shielded cross sections are being prepared are assumed to be energy dependent. Total and potential cross sections as well as logarithmic decrements for other materials in the mixtures are assumed to be energy independent. Other input parameters are the radii of the particle and moderator regions and the concentrations of the constituents of particle and surrounding moderator regions. Also the energy range over which the grain shielding is applied is specified. Input specifications are given in Table A-I.

TABLE A-I

PETOPES INPUT SPECIFICATIONS

Card No.	Format	Variable	Comment
1	6A10	A(I)	Title card.
2	6E11.4	RADP	Radius of particle region.
		RADC	Radius of moderator region.
		EMAX	Upper energy bound of resonance region.
		EMIN	Lower energy bound of resonance region.
3	6I11	NMP	No. of materials in particle region.
		NMC	No. of materials in moderator region.
		NOQCAL	Obsolete.
4	6E11.4	PSIP(I)	NMP values of ξ_i for the materials in particle region. Note I = 1 is always material for which grain-shielded cross sections are being produced, e.g., Th.
5	6E11.4	PSIC(I)	NMC values of ξ_i for the materials in moderator region. Note I = 1 is always for the moderating material, e.g., c.
6	6E11.4	CONP(I)	NMP concentrations for the materials in the particle region. Order same as for PSIP.
7	6E11.4	CONC(I)	NMC concentrations for the materials in the moderator region. Order same as for PSIC.
8	6E11.4	XSP(I),XP(I)	NMP values for total and potential cross sections for materials in particle region. Order same as for PSIP but XSP(I) and XP(I), for the grain-shielded material, are not used because the energy-dependent cross sections are read from input tape.
9	6E11.4	XSC(I),XC(I)	NMC values for total and potential cross sections for materials in the moderator region. Order same as for PSIC.

Comparison of $\Gamma(E)$ as computed by the PETOPES with a calculation of Wälti's[13] for the 21.8 and 23.5 eV ^{232}Th resonances for ThC_2 particles is shown in Fig. A-1. The agreement is good and differences are attributed to the fact that a different evaluation for ^{232}Th (ENDF/B-III) was used in the PETOPES code from that used by Wälti. This is evident from the fact that the resonances occur at slightly different energies. Figure A-2 shows the variation of $\Gamma(E)$ with temperature for the same two resonances.

A listing of the PETOPES code is given at the end of this appendix. In addition to the grain-shielded file output by the code, printed output includes the input and a limited number of grain-shielding factors and values of $\Gamma(E)$ for each temperature. Plots are also made of these for the various temperatures.

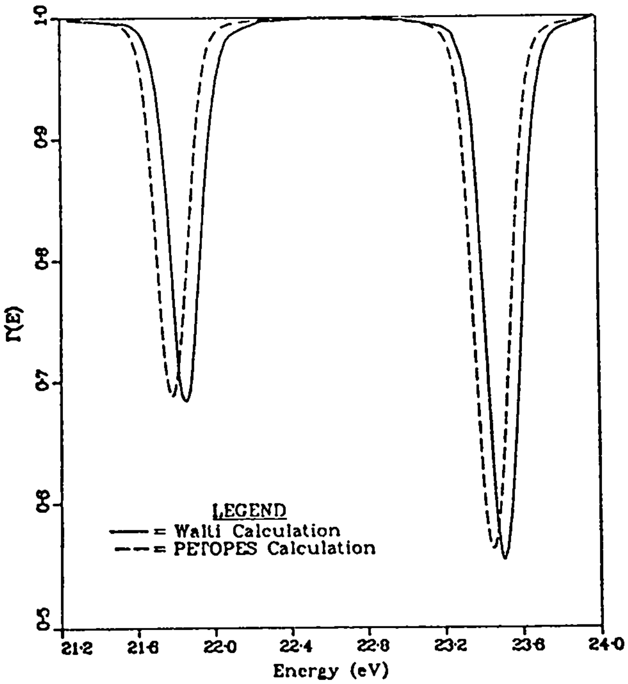


Fig. A-1.
Comparison of Wälti and PETOPES calculations for $\Gamma(E)$ for the 21.8-and 23.5-eV resonances of ^{232}Th at 300 K.

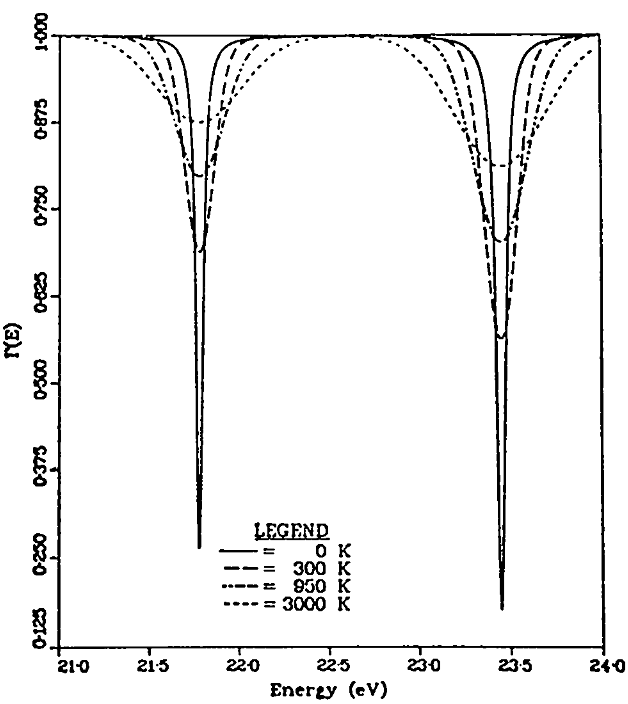


Fig. A-2.
$\Gamma(E)$ for 0, 300, 950, and 3000 K for the 21.8-and 23.5-eV resonances of ^{232}Th.

```
      PROGRAM PETOPES (INP,OUT,FSET5=INP,FSET6=OUT,FSET10,FSET11,FSET12,PETOP
     1    FILM,FSET9)                                                    PETOP
C     PURPOSE OF PROGRAM - TO CONVERT A PENDF TAPE TO SHIELDED PENDF.    PETOP
C     PENDF TO PENDF-SHIELDED.                                           PETOP
C     --     -- --    -                                                  PETOP
      LCM/XSECTT/XT(60000),YT(60000),NPTT                                PETOP
      LCM/XSECTE/XE(60000),YE(60000),NPEE                                PETOP
      COMMON/CONS/RADC,RADP,VOLC,VOLP,SURC,SURP,PSIP(10),PSIC(10),EMAX,  PETOP
     1    EMIN,MT                                                        PETOP
      COMMON/CALC/H1TAU,VOLF,Q,SEEP,SEEC,SIGPSP,SIGPSC,TAUTC,TAUXC       PETOP
      COMMON/CON1/CONP(10),CONC(10),XSC(9),XSP(9),NMP,NMC,XP(9),XC(9)    PETOP
      COMMON/PLOTS/ENG(5000),FAX(5000),GAMX(5000),NX,TITL(5),XLB(5),     PETOP
     1    YLB(5)                                                         PETOP
      DIMENSION F(10),S(10),J(10),A(8),HOL(7),X(10),Y(10)                PETOP
    4 READ (11,15) (A(I),I=1,6),ANEXT1,MCHECK,ANEXT2                     PETOP
    5 WRITE (10,15) (A(I),I=1,6),ANEXT1,MCHECK,ANEXT2                    PETOP
      IF (MCHECK.EQ.4H  -1)  GO TO 6                                     PETOP
      GO TO 4                                                            PETOP
    6 END FILE 10                                                        PETOP
      REWIND 10                                                          PETOP
      REWIND 11                                                          PETOP
   15 FORMAT (6A10,A6,A4,A10)                                            PETOP
C                                                                        PETOP
C     INPUT DEFINITIONS-                                                 PETOP
C     RADP - RADIUS OF PARTICLE,E.G. THORIUM CORE  OF THORIUM COATED     PETOP
C          PARTICLE IN HTGR                                              PETOP
C     RADC - RADIUS OF EFFECTIVE SPHERICAL SHELL,E.G. RADIUS OF EFFECTIVPETOP
C          MEDIA SURROUNDING THORIUM CORE IN HTGR FUEL ELEMENT.          PETOP
C     VOLP-PARTICLE VOLUME CORRESPONDING TO RADP.                        PETOP
C     VOLC-VOLUME CORRESPONDING TO MEDIA SURROUNDING PARTICLE REGION.    PETOP
C     SURP-SURFACE AREA OF PARTICLE.                                     PETOP
C     SURC-SURFACE AREA OF SURROUNDING MEDIA.                            PETOP
C     PSIP- LOG-DEC (MT252) FOR MATERIALS IN PARTICLE REGION,E.G.        PETOP
C            FOR THORIUM PSIP= 0.008669.                                 PETOP
C     PSIC- -LOG-DEC (MT252) FOR MATERIALS OUTSIDE,PARTICLE REGION. FOR  PETOP
C            CARBON,PSIC= 0.1589.                                        PETOP
C     NMP=NO OF MATS IN PARTICLE REGION.                                 PETOP
C     NMC=NO OF MATS IN OUTER REGION.                                    PETOP
C     CONP-ATOMS/CC OF MATS IN PARTICLE REGION.  CONP(1) IS FOR THORIUM  PETOP
C     CONC-ATOMS/CC OF MATS OUTSIDE PARTICLE REGION.                     PETOP
C     XSP,XP-TOT.POT XSEC FOR MATERIALS WITH CONSTANT XSEC IN PARTICLE   PETOP
C          REGION.  XSP(1),XP(1),ARE FOR THORIUM-COMPUTED IN GRANSHL.    PETOP
C     XSC,XC-TOT.POT XSEC FOR MATERIALS OUTSIDE PARTICLE REGION.         PETOP
C     EMAX-ENERGY BOUNDING RESONANCE REGION FOR PARTICLE SHIELDING,E.G.  PETOP
C          EMAX=4.0 KEV FOR THORIUM.                                     PETOP
C     EMIN- LOWER BOUND OF RESONANCE REGION,E.G.,EMIN=21EV FOR TH-232.   PETOP
C     NOQCAL = 0 FOR FERTILE MATS,E.G. THORIUM IN RES. REGION.           PETOP
C            = 1 FOR FISSILE MATS,E.G. U-235 AND U-233 IN THERMAL        PETOP
C              REGION ONLY.(NOTE THERMAL REGION MUST BE RUN SOLO         PETOP
C              BECAUSE OF THIS)                                          PETOP
      READ (5,15) (A(I),I=1,6)                                           PETOP
      WRITE (6,15) (A(I),I=1,6)                                          PETOP
      READ (5,18) RADP,RADC,EMAX,EMIN                                    PETOP
    7 FORMAT (1H0,*  INPUT*//*    RADP = *1PE12.5,* RADC = *1PE12.5,     PETOP
     1    * EMAX = *1PE12.5,* EMIN = *1PE12.5,* NOQCAL = *I3)            PETOP
   18 FORMAT (6E11.4)                                                    PETOP
      READ (5,19) NMP,NMC,NOQCAL                                         PETOP
      WRITE (6,7) RADP,RADC,EMAX,EMIN,NOQCAL                             PETOP
   19 FORMAT (6I11)                                                      PETOP
      READ (5,18) (PSIP(I),I=1,NMP)                                      PETOP
      READ (5,18) (PSIC(I),I=1,NMC)                                      PETOP
      READ (5,18) (CONP(I),I=1,NMP)                                      PETOP
```

```
      READ (5,18) (CONC(I),I=1,NMC)
      READ (5,18) (XSP(I),XP(I),I=1,NMP)
      READ (5,18) (XSC(I),XC(I),I=1,NMC)
C
C     THESE CONSTANTS ARE NEEDED IN SUBROUTINE GRANSHL.
C
      VOLP=4./3.*3.14159*RADP**3
      VOLC=4./3.*3.14159*RADC**3 -VOLP
      SURP=4.*3.14159*RADP**2
      SURC=SURP
      R=RADP/RADC
      GAM=3*R**2/(4.0*(1.0-R**3))
      TRM1=(1.0-R**2)**2*(1.0+0.25*ALOG((1.0+R)/(1.0-R)))-
     1     0.5*R*(1.0-R)**2
      TRM2=(2.0/(3.0*R))**2
      TRM3=(1.0-R**2)**3-3.0*(1.0-R**3)**2+2.0*(1.0-R**3)*(1.0-R**2)**1.
     1 5
      HITAU=(GAM/R)**2*(TRM1+TRM2*TRM3)
      VOLF=VOLC+VOLP
C
C     CALCULATE Q.
C
      SEENUM=0.
       SIGPSP=0.
      DO 130 I=1,NMP
      SEENUM=PSIP(I)*CONP(I)*XP(I)+SEENUM
      SIGPSP=CONP(I)*XP(I)+SIGPSP
  130 CONTINUE
      Q=SEENUM
      SEEP=SEENUM/SIGPSP
      SEENUM=0.
      SIGPSC=0.
      SIGTC=0.
      DO 140 I=1,NMC
      SEENUM=PSIC(I)*CONC(I)*XC(I)+SEENUM
      SIGPSC=CONC(I)*XC(I)+SIGPSC
      SIGTC=CONC(I)*XSC(I)+SIGTC
  140 CONTINUE
      SEEC=SEENUM/SIGPSC
      Q=Q/SEENUM
      TAUTC=4.0*VOLC*SIGTC/SURC
      TAUXC=TAUTC
C
C     END OF Q CALCULATION
C
      TITL(1)=10HGAMMA PLOT
      TITL(2)=10H TO COMPAR
      TITL(3)=10HE WITH OTH
      TITL(4)=10HR METHODS.
      XLB(1)=10HENERGY IN
      XLB(2)=10HE.V. UNITS
   10 READ (11,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ
      HOL(1)=10H  THIS TAP
      HOL(2)=10HE HAS BEEN
      HOL(3)=10H CHANGED T
      HOL(4)=10HO A PENDF-
      HOL(5)=10HSHIELDED F
      HOL(6)=10HILE.
   20 FORMAT (6A10,A6,I4,I2,I3,I5)
      WRITE (12,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ
      READ (10,20) DUM
      READ (10,80) ZA,AWR
      CALL STORXS
```

```
      NX=0
      PRINT 2020,MAT
 2020 FORMAT (1H1,*WELL,WE MADE IT OUT OF STORXS ONCE,  MAT=*I4)
   30 READ (11,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ
      IF (MAT.EQ.0)  CALL STORXS
      IF (MAT.EQ.0) NX=0
 2030 FORMAT (1H ,*WE ARE LOOPING NOW,  MAT=*I4)
      WRITE (12,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ
      IF (MAT.EQ.-1) GO TO 2000
      IF (MF.NE.3) GO TO 30
      IF (MT.EQ.1)  GO TO 31
      IF (MT.EQ.2)  GO TO 31
      IF (MT.EQ.3)  GO TO 31
      IF (MT.EQ.18)  GO TO 31
      IF (MT.EQ.102)  GO TO 31
      GO TO 30
   31 CONTINUE
      MTXX=MT
      READ (11,40) C1,C2,N1,N2,N3,N4,MAT,MF,MT,NSEQ
      CALL CXFP (C1,F(1),S(1),J(1))
      CALL CXFP (C2,F(2),S(2),J(2))
      WRITE (12,50) (F(I),S(I),J(I),I=1,2),N1,N2,N3,N4,MAT,MF,MT,NSEQ
   40 FORMAT (1P2E11.4,4I11,I4,I2,I3,I5)
   50 FORMAT (2(F8.5,A1,I2),4I11,I4,I2,I3,I5)
      READ (11,60) NPT,INT,N0,N0,N0,N0,MAT,MF,MT,NSEQ
      WRITE (12,60) NPT,INT,N0,N0,N0,N0,MAT,MF,MT,NSEQ
      WRITE (6,100) C1,MAT,MT
   60 FORMAT (6I11,I4,I2,I3,I5)
      NN1=1
   70 NN2=NN1+2
      READ (11,80) (X(I),Y(I),I=1,3),MAT,MF,MT,NSEQ
   80 FORMAT (1P6E11.4,I4,I2,I3,I5)
      LOOP=0
      DO 85 I=1,3
      E=X(I)
      CALL GRANSHL (E,FACT)
      Y(I)=Y(I)*FACT
      LOOP=LOOP+1
   85 CONTINUE
      CALL CXFP (X(1),F(1),S(1),J(1))
      CALL CXFP (Y(1),F(2),S(2),J(2))
      CALL CXFP (X(2),F(3),S(3),J(3))
      CALL CXFP (Y(2),F(4),S(4),J(4))
      CALL CXFP (X(3),F(5),S(5),J(5))
      CALL CXFP (Y(3),F(6),S(6),J(6))
      WRITE(12,90) (F(I),S(I),J(I),I=1,6),MAT,MF,MT,NSEQ
   90 FORMAT (6(F8.5,A1,I2),I4,I2,I3,I5)
   95 FORMAT (*  M = *I6,*  E = *1PE12.5,*  FACT = *1PE12.5)
  100 FORMAT (1H1,*  TEMPERATURE = *1PE12.5,*  MAT = *I4,*  MT = *I3)
      NN1=NN2+1
      IF (NN1.LE.N4) GO TO 70
      READ (11,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ
      WRITE (12,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ
      IF (MTXX.GT.1)  GO TO 30
      WRITE (6,200) NX,C1
  200 FORMAT (1H1,*  NX= *I6,* FOR TEMP = *1PE12.5//7X,*ENERGY*,15X,
     1     *FACT*13X,*GAMMA*)
      WRITE (6,210) (ENG(N),FAX(N),GAMX(N),N=1,NX)
  210 FORMAT (1P3E18.5)
      WRITE (9) NX,(ENG(N),FAX(N),GAMX(N),N=1,NX),NX
      GO TO 30
 2000 WRITE (6,2010) MAT
 2010 FORMAT (1H1,*  PROCESSING COMPLETE,  MAT = *I4)
```

```
      CALL GFPLT
      END
```

```
      SUBROUTINE STORXS
C     STORE TOTAL AND ELASTIC XSEC FOR THORIUM.
      LCM/XSECTT/XT(60000),YT(60000),NPTT
      LCM/XSECTE/XE(60000),YE(60000),NPEE
      DIMENSION A(7)
   10 READ (10,20) (A(I),I=1,7),MAT,MF,MT,NSEQ
   20 FORMAT (6A10,A6,I4,I2,I3,I5)
      IF (MAT.EQ.-1) GO TO 2000
      IF (MF.GT.3)  GO TO 300
      IF (MF.LT.3) GO TO 10
      IF (MT.EQ.1)  GO TO 100
      IF (MT.EQ.2)  GO TO 200
      GO TO 10
  100 READ (10,30) NPTT
      PRINT 2020,NPTT
      READ (10,20) (A(I),I=1,7)
      IF (NPTT.GT.60000) GO TO 130
      NPTTS=NPTT
  105 READ (10,40) (XT(I),YT(I),I=1,NPTTS)
      PRINT 40,(XT(I),YT(I),I=1,99)
      PRINT 2040,NPTTS,XT(NPTTS),YT(NPTTS)
  110 IF (NPTT.EQ.NPTTS)  GO TO 10
  120 IF (MT.EQ.0)  GO TO 10
      READ (10,20)(A(I),I=1,7),MAT,MF,MT
      GO TO 120
  130 NPTTS=60000
      GO TO 105
  200 READ (10,30) NPEE
      PRINT 2030,NPEE
      READ (10,20) (A(I),I=1,7)
      IF (NPEE.GT.60000) GO TO 230
      NPEES=NPEE
  205 READ (10,40)(XE(I),YE(I),I=1,NPEES)
      PRINT 40,(XE(I),YE(I),I=1,99)
      PRINT 2050,NPEES,XE(NPEES),YE(NPEES)
  210 IF (NPEE.EQ.NPEES)  GO TO 10
  220 IF (MT.EQ.0)  GO TO 10
      READ (10,20)(A(I),I=1,7),MAT,MF,MT
      GO TO 220
  230 NPEES=60000
      GO TO 205
  300 READ (10,20) (A(I),I=1,7),MAT,MF,MT,NSEQ
      IF (MAT.EQ.-1)  GO TO 2000
      IF (MAT.NE.0)  GO TO 300
      RETURN
   30 FORMAT (55X,I11)
   40 FORMAT (6E11.4)
 2000 WRITE(6,2010) MAT
 2010 FORMAT (1H1,*  SORRY TAPE IS OUT OF TEMPS, MAT=*I4)
 2020 FORMAT (1H1,10X,*XT,YT TABLE*10X,*NPTT=*I11)
 2030 FORMAT (1H0,10X,*XE,YE TABLE*10X,*NPEE=*I11)
 2040 FORMAT (1H ,*NPTTS=*I6,4X,*XT(NPTTS)=*,E11.4,4X,*YT(NPTTS)=*E11.4)
 2050 FORMAT (1H ,*NPEES=*I6,4X,*XE(NPEES)=*,E11.4,4X,*YE(NPEES)=*E11.4)
      RETURN
      END
```

```
      SUBROUTINE GRANSHL (E,FACT)
C     PURPOSE = TO CALCULATE SHIELDING FACTOR FOR TWO REGION PARTICLE.
C
      LCM/XSECTT/XT(60000),YT(60000),NPTT
      LCM/XSECTE/XE(60000),YE(60000),NPEE
      COMMON/CONS/RADC,RADP,VOLC,VOLP,SURC,SURP,PSIP(10),PSIC(10),EMAX,
     1    EMIN,MT
      COMMON/CALC/H1TAU,VOLF,Q,SEEP,SEEC,SIGPSP,SIGPSC,TAUTC,TAUXC
      COMMON/CON1/CONP(10),CONC(10),XSC(9),XSP(9),NMP,NMC,XP(9),XC(9)
      COMMON/PLOTS/ENG(5000),FAX(5000),GAMX(5000),NX,TITL(5),XLB(5),
     1    YLB(5)
C
C     CONDITIONAL RETURNS
C
      IF (E.GT.1.0E-10) GO TO 10
      E=0.
      FACT=0.
      RETURN
   10 CONTINUE
      IF (E.LT.EMAX) GO TO 20
      FACT=1.0
      RETURN
   20 CONTINUE
      QQ=Q
      IF (E.LT.EMIN) QQ=0.
C
C     FIND TOTAL AND ELASTIC CROSS SECTIONS CORRESPONDING TO ENERGY E.
C
      CALL LOCT1(E,ILK,ILOT)
      IHIT=ILOT+1
      CALL LOCT2(E,ILK,ILOE)
      IHIE=ILOE+1
      DXTL=XT(ILOT)
      DYTL=YT(ILOT)
      DXTH=XT(IHIT)
      DYTH=YT(IHIT)
      CALL TERP1 (DXTL,DYTL,DXTH,DYTH,E,ST,2,1)
      DXEL=XE(ILOE)
      DYEL=YE(ILOE)
      DXEH=XE(IHIE)
      DYEH=YE(IHIE)
      CALL TERP1 (DXEL,DYEL,DXEH,DYEH,E,SE,2,2)
C
      XP(1)=SE
      XSP(1)=ST
       SIGPSP1=0.
      SIGTP=0.
      SEENUM=0.
      DO 30 I=1,NMP
      SEENUM=PSIP(I)*CONP(I)*XP(I)+SEENUM
      SIGPSP1=CONP(I)*XP(I)+SIGPSP1
      SIGTP=CONP(I)*XSP(I)+SIGTP
   30 CONTINUE
      SEEP1=SEENUM/SIGPSP1
      TAUTP=4.0*VOLP*SIGTP/SURP
      TAUXP=TAUTP*(1.0-(1.0-SEEP1/SEEC)*SIGPSP1/SIGTP)
      X=3.0*TAUTP/4.0
      P0TAU=3.0/(8.0*X**3)*(2.0*X**2-1.0+(1.0+2.0*X)*EXP(-2.0*X))
      H0TAU=(1.0-P0TAU)/(TAUTP*P0TAU)-1.0
      W=1.0+H0TAU*H1TAU
      RHOQ=VOLP/VOLC*QQ
      UPPER=1.0+RHOQ*(1.0+TAUXC*W)
```

```
      UNDER=1.0+RHOQ*TAUXP*W                                            GRANS 63
      GAMMA=UPPER/UNDER                                                 GRANS 64
      FACT=VOLF/VOLC*(GAMMA/(1.0+VOLP/VOLC*GAMMA))                      GRANS 65
      IF (FACT.GT.0.999 ) GO TO 40                                      GRANS 66
      IF (MT.GT.1) GO TO 40                                             GRANS 67
      IF (E.LT.EMIN) GO TO 40                                           GRANS 68
      IF(NX.GT.1200) GO TO 40                                           GRANS 69
      NX=NX+1                                                           GRANS 70
      ENG(NX)=E                                                         GRANS 71
      FAX(NX)=FACT                                                      GRANS 72
      GAMX(NX)=GAMMA                                                    GRANS 73
   40 CONTINUE                                                          GRANS 74
   45 FORMAT (1H0,*  PLOTS GO ONLY TO *1PE12.5,* E.V.*)                 GRANS 75
      RETURN                                                            GRANS 76
      END                                                               GRANS 77
```

```
      SUBROUTINE CXFP(X,F,S,N)                                          CXFP   1
C***********************************************************************CXFP   2
C*    CONVERT X FOR PUNCHING                                           *CXFP   3
C*    X - FLOATING POINT NUMBER = F*10.0**N                            *CXFP   4
C*    F - 0.999995 LE F LT 9.999995                                    *CXFP   5
C*    S - SIGN (HOLLERITH + OR -) OF EXPONENT                           CXFP   6
C*    N - EXPONENT                                                     *CXFP   7
C***********************************************************************CXFP   8
      DATA SP/1H+/,SM/1H-/                                              CXFP   9
      IF(X.NE.0.0) GO TO 10                                             CXFP  10
      F=0.0                                                             CXFP  11
      S=SP                                                              CXFP  12
      N=0                                                               CXFP  13
      RETURN                                                            CXFP  14
   10 N=ALOG10(ABS(X))                                                  CXFP  15
      IF (ABS(X)-1.0) 40,20,20                                          CXFP  16
   20 F=X/10.0**N                                                       CXFP  17
      S=SP                                                              CXFP  18
      IF(ABS(F)-9.999995) 70,30,30                                      CXFP  19
   30 F=F/10.0                                                          CXFP  20
      N=N+1                                                             CXFP  21
      GO TO 70                                                          CXFP  22
   40 N=1-N                                                             CXFP  23
      F=X*10.0**N                                                       CXFP  24
      S=SM                                                              CXFP  25
      IF(ABS(F)-9.999995) 70,50,50                                      CXFP  26
   50 F=F/10.0                                                          CXFP  27
      N=N-1                                                             CXFP  28
      IF(N) 60,60,70                                                    CXFP  29
   60 S=SP                                                              CXFP  30
   70 CONTINUE                                                          CXFP  31
      RETURN                                                            CXFP  32
      END                                                               CXFP  33
```

```
      SUBROUTINE TERP1 (X1,Y1,X2,Y2,X,Y,I,NERR)
C     =====INTERPOLATE ONE PT.=====
C     (X1,Y1) AND (X2,Y2) ARE END PTS. OF THE LINE
C     (X,Y) IS INTERPOLATED POINT
C     I=INTERPOLATION CODE
C     NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE
C             INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.
C     ERROR STOPS - 301 (X1=X2,DISCONTINUITY)
C                   302 (INTERPOLATION CODE IS OUT OF RANGE)
C                   303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT.)
    5 XA=X1
      YA=Y1
      XB=X2
      YB=Y2
      XP=X
      II=I
      IF ((XB-XA).GT.1.E-10) GO TO 7
      IF (X.EQ.XA) Y=YA
      PRINT 6,XA,YA,XB,YB,X,Y,I,NERR
    6 FORMAT (1H0,*  ERROR STOP 301 *1P6E12.5,2I3)
      RETURN
    7 CONTINUE
      IF (II) 10,10,15
   10 CALL ERROR (302)
   15 IF (II-5) 20,20,10
   20 GO TO (25,30,35,60,75), II
   25 YP=YA
      IF (XP.EQ.XB)  YP=YB
      GO TO 105
   30 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)
      GO TO 105
   35 IF (XA) 30,30,40
   40 IF (XB) 30,30,45
   45 IF (XP) 50,50,55
   50 CALL ERROR (303)
   55 YP=YA+ALOG(XP/XA)*(YB-YA)/ALOG(XB/XA)
      GO TO 105
   60 IF (YA) 30,30,65
   65 IF (YB) 30,30,70
   70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))
      GO TO 105
   75 IF (YA) 35,35,80
   80 IF (YB) 35,35,85
   85 IF (XA) 70,70,90
   90 IF (XB) 70,70,95
   95 IF (XP) 50,50,100
  100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))
  105 Y=YP
      RETURN
      END
```

```
      SUBROUTINE TERP1 (X1,Y1,X2,Y2,X,Y,I,NERR)
C     =====INTERPOLATE ONE PT.====*
C     (X1,Y1) AND (X2,Y2) ARE END PTS. OF THE LINE
C     (X,Y) IS INTERPOLATED POINT
C     I=INTERPOLATION CODE
C     NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE
C             INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.
C     ERROR STOPS - 301 (X1=X2,DISCONTINUITY)
C                   302 (INTERPOLATION CODE IS OUT OF RANGE)
C                   303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT.)
    5 XA=X1
      YA=Y1
      XB=X2
      YB=Y2
      XP=X
      II=I
      IF ((XB-XA).GT.1.E-10) GO TO 7
      IF (X.EQ.XA) Y=YA
      PRINT 6,XA,YA,XB,YB,X,Y,I,NERR
    6 FORMAT (1H0,*  ERROR STOP 301 *1P6E12.5,2I3)
      RETURN
    7 CONTINUE
      IF (II) 10,10,15
   10 CALL ERROR (302)
   15 IF (II-5) 20,20,10
   20 GO TO (25,30,35,60,75), II
   25 YP=YA
      IF (XP.EQ.XB)  YP=YB
      GO TO 105
   30 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)
      GO TO 105
   35 IF (XA) 30,30,40
   40 IF (XB) 30,30,45
   45 IF (XP) 50,50,55
   50 CALL ERROR (303)
   55 YP=YA+ALOG(XP/XA)*(YB-YA)/ALOG(XB/XA)
      GO TO 105
   60 IF (YA) 30,30,65
   65 IF (YB) 30,30,70
   70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))
      GO TO 105
   75 IF (YA) 35,35,80
   80 IF (YB) 35,35,85
   85 IF (XA) 70,70,90
   90 IF (XB) 70,70,95
   95 IF (XP) 50,50,100
  100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))
  105 Y=YP
      RETURN
      END
```

```
      SUBROUTINE ERROR (N)
      IOS=9
    5 PRINT 10,N
      WRITE(99,10)
   10 FORMAT (11H ERROR STOP,I6)
      END
```

```
      SUBROUTINE LOCT1(X,ILO,LOCT)
C     BINARY SEARCH ROUTINE WRITTEN BY P. SORAN.  MODIFIED 10-30-73
C         TO GIVE RESULTS IDENTICAL TO EARLER LOCT ROUTINE.
C         THAT IS,  FIND X SUCH THAT A(LOCT+1).GT.X.GE.A(LOCT), EXCEPT W
C         X IS EQUAL TO A(N).  IN THAT CASE, LOCT IS SET TO (N-1).
C         WHEN X IS NOT BINNABLE, THAT IS WHEN X IS OUTSIDE THE RANGE OF
C         A-VALUES OR IF A CONTAINS ONLY A SINGLE POINT, THE VALUE LOCT=
C         IS RETURNED.
      LCM/XSECTT/A(60000),YT(60000),N
      IF(N.EQ.1) GO TO 3001
      IF(X.LT.A(1)) GO TO 3001
      IF(X.GT.A(N)) GO TO 3001
      IF((A(N-1).EQ.A(N)).AND.(X.EQ.A(N))) GO TO 3001
      LOCT=1
      IF(A(1).EQ.X) RETURN
      ILO=1
      ISRCH=N
 1000 IF(ISRCH.LE.ILO+1)GO TO 3000
      I=(ISRCH+ILO)/2
      IF(A(I).LT.X) GO TO 2000
      ISRCH=I
      GO TO 1000
 2000 ILO=I
      GO TO 1000
C     X HAS BEEN BINNED.  CONVERT FROM ISRCH TO LOCT HERE.
 3000 IF(X.NE.A(ISRCH)) LOCT=ISRCH-1
      IF(X.EQ.A(ISRCH)) LOCT=ISRCH
      IF(X.EQ.A(N)) LOCT=N-1
      RETURN
 3001 WRITE (99,10) LOCT
   10 FORMAT (1H ,I6)
      RETURN
      END
```

```
      SUBROUTINE LOCT2(X,ILO,LOCT)
C     BINARY SEARCH ROUTINE WRITTEN BY P. SORAN.  MODIFIED 10-30-73
C         TO GIVE RESULTS IDENTICAL TO EARLER LOCT ROUTINE.
C         THAT IS,  FIND X SUCH THAT A(LOCT+1).GT.X.GE.A(LOCT), EXCEPT W
C         X IS EQUAL TO A(N).  IN THAT CASE, LOCT IS SET TO (N-1).
C         WHEN X IS NOT BINNABLE, THAT IS WHEN X IS OUTSIDE THE RANGE OF
C         A-VALUES OR IF A CONTAINS ONLY A SINGLE POINT, THE VALUE LOCT=
C         IS RETURNED.
      LCM/XSECTE/A(60000),XE(60000),N
      IF(N.EQ.1) GO TO 3001
      IF(X.LT.A(1)) GO TO 3001
      IF(X.GT.A(N)) GO TO 3001
      IF((A(N-1).EQ.A(N)).AND.(X.EQ.A(N))) GO TO 3001
      LOCT=1
      IF(A(1).EQ.X) RETURN
      ILO=1
      ISRCH=N
 1000 IF(ISRCH.LE.ILO+1)GO TO 3000
      I=(ISRCH+ILO)/2
      IF(A(I).LT.X) GO TO 2000
      ISRCH=I
      GO TO 1000
 2000 ILO=I
      GO TO 1000
C     X HAS BEEN BINNED.  CONVERT FROM ISRCH TO LOCT HERE.
 3000 IF(X.NE.A(ISRCH)) LOCT=ISRCH-1
      IF(X.EQ.A(ISRCH)) LOCT=ISRCH
      IF(X.EQ.A(N)) LOCT=N-1
      RETURN
```

```
3001 WRITE (99,10) LOCT                                                  LOCT2 30
  10 FORMAT (1H ,I6)                                                     LOCT2 31
     RETURN                                                              LOCT2 32
     END                                                                 LOCT2 33
```

```
      SUBROUTINE GFPLT                                                   GFPLT  1
C     SUBROUTINE TO PLOT GAMMA AND FACT FOR REPORT.                      GFPLT  2
      COMMON/CONS/RADC,RADP,VOLC,VOLP,SURC,SURP,PSIP(10),PSIC(10),EMAX,  GFPLT  3
     1     EMIN,MT                                                       GFPLT  4
      COMMON/PLOTS/E(4,1250),F(4,1250),G(4,1250),NX,TITL(5),XLBS(5),     GFPLT  5
     1     YLB(5)                                                        GFPLT  6
      DIMENSION XLAB(5),YLAB(5),ZLAB(5),NPT(5),X(5000),                  GFPLT  7
     1     Y(5000),Z(5000)                                               GFPLT  8
      REWIND 9                                                           GFPLT  9
      NT=4                                                               GFPLT 10
      DO 10 N=1,NT                                                       GFPLT 11
      READ (9) NX,(E(N,I),F(N,I),G(N,I),I=1,NX),NPT(N)                   GFPLT 12
   10 CONTINUE                                                           GFPLT 13
      NN1=1                                                              GFPLT 14
      X(1)=EMIN                                                          GFPLT 15
      Y(1)=1.                                                            GFPLT 16
      Z(1)=1.                                                            GFPLT 17
      DO 30 N=1,NT                                                       GFPLT 18
      JP=NPT(N)                                                          GFPLT 19
      DO 20 J=1,JP                                                       GFPLT 20
      NN1=NN1+1                                                          GFPLT 21
      X(NN1)=E(N,J)                                                      GFPLT 22
      Y(NN1)=F(N,J)                                                      GFPLT 23
      Z(NN1)=G(N,J)                                                      GFPLT 24
   20 CONTINUE                                                           GFPLT 25
      NN1=NN1+1                                                          GFPLT 26
      X(NN1)=EMAX                                                        GFPLT 27
      Y(NN1)=1.                                                          GFPLT 28
      Z(NN1)=1.                                                          GFPLT 29
      NN1=NN1+1                                                          GFPLT 30
      X(NN1)=EMAX                                                        GFPLT 31
      Y(NN1)=0.                                                          GFPLT 32
      Z(NN1)=0.                                                          GFPLT 33
      NN1=NN1+1                                                          GFPLT 34
      X(NN1)=EMIN                                                        GFPLT 35
      Y(NN1)=0.                                                          GFPLT 36
      Z(NN1)=0.                                                          GFPLT 37
      NN1=NN1+1                                                          GFPLT 38
      X(NN1)=EMIN                                                        GFPLT 39
      Y(NN1)=1.                                                          GFPLT 40
      Z(NN1)=1.                                                          GFPLT 41
   30 CONTINUE                                                           GFPLT 42
      TITL(1)=10H FACT FOR                                               GFPLT 43
      TITL(2)=10H0-300-950.                                              GFPLT 44
      TITL(3)=10H3000 DEG K                                              GFPLT 45
      XLAB(1)=10HENERGY IN                                               GFPLT 46
      XLAB(2)=10HE.V. UNITS                                              GFPLT 47
      YLAB(1)=10H FACT OF E                                              GFPLT 48
      CALL PLOJB(X,Y,NN1,1,0,0,0,0,1.0,1.0,TITL,30,XLAB,20,YLAB,10)      GFPLT 49
      TITL(1)=10HGAMMA FOR                                               GFPLT 50
      YLAB(1)=10HGAMMA OF E                                              GFPLT 51
      CALL PLOJB (X,Z,NN1,1,0,0,0,0,1.0,1.0,TITL,30,XLAB,20,YLAB,10)     GFPLT 52
      RETURN                                                             GFPLT 53
      END                                                                GFPLT 54
```

APPENDIX B

ETOGLEN

A code to prepare absorber cross-section input for the GLEN code - ENDF/B to GLEN.

In addition to microscopic cross-section data for the moderating materials, which are supplied by the TOR code, the GLEN thermal multigroup-averaging code also requires pointwise data for elastic scattering, fission, and absorption cross sections for the absorbing materials in the reactor model being calculated. These need be only supplied in the thermal energy range, e.g., up to 2.38 eV for the HTGR but, because of storage limitations in present versions of GLEN, the data must be restricted to fewer than 88 energy cross-section pairs for each reaction. Consequently, some care must be taken in choosing a fine energy grid for a particular problem that adequately reproduces the shapes of the cross sections for all materials present and preserves the resonance integral of the principal constituents.

The purpose of the ETOGLEN code is : a) retrieve pointwise cross-section data from a pointwise ENDF/B file (PENDF), created by the MINX code; b) assist the user in choosing a fine energy grid for a problem by allowing flexible grid input, by providing comparison plots of the selected grid vs the ENDF/B points, and by calculating weighted resonance integrals over specified intervals for data on both the selected grid and the original ENDF/B grid; and c) output absorber cross-section data in the format required by the GLEN code.

The selected grid need not be a subset of the original ENDF/B grid, as the code will interpolate on any given mesh. If a representative spectrum is taken for the weighting function for the resonance integral calculations and broad-group boundaries are taken for the calculational intervals, one obtains the error in the broad-group cross sections incurred by grid selection.

The code calculates an energy grid on the basis of a set of incremental values of lethargy (or velocity increments in another version of ETOGLEN) specified for several energy intervals. To these are added additional points, input by the user, such as peaks and valleys of important resonances and the cut points for the broad group cross sections which are required by the GLEN code.

Normally cross sections for several temperatures are given on a PENDF. ETOGLEN will automatically process the cross sections for all temperatures given. These temperatures are initially chosen to span the range of interest and at intervals frequent enough to represent cross sections at a number of neighboring temperatures. For the HTGR problem, for example, cross sections for 12 temperatures were required over a range from 300 to 3000 K. PENDF cross sections were generated at 0, 300, 950, and 3000 K, and Table B-I shows which of the PENDF values were used for each of the 12 temperatures. Table B-II describes the input specifications for ETOGLEN, and a listing of the code appears at the end of this appendix.

Sample results from ETOGLEN are shown in Table B-III and Figs. B-1 and B-2. These are for ^{233}U, MAT-1260; for this problem, an 86-point energy grid was previously optimized for the thermal resonances of ^{235}U. The graphical output from ETOGLEN (Figs. B-1 and B-2) demonstrates the accuracy with which the resonance structure is reproduced with the coarser grid, and the weighted averaging done in the code indicates the amount of error incurred in the multigroup cross sections by using the coarser grid (Table B-III). Also note in this table the small effect of temperature on average cross sections for this isotope and this energy group structure.

TABLE B-I

PENDF TEMPERATURES USED FOR CROSS SECTIONS

FOR END-OF-EQUILIBRIUM CYCLE (EOEC) HTGR CASES

HTGR-EOEC Case No.	Temperature (K)	PENDF Temperature Used for Thermal Cross Sections
1	300	300
2	500	300
3	600	300
4	800	950
5	1000	950
6	1200	950
7	1500	950
8	1700	950
9	2000	3000
10	2300	3000
11	2600	3000
12	3000	3000

TABLE B-II

INPUT SPECIFICATIONS FOR ETOGLEN

Card No.	Format	Variable	Comment
1	I11	NUMBIN	Number of energy mesh intervals over which equal lethargy intervals are specified.
2	6E11.4	BMIN(N), BMAX(N), DELU(N)	The minimum energy, the maximum energy, and the lethargy increment for each of the NUMBIN intervals.
3	I11	NPD	Number of additional energy points to be added.
4	6E11.4	ED(I)	NPD values of additional energies.
5	I11	NFGP	Number of cut points of intervals over which resonance integrals are to be computed (normally number of few groups, i.e., broad groups).
6	6E11.4	EC(I)	NFGP values of cut point energies. Note - if EC(I) are broad-group boundaries, they must also be specified in the ED list.
7	I11	NW	Number of energy-flux pairs given for the weighting function.
8	6E11.4	EWI(N), WI(N)	NW values for energy-flux pairs of specified weighting function.

NDF is name of the file containing pointwise data at several temperatures for the absorber being processed.

TABLE B-III

WEIGHTED AVERAGE FISSION
CROSS SECTIONS FOR ^{233}U

T = 300 K

Upper Energy Boundary (eV)	Calculated Using Original PENDF Data	Calculated Using Data on Reduced Mesh	% Diff.
0.04	566.6	567.5	0.2
0.10	324.9	328.5	1.1
0.414	201.6	205.3	1.8
2.38	232.9	234.6	0.7
	T = 3000 K		
0.04	566.7	567.5	0.2
0.10	324.9	328.6	1.1
0.414	202.1	205.8	1.8
2.38	232.2	233.4	0.5

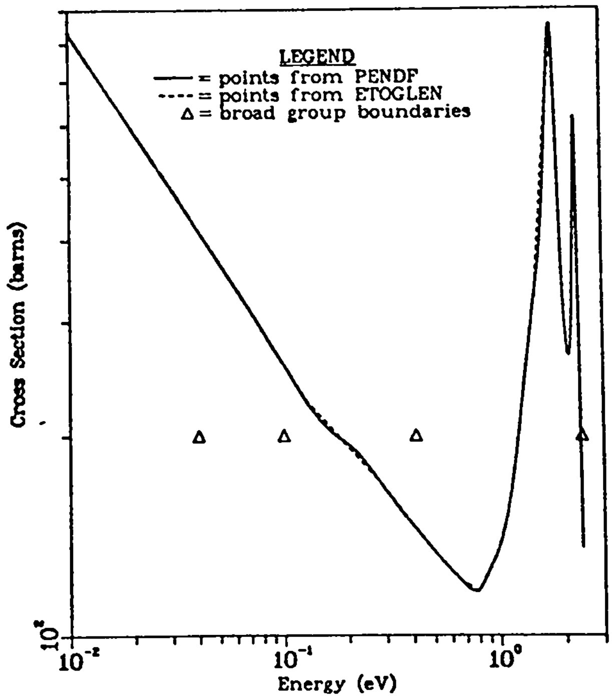


Fig. B-1.
^{233}U fission cross section at 300 K from 0.01 to 2.38 eV.

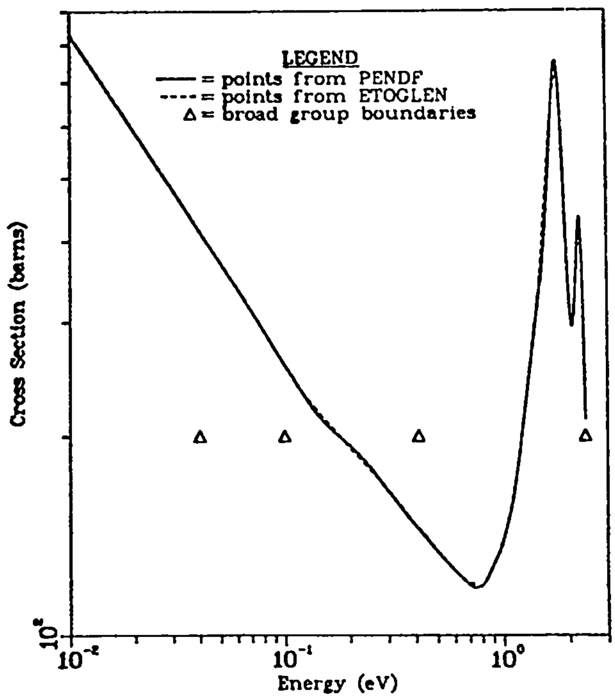


Fig. B-2.
^{233}U fission cross section at 3000 K from 0.01 to 2.38 eV.

```
      PROGRAM ETOGLEN(INP,OUT,PUN,FILM,FSET11,FSET12,FSET5=INP,
     1                 FSET6=OUT)
C     PROGRAM TO GET GLEN INPUT CROSS SECTIONS FROM PENDF TAPE,I.E.
C     ENDF/B TO GLEN
C     -       -- ----
C
      DIMENSION XFISS(200),XCAP(200)
      DIMENSION EC(200),ED(200),EM(200),INT(10),NPT(10),E(2000),S(2000),
     1   HOL(10),SM(200),TITL(10),XLAB(10),YLAB(10),ER(2000),EMR(200),
     2    SC(200),SR(2000),WI(200),W(2000),WM(200),EWI(200),NECT(10),
     3    NEMCT(10),SBD(10),SMBD(10),BMIN(12),BMAX(12),DELU(12)

      DIMENSION XP(500),YP(500)

C     E. IS BASIC CALCULATED MESH,ED ARE ADDITIONAL POINTS TO BE ADDED,
C     E.G. BROAD GROUP MESH BREAK POINTS IN GLEN,RESONANCE PEAKS,
C     VALLEYS,ETC.,AND EM IS COMBINED MESH.
C     FIRST READ BMIN,BMAX,DELU FOR EACH REGION (UP TO 8) FOR
C     CALCULATION OF BASIC E MESH -- DESCENDING ORDER.
C     EMIN IS LOWEST ENERGY BOUND. EMAX IS HIGHEST
C     EC ARE BROAD GROUP BREAK POINTS TO BE ADDED TO PENDF MESH
C      FOR INTEGRAL CHECK
C     EWI,WI ARE ENERGY,WEIGHT FUNCTION PAIRS FOR WEIGHTING
C     IF INTEGRAL CHECK
C
      READ (5,30) NUMBIN
      DO 5 N=1,NUMBIN
      READ (5,10) BMIN(N),BMAX(N),DELU(N)
    5 CONTINUE
   10 FORMAT (6E11.4)
      IF (NUMBIN.EQ.1)  GO TO 7
      NUMED=NUMBIN-1
      DO 6 N=1,NUMED
      IF (BMIN(N).NE.BMAX(N+1))  BMIN(N)=BMAX(N+1)
    6 CONTINUE
    7 NPC=1
      WRITE (6,12) NUMBIN
   12 FORMAT (1H1,17X,I2,20H INPUT ENERGY GROUPS//2X,9HGROUP NO.,
     1     2X,14HGROUP MAX (EV),2X,14HGROUP MIN (EV),2X,
     2     14HLETHARGY WIDTH)
      DO 8 N=1,NUMBIN
      WRITE (6,13) N,BMAX(N),BMIN(N),DELU(N)
    8 CONTINUE
   13 FORMAT (4X,I3,8X,E11.5,5X,E11.5,5X,E11.5)
      DO 20 N=1,NUMBIN
      U=0.
      E(NPC)=BMAX(N)
   11 U=U+DELU(N)
      NPC=NPC+1
      E(NPC)=BMAX(N)/EXP(U)
      IF (E(NPC).LE.BMIN(N))  GO TO 20
      GO TO 11
   20 CONTINUE
      E(NPC)=BMIN(N)
      EMIN=BMIN(NUMBIN)
      EMAX=BMAX(1)
      DO 21 N=1,NPC
      SR(N)=N
   21 CONTINUE
C
C     READ ED MESH FROM CARDS.
C
```

```
      READ (5,30) NPD
   30 FORMAT (6I11)
      READ (5,10) (ED(I),I=1,NPD)
C
C     READ BROAD GROUP ENERGIES.  READ FROM HI TO LO.
C
      READ (5,30) NFGP
      READ (5,10) (EC(I),I=1,NFGP)
C
C  READ IN WEIGHT FCNS.  READ IN E-LO TO E-HI.  MUST BE LOG-LOG INTERP.
C
      READ (5,30) NW
      READ (5,10) (EWI(N),WI(N),N=1,NW)
      TITL(1)=10HTHERMAL WE
      TITL(2)=10HIGHT FUNCT
      TITL(3)=10HION (GLEN)
      NWPLT=-NW
      XLAB(1)=10HENERGY(EV)
      YLAB(1)=10HTHERM FLUX
      CALL PLOJB (EWI,WI,NWPLT,-1,0,0,0.0,1.0,1.0,TITL,30,XLAB,10,YLAB,
     1   10)
C
C
C     COMBINE E  AND ED TO FORM EM.
C
      CALL UNION (E,ED,NPC,NPD)
C
C     REORDER E TO GET EM
C
      DO 25 N=1,NPC
      NN1=NPC-N+1
      EM(N)=E(NN1)
   25 CONTINUE
      JNPC=NPC-1
      NNTST=0
      DO 35 JI=1,JNPC
      DLTST=(EM(JI+1)-EM(JI))/EM(JI)*100.
      IF (DLTST.GT.1.0) GO TO 35
      PRINT 36,JI,EM(JI),EM(JI+1),DLTST
      NNTST=NNTST+1
   35 CONTINUE
   36 FORMAT (1H0,* DUPLICATE ENERGIES AT *I4.*,*1PE12.5,* AND *1PE12.5,
     1     *  PCT DIFF = *F6.3)
      IF (NNTST.EQ.0) PRINT 37
   37 FORMAT (1H0,*  THERE ARE NO ENERGY PAIRS WITHIN ONE PERLENT.*)
      NPM=NPC
   40 FORMAT (1H0,*  I = *I3,*  E = *1PE12.5)
C
C     PUNCH ENERGY MESH FOR GLEN.
C
      DO 45 N=1,NPM
      NN1=NPM-N+1
      E(N)=EM(NN1)
   45 CONTINUE
      TITL(5)=10HENERGY MES
      TITL(6)=10HH FOR GLEN
      PUNCH 210,(TITL(I),I=5,6)
      PUNCH 150,(E(N),N=1,NPM)
C
C     THERE ARE SEVERAL TEMPERATURES ON TAPE.  READ NOTEMP=NO. OF TEMPS.
C
      READ (5,30) NOTEMP,MAT1
      DO 1000 NNT=1,NOTEMP
```

```
C
C     FOR EACH TEMP,GET XSEC FOR MT=2,MT=18,MT=102.
C
      DO 900 NMT=1,3
      MF1=3
      MT1=2
      IF (NMT.EQ.2) MT1=18
      IF (NMT.EQ.3) MT1=102
      IF (NMT.EQ.3.AND.MAT1.EQ.1155) MT1=107
   50 READ (11,60) (HOL(I),I=1,7),MAT,MF,MT,NSEQ
   60 FORMAT (6A10,A6,I4,I2,I3,I5)
      IF (MAT.LT.0)  GO TO 2000
      IF (MAT.LT.MAT1)  GO TO 50
      IF (MAT.GT.MAT1)  GO TO 2000
      IF (MF.NE.3) GO TO 50
      IF (MT.NE.MT1) GO TO 50
      WRITE (6,2020) MAT,MAT1,MF,MT,MT1,NMT
 2020 FORMAT (1H ,*MAT=*I4,3X,*MAT1=*I4,3X,*MF=*I2,3X,*MT=*I3,3X,*MT1=*,
     1     I3,3X,*NMT=*I2)
      READ (11,70) C1,C2,N1,N2,NR,NP
   70 FORMAT(1P2E11.4,4I11)
      TEMDS=C1
      READ (11,30) (NPT(I),INT(I),I=1,NR)
C
C     ASSUME THERMAL RANGE IS WITHIN FIRST 2000 PTS ON TAPE.
C
      NPTH=NP
      IF (NP.GT.2000) NPTH=2000
      READ (11,10) (E(I),S(I),I=1,NPTH)
   80 READ (11,60) (HOL(I),I=1,7),MAT,MF,MT,NSEQ
      IF (MT.NE.0) GO TO 80
      IF (E(NPTH).GT.EMAX) GO TO 82
      WRITE (6,81) NPTH,E(NPTH)
   81 FORMAT (1H1,*  EMAX NOT WITHIN *I4,* PTS. LAST ENERGY = *1PE12.5)
      STOP
   82 CONTINUE
C
C     GET XSEC,SM,CORRESPONDING TO EM.
C
      DO 100 I=1,NPM
      ILO=LOCT(E,EM(I),NPTH)
      IF (ILO.EQ.-1) CALL ERROR(100)
      IHI=ILO+1
      DO 85 J=1,NR
      IF (IHI.LE.NPT(J)) GO TO 90
   85 CONTINUE
      CALL ERROR (200)
   90 CALL TERP1 (E(ILO),S(ILO),E(IHI),S(IHI),EM(I),CSEC,INT(J))
      SM(I)=CSEC
  100 CONTINUE
  110 FORMAT (1H0,*  I=*I6,*  EM = *1PE11.4,*  SM = *1PE11.4)
C
C     CHECK INTEGRALS AND MAKE COMPARISON PLOTS.
C
C     CUT OFF MESH POINTS ABOVE EMAX
      KTHRM=0
      DO 120 N=1,NPTH
      IF (E(N).GT.EMAX) GO TO 130
      KTHRM=KTHRM+1
  120 CONTINUE
  130 CONTINUE
      TITL(1)=10HETOGLEN VS
      TITL(2)=10H PENDF PTS
```

```
      TITL(3)=10H. ELASTIC                                              ETOGL189
      TITL(4)=10HCROSS SECT                                             ETOGL190
      IF (NMT.EQ.2) TITL(3)=10H. FISSION                                ETOGL191
      IF (NMT.EQ.3) TITL(3)=10H. CAPTURE                                ETOGL192
      NPTR=KTHRM                                                        ETOGL193
C                                                                       ETOGL194
C     REVERSE E-MESH AND ADD BROAD GROUP CUT POINTS                     ETOGL195
C                                                                       ETOGL196
      DO 300 N=1,NPTR                                                   ETOGL197
      N1=NPTR-N+1                                                       ETOGL198
      ER(N)=E(N1)                                                       ETOGL199
  300 CONTINUE                                                          ETOGL200
      CALL UNION(ER,EC,NPTR,NFGP)                                       ETOGL201
      DO 320 N=1,NPTR                                                   ETOGL202
      ILO =LOCT (E,ER(N),NPTR)                                          ETOGL203
      IF (ILO.EQ.-1) CALL ERROR(200)                                    ETOGL204
      IHI=ILO+1                                                         ETOGL205
      CALL TERPI (E(ILO),S(ILO),E(IHI),S(IHI),ER(N),CSEC,2)             ETOGL206
      SR(N)=CSEC                                                        ETOGL207
  320 CONTINUE                                                          ETOGL208
C                                                                       ETOGL209
C     CUT OFF MESH POINTS BELOW EMIN                                    ETOGL210
C                                                                       ETOGL211
  321 DO 326 N=1,NPTR                                                   ETOGL212
      IF (ER(N).LT.EMIN) GO TO 327                                      ETOGL213
      NSTOR=N                                                           ETOGL214
  326 CONTINUE                                                          ETOGL215
  327 CONTINUE                                                          ETOGL216
      IF (ER(NSTOR).EQ.EMIN)  N=NSTOR+1                                 ETOGL217
      NPTR=N-1                                                          ETOGL218
      KTHRM=NPTR                                                        ETOGL219
C                                                                       ETOGL220
C     REORDER LOW TO HIGH                                               ETOGL221
C                                                                       ETOGL222
      DO 330 N=1,NPTR                                                   ETOGL223
      N1=NPTR-N+1                                                       ETOGL224
      E(N)=ER(N1)                                                       ETOGL225
      S(N)=SR(N1)                                                       ETOGL226
  330 CONTINUE                                                          ETOGL227
C     PUT WT FCN ON E AND EM MESHES.                                    ETOGL228
C                                                                       ETOGL229
      DO 340 N=1,NPTR                                                   ETOGL230
      ILO=LOCT(EWI,E(N),NW)                                             ETOGL231
      IF (ILO.EQ.-1) CALL ERROR (300)                                   ETOGL232
      IHI=ILO+1                                                         ETOGL233
      CALL TERPI (EWI(ILO),WI(ILO),EWI(IHI),WI(IHI),E(N),WSS,5)         ETOGL234
      W(N)=WSS                                                          ETOGL235
  340 CONTINUE                                                          ETOGL236
      DO 350 N=1,NPM                                                    ETOGL237
      ILO=LOCT(EWI,EM(N),NW)                                            ETOGL238
      IF (ILO.EQ.-1) CALL ERROR(400)                                    ETOGL239
      IHI=ILO+1                                                         ETOGL240
      CALL TERPI (EWI(ILO),WI(ILO),EWI(IHI),WI(IHI),EM(N),WSS,5)        ETOGL241
      WM(N)=WSS                                                         ETOGL242
  350 CONTINUE                                                          ETOGL243
C                                                                       ETOGL244
C     REVERSE EC MESH                                                   ETOGL245
C                                                                       ETOGL246
      DO 345 K=1,NFGP                                                   ETOGL247
      SC(K)=EC(K)                                                       ETOGL248
  345 CONTINUE                                                          ETOGL249
      DO 346 K=1,NFGP                                                   ETOGL250
      K1=NFGP-K+1                                                       ETOGL251
```

```
      EC(K)=SC(K1)                                                      ETOGL252
  346 CONTINUE                                                          ETOGL253
C                                                                       ETOGL254
C     FIND BROAD GROUP CUT POINTS IN E AND EM MESHES.                   ETOGL255
C                                                                       ETOGL256
      NN1=0                                                             ETOGL257
      DO 370 I=1,NFGP                                                   ETOGL258
      DO 360 N=1,NPM                                                    ETOGL259
      IF ((EM(N)-EC(I)).NE.0.0)    GO TO 360                            ETOGL260
      NN1=NN1+1                                                         ETOGL261
      NEMCT(NN1)=N                                                      ETOGL262
      GO TO 370                                                         ETOGL263
  360 CONTINUE                                                          ETOGL264
  370 CONTINUE                                                          ETOGL265
      WRITE (6,380) (NEMCT(N),N=1,NFGP)                                 ETOGL266
  380 FORMAT (1H0,*  BROAD GROUP CUT POINTS *//5X,10I6)                 ETOGL267
      NN1=0                                                             ETOGL268
      DO 400 I=1,NFGP                                                   ETOGL269
      DO 390 N=1,NPTR                                                   ETOGL270
      IF (E(N).NE.EC(I)) GO TO 390                                      ETOGL271
      NN1=NN1+1                                                         ETOGL272
      NECT(NN1)=N                                                       ETOGL273
      GO TO 400                                                         ETOGL274
  390 CONTINUE                                                          ETOGL275
  400 CONTINUE                                                          ETOGL276
      WRITE (6,380) (NECT(I),I=1,NFGP)                                  ETOGL277
      WRITE (6,405) (TITL(I),I=3,4),NNT                                 ETOGL278
  405 FORMAT (1H1,*  GLEN POINTS FOR *2A10,*  .  TEMP NUMBER *,I2,      ETOGL279
     1     //*  PT. NO.*4X,*ENERGY*8X,                                  ETOGL280
     2     *CROSS SECTION*4X,*WEIGHT FUNCTION*)                         ETOGL281
  407 FORMAT (1H1,* PENDF POINTS FOR *2A10,*  .  TEMP NUMBER  *,I2,     ETOGL282
     1     //*  PT. NO.*4X,*ENERGY*8X,                                  ETOGL283
     2     *CROSS SECTION*4X,*WEIGHT FUNCTION*)                         ETOGL284
      WRITE (6,406) (N,EM(N),SM(N),WM(N),N=1,NPM)                       ETOGL285
      WRITE (6,407) (TITL(I),I=3,4),NNT                                 ETOGL286
  406 FORMAT (I6,1P3E18.5)                                              ETOGL287
      WRITE (6,406) (N,E(N),S(N),W(N),N=1,NPTR)                         ETOGL288
C                                                                       ETOGL289
C     GET BROAD GROUP XSEC FOR BOTH PENDF AND GLEN DATA.                ETOGL290
C                                                                       ETOGL291
      NBG=NFGP-1                                                        ETOGL292
      DO 430 N=1,NBG                                                    ETOGL293
      TOP=0.                                                            ETOGL294
      DEM=0.                                                            ETOGL295
      NE1=NECT(N)                                                       ETOGL296
      NE2=NECT(N+1)-1                                                   ETOGL297
      NEM1=NEMCT(N)                                                     ETOGL298
      NEM2=NEMCT(N+1)-1                                                 ETOGL299
      DO 410 J=NE1,NE2                                                  ETOGL300
      TOP=TOP+(E(J+1)-E(J))*(W(J+1)*S(J+1)+W(J)*S(J))/2.                ETOGL301
      DEM=DEM+((E(J+1)-E(J))*(W(J+1)+W(J))/2.0)                         ETOGL302
  410 CONTINUE                                                          ETOGL303
      SBD(N)=TOP/DEM                                                    ETOGL304
      TOP=0.                                                            ETOGL305
      DEM=0.                                                            ETOGL306
      DO 420 J=NEM1,NEM2                                                ETOGL307
      TOP=TOP+(EM(J+1)-EM(J))*(WM(J+1)*SM(J+1)+WM(J)*SM(J))/2.          ETOGL308
      DEM=DEM+((EM(J+1)-EM(J))*(WM(J+1)+WM(J))/2.0)                     ETOGL309
  420 CONTINUE                                                          ETOGL310
      SMBD(N)=TOP/DEM                                                   ETOGL311
  430 CONTINUE                                                          ETOGL312
      WRITE (6,440) (TITL(I),I=3,4),NNT                                 ETOGL313
  440 FORMAT (1H1,20X,2A10,*  .  TEMP NUMBER  *I2,                      ETOGL314
```

```
    1    //*  BROAD GROUP ENERGY XSEC FROM PENDF XSEC FROM GLEN DATA*)  ETOGL315
      WRITE (6,450) (EC(I),SBD(I),SMBD(I),I=1,NBG)                 ETOGL316
  450 FORMAT (1P3E18.5)                                             ETOGL317
      NI=1                                                          ETOGL318
      XLAB(1)=10H ENERGY IN                                         ETOGL319
      XLAB(2)=10H EV. UNITS                                         ETOGL320
      YLAB(1)=10H CROSS SEC                                         ETOGL321
      YLAB(2)=10HTION (BNS)                                         ETOGL322
      NPLOT=0                                                       ETOGL323
      DO 455 N=1,KTHRM                                              ETOGL324
      IF (E(N).LT.0.01) GO TO 455                                   ETOGL325
      NPLOT=NPLOT+1                                                 ETOGL326
      XP(NPLOT)=E(N)                                                ETOGL327
      YP(NPLOT)=S(N)                                                ETOGL328
  455 CONTINUE                                                      ETOGL329
      NPLOT=-NPLOT                                                  ETOGL330
      NI=-1                                                         ETOGL331
      CALL PLOTM (XP,YP,NPLOT,NI,0,39,0.,1.,1.,TITL,40,XLAB,20,     ETOGL332
    1     YLAB,20)                                                  ETOGL333
      NPLT=NPM                                                      ETOGL334
      NPLOT=0                                                       ETOGL335
      DO 460 N=1,NPLT                                               ETOGL336
      IF (EM(N).LT.0.01) GO TO 460                                  ETOGL337
      NPLOT=NPLOT+1                                                 ETOGL338
      XP(NPLOT)=EM(N)                                               ETOGL339
      YP(NPLOT)=SM(N)                                               ETOGL340
  460 CONTINUE                                                      ETOGL341
      NPLOT=-NPLOT                                                  ETOGL342
      NI=-1                                                         ETOGL343
      CALL PLOTM (XP,YP,NPLOT,NI,0,-37,0.,1.,1.,TITL,40,XLAB,20,    ETOGL344
    1     YLAB,20)                                                  ETOGL345
C                                                                   ETOGL346
C     REORDER FOR GLEN PUNCH                                        ETOGL347
C                                                                   ETOGL348
      DO 140 N=1,NPM                                                ETOGL349
      NN1=NPM-N+1                                                   ETOGL350
      E(N)=EM(NN1)                                                  ETOGL351
      S(N)=SM(NN1)                                                  ETOGL352
  140 CONTINUE                                                      ETOGL353
C                                                                   ETOGL354
C     PUNCH FOR GLEN                                                ETOGL355
C                                                                   ETOGL356
      TITL(1)=10H  ELASTIC                                          ETOGL357
      TITL(2)=10HCROSS SECT                                         ETOGL358
      TITL(3)=10HION FOR MA                                         ETOGL359
      TITL(4)=10HTERIAL *                                           ETOGL360
      IF (NMT.NE.1) GO TO 141                                       ETOGL361
      PUNCH 200,(TITL(I),I=1,4),MAT1,TEMDS                          ETOGL362
      PUNCH 150,(S(N),N=1,NPM)                                      ETOGL363
  150 FORMAT (1P4E20.12)                                            ETOGL364
  200 FORMAT (4A10,I4,* TEMP=*1PE11.4,*DEG K*)                      ETOGL365
  210 FORMAT (4A10)                                                 ETOGL366
  141 CONTINUE                                                      ETOGL367
      IF (NMT.NE.2) GO TO 143                                       ETOGL368
      DO 142 N=1,NPM                                                ETOGL369
      XFISS(N)=S(N)                                                 ETOGL370
  142 CONTINUE                                                      ETOGL371
  143 CONTINUE                                                      ETOGL372
      IF (NMT.NE.3) GO TO 147                                       ETOGL373
      DO 144 N=1,NPM                                                ETOGL374
      XCAP(N)=S(N)                                                  ETOGL375
  144 CONTINUE                                                      ETOGL376
      DO 145 N=1,NPM                                                ETOGL377
```

```
      S(N)=XFISS(N)+XCAP(N)                                             ETOGL378
  145 CONTINUE                                                          ETOGL379
      TITL(1)=10H   ABSORPT                                             ETOGL380
      PUNCH 200,(TITL(I),I=1,4),MAT1,TEMDS                              ETOGL381
      PUNCH 150,(S(N),N=1,NPM)                                          ETOGL382
      XNU=1.0                                                           ETOGL383
      IF (MAT1.EQ.1157) XNU=2.4188                                      ETOGL384
      IF(MAT1.EQ.1260)XNU=2.498                                         ETOGL385
      DO 146 N=1,NPM                                                    ETOGL386
      S(N)=XNU*XFISS(N)                                                 ETOGL387
  146 CONTINUE                                                          ETOGL388
      TITL(1)=10H   NUFISSN                                             ETOGL389
      PUNCH 200,(TITL(I),I=1,4),MAT1,TEMDS                              ETOGL390
      PUNCH 150,(S(N),N=1,NPM)                                          ETOGL391
  147 CONTINUE                                                          ETOGL392
C                                                                       ETOGL393
C     REVERSE EC MESH FOR NEXT PASS.                                    ETOGL394
C                                                                       ETOGL395
      DO 211 K=1,NFGP                                                   ETOGL396
      SC(K)=EC(K)                                                       ETOGL397
  211 CONTINUE                                                          ETOGL398
      DO 212 K=1,NFGP                                                   ETOGL399
      K1=NFGP-K+1                                                       ETOGL400
      EC(K)=SC(K1)                                                      ETOGL401
  212 CONTINUE                                                          ETOGL402
  900 CONTINUE                                                          ETOGL403
 1000 CONTINUE                                                          ETOGL404
      STOP                                                              ETOGL405
 2000 WRITE (6,2010) MAT1,MAT                                           ETOGL406
 2010 FORMAT (1H1,*  SORRY,MAT = *I4,* NOT ON TAPE. LAST MAT = *I4)     ETOGL407
      STOP                                                              ETOGL408
      END                                                               ETOGL409
```

```
      FUNCTION LOCT (E,EK,N)                                            LOCT   1
C     BRACKETS EK IN E SO THAT EK.GE.E(LOCT) AND EK.LT.E(LOCT+1)        LOCT   2
C     IF EK CANNOT BE BRACKETED, LOCT=-1                                LOCT   3
      DIMENSION E(1)                                                    LOCT   4
C     RETURN LOCT=-1 IF ARRAY HAS ONLY ONE PT. (AS FOR NPTS IN XSEC).   LOCT   5
   50 FORMAT (1H0,1P8E15.5)                                             LOCT   6
      IF (N.LE.1) GO TO 10                                              LOCT   7
      M=N-1                                                             LOCT   8
      DO 5 I=1,M                                                        LOCT   9
    5 IF ((EK.GE.E(I)).AND.(EK.LT.E(I+1))) GO TO 15                     LOCT  10
      IF ((E(N-1).EQ.E(N)).AND.(EK.EQ.E(N))) GO TO 10                   LOCT  11
      IF (E(N).NE.EK) GO TO 10                                          LOCT  12
      LOCT=M                                                            LOCT  13
      RETURN                                                            LOCT  14
   10 LOCT=-1                                                           LOCT  15
      PRINT 50,(E(I),I=1,N),EK                                          LOCT  16
      RETURN                                                            LOCT  17
   15 LOCT=I                                                            LOCT  18
      RETURN                                                            LOCT  19
      END                                                               LOCT  20
```

```
      SUBROUTINE TERP1 (X1,Y1,X2,Y2,X,Y,I)
C     =====INTERPOLATE ONE PT.=====
C     (X1,Y1) AND (X2,Y2) ARE END PTS. OF THE LINE
C     (X,Y) IS INTERPOLATED POINT
C     I=INTERPOLATION CODE
C     NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE
C            INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.
C     ERROR STOPS - 301 (X1=X2,DISCONTINUITY)
C                   302 (INTERPOLATION CODE IS OUT OF RANGE)
C                   303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT.)
    5 XA=X1
      YA=Y1
      XB=X2
      YB=Y2
      XP=X
      II=I
      IF (XA.EQ.XB) CALL ERROR (301)
      IF (II) 10,10,15
   10 CALL ERROR (302)
   15 IF (II-5) 20,20,10
   20 GO TO (25,30,35,60,75), II
   25 YP=YA
      IF (XP.EQ.XB)  YP=YB
      GO TO 105
   30 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)
      GO TO 105
   35 IF (XA) 30,30,40
   40 IF (XB) 30,30,45
   45 IF (XP) 50,50,55
   50 CALL ERROR (303)
   55 YP=YA+ALOG(XP/XA)*(YB-YA)/ALOG(XB/XA)
      GO TO 105
   60 IF (YA) 30,30,65
   65 IF (YB) 30,30,70
   70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))
      GO TO 105
   75 IF (YA) 35,35,80
   80 IF (YB) 35,35,85
   85 IF (XA) 70,70,90
   90 IF (XB) 70,70,95
   95 IF (XP) 80,50,100
  100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))
  105 Y=YP
      RETURN
      END
```

```
      SUBROUTINE ERROR (N)
      IOS=9
    5 PRINT 10,N
      WRITE(99,10)
   10 FORMAT (11H ERROR STOP,I6)
      END
```

```
      SUBROUTINE UNION (XU,X,NPU,NP)
C
C  FUNCTION OF SUBROUTINE
C     UNION1 COMPUTES THE UNION OF INDEPENDENT VARIABLE SETS X(IP),IP=1,
C           XU(IP1),IP1=1,NPU, AND PLACES THE UNION INTO XU(IP2),IP2=1,NP
C  STORAGE
      DIMENSION XU(2000),KU(2000),X(200)
C
C     ADD A SET X   TO AN EXISTING UNION SET XU
      DO 106 IP=1,NPU
      KU(IP)=0
  106 CONTINUE
      DO 103 IP=1,NP
      IF (X(IP).LT.XU(NPU)) GO TO 120
      IF (X(IP).GT.XU(1)) GO TO 130
      DO 104 IP1=1,NPU
      IF (X(IP).EQ.XU(IP1)) GO TO 140
      IF (IP1.EQ.NPU) GO TO 105
      IF (X(IP).LT.XU(IP1).AND.X(IP).GT.XU(IP1+1)) GO TO 150
  105 CONTINUE
  104 CONTINUE
C
C     HERE NPU IS INCREMENTED BY ONE AND A POINT IS ADDED TO THE LEFT
  120 NPU=NPU+1
      XU(NPU)=X(IP)
      KU(NPU)=1
  121 CONTINUE
      GO TO 103
C
C     HERE CONTROLS ARE SET TO ADD A POINT ON THE RIGHT
  130 KONREL=1
      NPMOV=NPU
      GO TO 170
C
C     HERE NPU IS NOT INCREMENTED BY ONE .
 140  CONTINUE
      KU(IP1)=1
      GO TO 103
C
C     HERE NPU IS INCREMENTED BY ONE AND CONTROLS ARE SET TO ADD A POINT
C           BETWEEN POINTS IP1 AND IP1+1
  150 KONREL=2
      NPMOV=NPU-IP1
      GO TO 170
C
C     HERE WE INCREMENT NPU BY ONE AND MOVE THE LEFT-MOST NPMOV POINTS I
C                SET ONE POSITION TO THE LEFT
  170 NPU=NPU+1
      DO 171 IP2=1,NPMOV
      XU(NPU-IP2+1)=XU(NPU-IP2)
      KU(NPU-IP2+1)=KU(NPU-IP2)
  171 CONTINUE
C
C     HERE A NEW POINT IS ADDED
      NPADD=NPU-NPMOV
      XU(NPADD)=X(IP)
      KU(NPADD)=1
  172 CONTINUE
  103 CONTINUE
  102 RETURN
      END
```

APPENDIX C

MERGFAT

A code to merge fast and thermal cross section sets.

Multigroup cross-section data sets for energy groups above the thermal boundary energy (2.58 eV for the HTGR) are generated by the 1DX code, whereas data for those groups below this energy are generated by the GLEN code (see Fig. 3). Usually there is at least one overlapping group. MERGFAT is a small code, the purpose of which is to combine these two sets into a single set properly formatted for input to the Los Alamos S_n codes.

In addition to the files containing the fast and thermal data, the input consists of designations of groups to be merged, designations of materials to be read from the files, the final energy boundaries (which are used in the computation of the velocities needed by the S_n codes), and the final groupwise values of the fraction of the fissions in each group (χ), also needed in the S_n calculations. Table C-I describes the input needed for MERGFAT, and a listing of the code is given at the end of this appendix.

TABLE C-I

INPUT SPECIFICATIONS FOR MERGFAT

Card No.	Format	Variable	Comment
1	9A8	HLT(I)	Should read "LASTDECK" at the last set of input cards.
2	12I6	LENG	Table length of final output, including upscatter, self-scatter, and down-scatter.
		NDELU	Obsolete.
		LTL	Obsolete.
		LNGUP	Length of up-scatter table.
3	8A10	TITLE(I)	Title card.
4	3I12	NOBG	Total number of final groups.
		NOI	Number of materials for which cross sections are to be prepared.
		IOPT	Obsolete.

TABLE C-1 (cont)

5	6E12.5	GPEN(I)	NOBG values of lower group bounds in eV.
6	6E12.5	XI(I)	NOBG values of X.
7	12I6	NDKS	Number of sets to be merged (2).
		KG1(N),KG2(N)	NDKS values for first and last group in the sets being merged that are to be included in the final set.
8	A6,2X,A10,A6	NUCLE	Nuclide identifier assigned in 1DX input (see Ref. 14).
		MODER	Moderator, absorber identifier. Use word "MODERAT" for moderator and "ISOTOPE" for absorber.
		MATID	ENDF/B MAT number.

The "fast" data file output by the 1DX code is designated as NTPF in MERGFAT, and the "thermal" data file is designated as NTPT. Card 8 is repeated NOI times for the number of materials to be processed in one run.

LASL Identification No. LP-0757

```
      PROGRAM MERGFAT (INP,OUT,PUN,FSET6=OUT,FSET7=PUN,FSET8=INP,
     1    FSET9,FSET10)
C     MERGFAT IS A VERSION OF JUMBLFAT THAT ACCEPTS FAST XSEC AS
C     OUTPUT BY THE HONEDX VERSION OF THE 1DX CODE.
C     REVISIONS MADE AT LASL BY LABAUVE,NOV75.
C
      DIMENSION TOTP0(70),TOTP1(70),TOTRA(70)
      DIMENSION NUCID(20),N2N(20),SK2KF(70,70),DIAGSM(70),CHKSM(70),
     1    DIFF(70),CAPA(70),FISA(70),STR(70),AVNU(70),SINTRA(70,70),
     2    SCAP(70),SS(70),SABS(70),HOL(70),KK(70),SSN2N(70,70),
     3    XNUSTG(70),A(2000),P0(70,70),P1(70,70),P2(70,70),P3(70,70),
     4    XS(70),TITLE(12),TOTN2N(70),SNP(70),SND(70),SNT(70),SNHE3(70),
     5    SNA(70),SN2A(70)
      DIMENSION XI(50),GPEN(50),TTL(12),VEL(50),FACAP(50),FAFIS(50),
     1  ADEN(20),TOTIN(70)
      DIMENSION KG1(20),KG2(20)
      DIMENSION V(50),HLT(10)
      NTPF=9
      NTPT=10
 5002 READ(8,5001)(HLT(I),I=1,9)
 5001 FORMAT(9A8)
      PRINT 600
  600 FORMAT (1H120X10HINPUT DATA ///)
      READ (8,5) LENG,NDELU,LTL,LNGUP
      PRINT 601,LENG,NDELU,LTL,LNGUP
```

```
 601 FORMAT (1H010X7HLENG = ,I6,10H, NDELU = ,I6,8H, LTL = ,I6,
    110H, LNGUP = ,I6)
     READ( 8,70   )(TITLE(I),I=1,7)
     PRINT 602,(TITLE(I),I=1,7)
 602 FORMAT (1H05X,8A10)
     READ (8,71) NOBG,NOI,IOPT
     PRINT 603,NOBG,NOI,IOPT
 603 FORMAT (1H010X7HNOBG = ,I12,8H, NOI = ,I12,9H, IOPT = ,I12)
     NMAT=NOI
  71 FORMAT (3I12)
     KGROPS=NOBG
  70 FORMAT (8A10)
      PUNCH 1000, (TITLE(I),I=1,12)
     READ ( 8,3)  (GPEN(I),I=1,KGROPS)
     PRINT 604
 604 FORMAT (4H0   I3X7HGPEN(I))
     PRINT 605,(I,GPEN(I),I=1,KGROPS)
 605 FORMAT (I3,1PE12.5)
     AVEN1=(GPEN(KGROPS-1)+GPEN(KGROPS))/2.0
     AVEN2=(GPEN(KGROPS))/2.0
     FAXT=0.
     READ ( 8,3)  (XI(I),I=1,KGROPS)
     PRINT 606
 606 FORMAT (4H0   I6X5HXI(I))
     PRINT 605,(I,XI(I),I=1,KGROPS)
     V(1)=1.0E+07
     DO 4000 K=1,KGROPS
4000 V(K+1)=GPEN(K)
     DO 72 K=1,KGROPS
  72 VEL(K)=0.007*(V(K)**0.5+V(K+1)**0.5)
C     *** NOTE *** THESE ARE JUST AVERAGE BROAD GROUP VELOCITIES.
     LBL1=6H SORS
     LBL2=6H VELS
     I1=1
     M1=1
  73 I2=I1+5
     IF (I2.LE.KGROPS) GO TO 74
     PUNCH 75,(XI(I),I=I1,KGROPS)
  75 FORMAT (6F12.6)
     GO TO 76
  74 PUNCH 87,(XI(I),I=I1,I2),LBL1,M1
     M1=M1+1
     I1=I2+1
     GO TO 73
  76 CONTINUE
     I1=1
     M1=1
  77 I2=I1+5
     IF (I2.LE.KGROPS) GO TO 78
     PUNCH 79,(VEL(I),I=I1,KGROPS)
  79 FORMAT (1P6E12.2)
     GO TO 1180
  78 PUNCH 81,(VEL(I),I=I1,I2),LBL2,M1
     M1=M1+1
     I1=I2+1
     GO TO 77
1180 CONTINUE
  87 FORMAT (   6F12.6,A6,I2)
  81 FORMAT (1P6E12.2,A6,I2)
   5 FORMAT (12I6)
     MATNO=1
     READ (8,5) NDKS,(KG1(N),KG2(N),N=1,NDKS)
     PRINT 607,NDKS
```

```
  607 FORMAT (1H010X7HNDKS = ,I6)
      PRINT 608
  608 FORMAT (4H0   N3X6HKG1(N)7X6HKG2(N))
      PRINT 609,(N,KG1(N),KG2(N),N=1,NDKS)
  609 FORMAT (I3,2I6)
      DO 999 M=1,NMAT
      REWIND NTPF $ REWIND NTPT
      READ (8,4100) NUCLE,MODER,MATID
 4100 FORMAT (A6,2X,A10,A6)
      DO 500 N=1,NDKS
      K1=KG1(N)
      NOBG=KG2(N)
      LTABL=LENG+6
         NOBG2 = NOBG + 1
      FAFIS(N)=1.0
 2102 FACAP(N)=1.
   85 FORMAT (15H              G=03,3H L=03,13H NUCLIDE NO.=05,1H 3A10)
 4110 IF (N.EQ.1) READ (NTPF,4115) NUCID(M),FID,(TTL(KK),KK=1,11)
 4115 FORMAT (13A6)
      IF (NUCID(M).NE.NUCLE) GO TO 4110
 1085 FORMAT (A6,F6.2,11A6)
      DO 3070 K=K1,NOBG
      DO 3070 KF=K1,NOBG
      P0(K,KF)=0.0
      SINTRA(K,KF)=0.0
      SSN2N(K,KF)=0.0
 3070 CONTINUE
      IF (N.EQ.1) GO TO 3000
      GNU=AVNU(K1)
      MTBL=2*LNGUP+5
      MGPS=NOBG-K1+1
      MVLG=MTBL*MGPS
      DO 3010 L=1,2
 4120 READ (NTPT,4130) MODES,MATCH
 4130 FORMAT (22X,A10,15X,A6)
      IF (MODES.EQ.MODER.AND.MATCH.EQ.MATID) GO TO 4135
      GO TO 4120
115        .03
 4135 CONTINUE
      IF (L.EQ.1) MADD=0
      IF (L.EQ.2) MADD=MVLG
      READ (NTPT,3) (A(M+MADD),M=1,MVLG)
 3010 CONTINUE
      DO 3020 MG=1,MGPS
      DO 3030 MGM=1,MGPS
      KK1=K1+MG-1
      KK2=K1+MGM-1
      LCK=9-MG+(MGM-1)*(MTBL+1)
      P0(KK1,KK2)=A(LCK)
      P1(MG,MGM)=A(LCK+MVLG)
 3030 CONTINUE
      PRINT 3040,KK1
 3040 FORMAT (1H0,*  SCATT. MATRIX FOR GP *I3)
      PRINT 3050,(P0(KK1,KK2),KK2=K1,NOBG)
      PRINT 3050,(P1(MG,MGM),MGM=1,MGPS)
      LCK=(MG-1)*MTBL+1
      SABS(KK1)=A(LCK)
      XNUSIG(KK1)=A(LCK+1)
      STR(KK1)=A(LCK+2)
      AVNU(KK1)=GNU
      IF (GNU.LE.0.0) GNU=1.E-10
      FISA(KK1)=XNUSIG(KK1)/GNU
      CAPA(KK1)=SABS(KK1)-FISA(KK1)
      P1TOT=0.0
```

```
3050 FORMAT (1P8E15.5)                                                    MERGF153
     DO 3060 MGM=1,MGPS                                                   MERGF154
     P1TOT=P1TOT+P1(MG,MGM)                                               MERGF155
3060 CONTINUE                                                             MERGF156
     PRINT 3050,P1TOT                                                     MERGF157
     P0(KK1,KK1)=P0(KK1,KK1)-P1TOT                                        MERGF158
     STR(KK1)=STR(KK1)-P1TOT                                              MERGF159
     PRINT 3050,P0(KK1,KK1),STR(KK1)                                      MERGF160
3020 CONTINUE                                                             MERGF161
     GO TO 3500                                                           MERGF162
3000 CONTINUE                                                             MERGF163
     PRINT 4115,NUCID(M),FID,(TTL(KK),KK=1,11)                            MERGF164
   1 FORMAT (8A10)                                                        MERGF165
     DO 10 K=1,NOBG                                                       MERGF166
     KF=K-1                                                               MERGF167
     IF (K.EQ.1) KF=NOBG2                                                 MERGF168
     READ (NTPF,3) FISA(K),SABS(K),XNUSIG(K),STR(K),P0(K,K),P0(KF,K)      MERGF169
     READ (NTPF,3) ZILCH                                                  MERGF170
     IF (ZILCH.EQ.0.0) GO TO 20                                           MERGF171
     PRINT 19,ZILCH                                                       MERGF172
  19 FORMAT (1H1,*  ZILCH = *1PE12.5,* READ ERROR*)                       MERGF173
  20 CONTINUE                                                             MERGF174
   3 FORMAT (6E12.5)                                                      MERGF175
     CAPA(K)=SABS(K)-FISA(K)                                              MERGF176
     IF (FISA(K).GT.0.0) AVNU(K)=XNUSIG(K)/FISA(K)                        MERGF177
  10 CONTINUE                                                             MERGF178
3500 CONTINUE                                                             MERGF179
 500 CONTINUE                                                             MERGF180
     LOD=-1                                                               MERGF181
     DO 95 K=1,NOBG                                                       MERGF182
     TOTP0(K)=0.                                                          MERGF183
     TOTP1(K)=0.                                                          MERGF184
     TOTRA(K)=0.                                                          MERGF185
     TOTN2N(K)=0.                                                         MERGF186
     SS(K)=0.                                                             MERGF187
     XNUSIG(K)=FISA(K)*AVNU(K)                                            MERGF188
     DO 80 KF=K,NOBG2                                                     MERGF189
     TOTRA(K)=TOTRA(K)+SINTRA(K,KF)                                       MERGF190
     IF (LOD.EQ.-1) SK2KF(K,KF)=P0(K,KF)+SINTRA(K,KF)+SSN2N(K,KF)*2.      MERGF191
     SS(K)=SS(K)+SK2KF(K,KF)                                              MERGF192
     TOTN2N(K)=TOTN2N(K)+SSN2N(K,KF)                                      MERGF193
  80 CONTINUE                                                             MERGF194
     CAPA(K)=CAPA(K)+SNP(K)+SNA(K)                                        MERGF195
     SABS(K)=CAPA(K)+FISA(K)                                              MERGF196
  95 CONTINUE                                                             MERGF197
     LOD1=LOD+2                                                           MERGF198
     HOL(2)=10H ETOG-ENDF                                                 MERGF199
     GO TO (110,120,130,140,150),LOD1                                     MERGF200
 110 HOL(1)=10H ISOTROP T                                                 MERGF201
     HOL(2)=10HRNSP TABLE                                                 MERGF202
     GO TO 160                                                            MERGF203
 120 HOL(1)=10H P-0 TABLE                                                 MERGF204
     GO TO 160                                                            MERGF205
 130 HOL(1)=10H P-1 TABLE                                                 MERGF206
     GO TO 160                                                            MERGF207
 140 HOL(1)=10H P-2 TABLE                                                 MERGF208
     GO TO 160                                                            MERGF209
 150 HOL(1)=10H P-3 TABLE                                                 MERGF210
 160 CONTINUE                                                             MERGF211
     LOD=LOD+1                                                            MERGF212
     L=LENG                                                               MERGF213
     N=LENG-1                                                             MERGF214
```

```
C DOWN SCATTERING IS CONSIDERED ONLY BETWEEN ADJACENT GROUPS EXCEPT FOR
C FIRST L GROUPS IN WHICH ALL CASES ARE CONSIDERED.  ADDITIONAL VALUES A
C ADDED INTO THE L -TH GROUP.
      DO 90 K=1,NOBG
      MN=LENG+K
      IF (MN.GT.NOBG+1) GO TO 90
      DO 91 KF=MN,NOBG2
   91 SK2KF(K,MN-1)=SK2KF(K,MN-1)+SK2KF(K,KF)
   90 CONTINUE
      MK=NOBG-LENG+2
      DO 51 K1=MK,NOBG
   51 SK2KF(K1,NOBG)=SK2KF(K1,NOBG)+SK2KF(K1,NOBG2)
  300 CONTINUE
      LTABL=LENG+LNGUP+7
      NXC=LTABL*NOBG
      NCX6=NXC+LTABL
      DO 360 JT=1,NCX6
  360 A(JT)=0.
      DO 361 K=1,NOBG
      J1=(K-1)*LTABL+1
      J2=J1+1
      J3=J1+2
      J4=J3+1
      J5=J4+1
      J6=J5+1
      J7=J6+1
      A(J1)=FISA(K)
      A(J4)=CAPA(K)
      A(J5)=SABS(K)
      A(J6)=XNUSIG(K)
      A(J7)=STR(K)
      DO 362 KF= K,NOBG2
      J8=(KF-1)*LTABL+KF+7-K+1+LNGUP
      A(J8)=SK2KF(K,KF)
      IF ((KF-K).GE.  LENG) GO TO 361
  362 CONTINUE
      IF (FID.NE.6H 12.00) GO TO 361
      IF(K.LT.KGI(NDKS)) GO TO 361
      KUP1=K+1
      IF(KUP1.GT.NOBG) GO TO 361
      DO 3561 KF=KUP1,NOBG
      J9=LTABL*(K-1)+8+LNGUP-(KF-K)
      A(J9)=P0(KF,K)
 3561 CONTINUE
  361 CONTINUE
  250 CONTINUE
      PUNCH 85,NOBG,LTABL,MATNO,NUCID(M),(HOL(I),I=1,2)
      NP1=1
      NBR=1
   63 NP2=NP1+5
      IF (NP2.LE.NXC ) GOTO 67
      PUNCH 65,(A(NP),NP=NP1,NP2),NUCID(M),NBR
   68 FORMAT (1P6E12.5)
      GO TO 64
   67 PUNCH 65,(A(NP),NP=NP1,NP2),NUCID(M),NBR
      IF (NP2.EQ.NXC ) GOTO 64
      NBR=NBR+1
      NP1=NP2+1
      GO TO 63
   64 CONTINUE
   65 FORMAT (1P6E12.5,A6,I2)
      DO 103 K=1,NOBG
```

```
  103 KK(K)=K
      IF(M.NE.1)GOTO9
      WRITE(6,1000)(TITLE(I),I=1,7)
 1000 FORMAT(8A10)
    9 CONTINUE
      WRITE(6,85) NOBG,LTABL,MATNO,NUCID(M),(HOL(I),I=1,2)
C CHECK TO ADD UP SK2KF
C  THE DIAGONAL SUM OF THE DOWN SCATTERING AND SELF SCATTERING ADDED TO
C ABSORPTION MUST EQUAL THE TRANSPORT CROSS SECTION.
      DO 400 K=1,NOBG
      DIAGSM(K)=0.
      KK=K+LENG-1
      IF (KK.GT.NOBG) KK=NOBG
      DO 450 KF=K,KK
  450 DIAGSM(K)=DIAGSM(K)+SK2KF(K,KF)
      CHKSM(K)=DIAGSM(K)+SABS(K)-TOTN2N(K)
      DIFF(K)=CHKSM(K)-STR(K)
      WRITE(6,460)
  460 FORMAT (20H0TRANSPORT          18HCHECKSUM
    1     10HDIFFERENCE)
      WRITE(6,470) STR(K),CHKSM(K),DIFF(K)
  470 FORMAT(E12.4,7X,E12.4,8X,E12.4)
      IF(DIFF(K).LE..1*STR(K))GOTO400
      WRITE(6,480)
  480 FORMAT(31H0DIFFERENCE EXCEEDS 10 PER CENT)
  400 CONTINUE
      WRITE(6,85) NOBG,LTABL,MATNO,NUCID(M),(HOL(I),I=1,2)
      DO 701 I=1,NOBG
      K=LTABL*(I-1)
      DO 701 J=1,LTABL
  701 SK2KF(I,J)=A(K+J)
      KA=1 $ KB=8
  724 KC=MIN0(KB,NOBG)
      WRITE(6,720) (K,K=KA,KC)
      WRITE(6,721)
      DO 722 J=1,LTABL
  722 WRITE (6,723) J,(SK2KF(I,J),I=KA,KC)
      KA=KA+8 $ KB=KB+8
      IF(KA.LE.NOBG) GO TO 724
  720 FORMAT (1H0,5X,8(2X,5HGROUP,I3,3X))
  721 FORMAT (1H )
  723 FORMAT (I4,1P8E13.5)
  100 CONTINUE
      MATNO=MATNO+1
  999 CONTINUE
      IF(HLT(1).NE.8HLASTDECK) GO TO 5002
      END
```

APPENDIX D

DANCOFF FACTOR FOR A REGULAR ARRAY OF CYLINDRICAL FUEL RODS

The Dancoff factor is an important quantity in the Levine method of space-shielding cross sections to account for the gross (fuel-rod) heterogeneity in the reactor core. For this purpose, a special computer program was written to calculate the Dancoff factor by three methods and to compare their results.

One method, due to Carlvik,[22] gives the Dancoff factor by exact integration:

$$C = \frac{1}{\alpha_0} \int_0^{\alpha_0} d\alpha \, \frac{1}{2r} \int_{-r}^{r} dy \, \frac{Ki_3(t)}{Ki_e(0)} \quad , \tag{D-1}$$

where Ki_3 is the Bickley function, t is the optical length between rods, r is the radius of one rod, and α_0 is 30° for a hexagonal lattice.

For a hexagonal lattice, y = r/d, where d is the lattice pitch and y is related to the mean chord length $\bar{\ell}$ in the moderator through

$$\bar{\ell} \equiv \frac{4V_m}{S} = \frac{2d}{\pi y}\left(\frac{\sqrt{3}}{2} - \pi y^2\right) \quad . \tag{D-2}$$

Sauer[30] has found a good approximation for C:

$$C = \frac{e^{-t\Sigma\bar{\ell}}}{1 + (1 - t)\Sigma\bar{\ell}} \quad , \tag{D-3}$$

where Σ is the moderator cross section and, for a hexagonal lattice

$$t = \frac{\pi}{2} y \, \frac{1 - 2y}{\frac{\sqrt{3}}{2} - \pi y^2} - 0.12 \quad . \tag{D-4}$$

Bonalumi[31] has pointed out that Sauer's Dancoff correction is bad approximation for very large moderator cross sections in the two cases of very large and very small volume ratios, i.e., for y near 0 and near 0.5.

Bonalumi has, therefore, suggested the following modification:

$$C = \frac{e^{-t\Sigma\bar{\ell}}}{1 + (1 - t_1)\Sigma\bar{\ell}} \quad , \tag{D-5}$$

where

$$t_1 = t + \frac{\Sigma\bar{\ell}}{7 + \beta\Sigma\bar{\ell}} \quad , \tag{D-6}$$

and $\beta = 2.125$ for a hexagonal lattice.

For the HTGR core configurations under study, all three methods of calculating the Dancoff factor have been found good, yielding very close answers. The listing of the computer program used for this comparison is included at the end of this appendix.

LASL Identification No. LP-0758

```
      PROGRAM DANCPIN(INP,OUT,PUN,FILM)                                 DANCP  1
C     CALCULATES DANCOFF FACTOR FOR A REGULAR ARRAY OF INFINITE CYLINDERSDANCP  2
C      INPUT QUANTITIES                                                 DANCP  3
C      NALF AND NRAD DETERMINE AN INTEGRATION GRID FOR THE CARLVIK      DANCP  4
C      INTEGRATION. THEY ARE BOTH TAKEN TO BE 128 .                     DANCP  5
C      NLAT DETERMINES THE TYPE OF LATTICE , IT IS 4 FOR A SQUARE       DANCP  6
C      LATTICE AND 6 FOR A HEXAGONAL LATTICE .                          DANCP  7
C      IF IOPTC=0, ALL THREE METHODS ARE COMPARED , IF IOPTC=1,         DANCP  8
C      THE CARLVIK ROUTINE IS USED , IF IOPTC=2, THE BONALUMI           DANCP  9
C      APPROXIMATION ONLY IS USED , IF IOPTC=3, THE SAUER               DANCP 10
C      APPROXIMATION ONLY IS USED  .                                    DANCP 11
C      RAD0 IS THE PIN RADIUS IN CM .                                   DANCP 12
C      RAD1 IS THE MODERATOR RADIUS IN THE THREE-REGION MODEL .         DANCP 13
C      GAPWID IS THE GAP WIDTH (CM) AROUND THE PIN .                    DANCP 14
C      RAD1S IS THE MODERATOR RADIUS WHEN THE GAP-WIDTH IS NOT          DANCP 15
C      EXPLICITLY GIVEN AS IN THE SAUER OR THE BONALUMI APPROXIMATIONS .DANCP 16
C      SIGF IS THE MACROSCOPIC FUEL-PIN CROSS SECTION (1/CM),           DANCP 17
C      DENSF IS THE ATOMIC CONCENTRATION OF THE FUEL PIN ,              DANCP 18
C      SIGMAM IS THE MODERATOR MACROSCOPIC CROSS SECTION (1/CM) .       DANCP 19
      READ 5,NLAT,NALF,NRAD,IOPTC,RAD1S                                 DANCP 20
    5 FORMAT(4I10,2E10.4)                                               DANCP 21
      IF(NLAT.NE.4.OR.NLAT.NE.6) PRINT 6                                DANCP 22
    6 FORMAT(1H0,*NLAT MUST BE EQUAL TO 4 OR TO 6 , TRY AGAIN*)         DANCP 23
      IF(NRAD.LE.128) NRAD=128                                          DANCP 24
      IF(NALF.LE.128) NALF=128                                          DANCP 25
      READ 10,RAD0,RAD1,GAPWID,SIGMAM,SIGF,DENSF                        DANCP 26
   10 FORMAT(6F12.6)                                                    DANCP 27
      PRINT 15,NLAT,NALF,NRAD,IOPTC,RAD1S                               DANCP 28
   15 FORMAT (1H1,*NLAT = *,I4,* NALF = *,I4, * NRAD = *,I4,* IOPTC = *,DANCP 29
     1I4,* RAD1S = *,E10.4)                                             DANCP 30
      PRINT 16,RAD0,RAD1,GAPWID,SIGMAM,SIGF,DENSF                       DANCP 31
   16 FORMAT(1H0,*RAD0 = *,E12.6,* RAD1 = *,E12.4,* GAPWID = *,E12.6//  DANCP 32
     1* SIGMAM = *,E12.6,* SIGF = *,E12.6,* DENSF = *,E12.6)            DANCP 33
      IF(IOPTC)20,20,30                                                 DANCP 34
```

```
20 DO 25 IOPTC=1,3
   CALL DANCOFF(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC,CS,
  1CSB,C,IOPTC,RAD1S)
   CALL LEVINE(SIGF,DENSF,C,RAD0,IOPTC)
25 CONTINUE
   GO TO 50
30 CALL DANCOFF(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC,CS,
  1CSB,C,IOPTC,RAD1S)
   CALL LEVINE(SIGF,DENSF,C,RAD0,IOPTC)
50 CONTINUE
   END
```

```
      SUBROUTINE LEVINE(SIGF,DENSF,C,RAD0,IOPTC)
C     CALCULATES THE EFFECTIVE GEOMETRIC SHIELDING CROSS SECTION BY THE
C     LEVINE METHOD AND USING THE OTTER APPROXIMATION FOR THE LEVINE
C     FACTOR
      ELBARF=2.*RAD0
      TAUF=SIGF*ELBARF
      IF(TAUF.LE.0.) PRINT 20
   20 FORMAT(1H0,*TAUF IS G. OR E. TO ZERO*)
      IF(TAUF.GE.2.) GO TO 30
      ALEVI=1.5013+0.14879*TAUF**0.5-0.17226*TAUF
      GO TO 40
   30 ALEVI=1.+1./TAUF-1./TAUF**3.
   40 CONTINUE
      PRINT 50,ALEVI
   50 FORMAT(1H0,*THE LEVINE FACTOR EQUALS *,1PE12.4)
      QDEN=ELBARF*(1.+(ALEVI-1.)*C)
      QNUM=ALEVI*(1.-C)
      EFXSEC=QNUM/QDEN/DENSF
      IF(IOPTC.EQ.1) ANAME=10H(CARLVIK)
      IF(IOPTC.EQ.2) ANAME=10H(BONALUMI)
      IF(IOPTC.EQ.3) ANAME=10H (SAUER)
      PRINT 10,ANAME,EFXSEC
   10 FORMAT(1H0,*EFFECTIVE SIGMA WITH DANCOFF FACTOR *,A10,* = *,
     11PE12.4)
      RETURN
      END
```

```
      SUBROUTINE DANCOFF(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC,CS,
     1                   CSB,C,IOPTC,RAD1S)
C     CALCULATES DANCOFF FACTORS BY THE ORIGINAL METHOD AS USED BY
C     CARLVIK, BY THE SAUER APPROXIMATION AND BY THE BONALUMI
C     APPROXIMATION AND COMPARES THE RESULTS OF THE THREE
      IF(IOPTC)10,10,20
   10 PRINT 15
   15 FORMAT(1H0,*IOPTC IS ZERO OR NEGATIVE,TRY AGAIN*)
      RETURN
```

```
20 IF(IOPTC-1)10,30,40                                                  DANCO 10
30 CALL CARLVIK(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC)              DANCO 11
   PRINT 35,CC                                                          DANCO 12
35 FORMAT(1H0,*DANCOFF FACTOR (CARLVIK) = *,E12.6)                      DANCO 13
   C=CC                                                                 DANCO 14
   RETURN                                                               DANCO 15
40 IF(IOPTC-2)30,50,60                                                  DANCO 16
50 CALL BONAL(RAD0,RAD1S,NLAT,SIGMAM,CSB)                               DANCO 17
   PRINT 45,CSB                                                         DANCO 18
45 FORMAT(1H0,*DANCOFF FACTOR (BONALUMI) = *,E12.6)                     DANCO 19
   C=CSB                                                                DANCO 20
   RETURN                                                               DANCO 21
60 IF(IOPTC-3)50,70,80                                                  DANCO 22
70 CALL SAUER(RAD0,RAD1S,NLAT,SIGMAM,CS)                                DANCO 23
   PRINT 55,CS                                                          DANCO 24
55 FORMAT(1H0,*DANCOFF FACTOR (SAUER) = *,E12.6)                        DANCO 25
   C=CS                                                                 DANCO 26
   RETURN                                                               DANCO 27
80 PRINT 90                                                             DANCO 28
90 FORMAT(1H0,*IOPTC IS GREATER THAN 3,TRY AGAIN*)                      DANCO 29
   RETURN                                                               DANCO 30
   END                                                                  DANCO 31
```

```
      SUBROUTINE CARLVIK(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC)     CARLV  1
C     CALCULATES DANCOFF FACTORS BY THE ORIGINAL METHOD AS IMPLEMENTED  CARLV  2
C     BY CARLVIK                                                        CARLV  3
      PI=3.141592654                                                    CARLV  4
      GAM=0.0                                                           CARLV  5
      IF(NLAT.EQ.6) GAM=PI/6.0                                          CARLV  6
      CNST1=(1./PI)**0.5                                                CARLV  7
      CNST2=(3.**0.5/2.)**0.5                                           CARLV  8
      IF(NLAT.EQ.6)CNST1=CNST1*CNST2                                    CARLV  9
      PITCH=RAD1/CNST1                                                  CARLV 10
      R=RAD0/PITCH                                                      CARLV 11
      E=(RAD0+GAPWID)/PITCH                                             CARLV 12
      E2=E*E                                                            CARLV 13
      CONST=2.0/(PI*NALF*NRAD)                                          CARLV 14
      SIG   =SIGMAM*PITCH                                               CARLV 15
      CC=0.0                                                            CARLV 16
      ISIG=10./SIG+1.                                                   CARLV 17
      NROW=MIN0(100,ISIG)                                               CARLV 18
      I1=2*NRAD                                                         CARLV 19
      DZ=R/NRAD                                                         CARLV 20
      DALF=PI/(NLAT*NALF)                                               CARLV 21
      ALF=-0.5*DALF                                                     CARLV 22
      DO 60 N=1,NALF                                                    CARLV 23
      ALF=ALF+DALF                                                      CARLV 24
      CAG=COS(ALF+GAM)                                                  CARLV 25
      DX=COS(GAM)/CAG                                                   CARLV 26
      DY=SIN(ALF)/CAG                                                   CARLV 27
      T=SIN(ALF+GAM)/CAG                                                CARLV 28
      Z=-R-0.5*DZ                                                       CARLV 29
      DO 50 I=1,I1                                                      CARLV 30
      Z=Z+DZ                                                            CARLV 31
      X=Z*T-SQRT(E2-Z*Z)                                                CARLV 32
      F=CAG-Z                                                           CARLV 33
      IF(F.GE.E)GO TO 10                                                CARLV 34
      IF(F.LE.R)GO TO 40                                                CARLV 35
      X=X-2.0*SQRT(E2-F*F)                                              CARLV 36
```

```
10 Y=Z/CAG+1.0                                                  CARLV 37
   DO 30 J=1,NROW                                               CARLV 38
   IY=Y+DY                                                      CARLV 39
   Y=Y+DY-IY                                                    CARLV 40
   X=X+DX                                                       CARLV 41
   F=-CAG*Y                                                     CARLV 42
   IF(F.LE.(-E))GO TO 12                                        CARLV 43
   IF(F.GE.(-R))GO TO 40                                        CARLV 44
   X=X-2.0*SQRT(E2-F*F)                                         CARLV 45
12 F=CAG+F                                                      CARLV 46
   IF(F.GE.E)GO TO 30                                           CARLV 47
   IF(F.LE.R)GO TO 40                                           CARLV 48
   X=X-2.0*SQRT(E2-F*F)                                         CARLV 49
30 CONTINUE                                                     CARLV 50
   GO TO 50                                                     CARLV 51
40 X=X+F*T-SQRT(E2-F*F)                                         CARLV 52
   Q=SIG*X                                                      CARLV 53
   CALL BKLY(Q,BIC3)                                            CARLV 54
   CC=CC+BIC3                                                   CARLV 55
50 CONTINUE                                                     CARLV 56
60 CONTINUE                                                     CARLV 57
   CC=CONST*CC                                                  CARLV 58
   RETURN                                                       CARLV 59
   END                                                          CARLV 60
```

```
      SUBROUTINE SAUER(RAD0,RAD1,NLAT,SIGMAM,CS)                          SAUER  1
C     CALCULATES DANCOFF FACTORS BY THE SAUER APPROXIMATION               SAUER  2
      PI=3.141592654                                                      SAUER  3
      RADRA=RAD1/RAD0                                                     SAUER  4
      VOLRA=RADRA*RADRA-1.                                                SAUER  5
      VOLSQR=(1.+VOLRA)**0.5                                              SAUER  6
      IF(NLAT.EQ.4)TAU=((PI/4.)**0.5*VOLSQR-1.)/VOLRA-0.08                SAUER  7
      IF(NLAT.EQ.6)TAU=((PI/(3.**0.5*2.))**0.5*VOLSQR-1.)/VOLRA-0.12      SAUER  8
      IF(TAU)10,10,20                                                     SAUER  9
   10 PRINT 15                                                            SAUER 10
   15 FORMAT(1H0,*TAU IS ZERO,NLAT IS WRONG*)                             SAUER 11
      RETURN                                                              SAUER 12
   20 ELBARF=2.*RAD0                                                      SAUER 13
      ELBARM=ELBARF*VOLRA                                                 SAUER 14
      PROD=SIGMAM*ELBARM                                                  SAUER 15
      DANCOF=EXP(-TAU*PROD)/(1.+(1.-TAU)*PROD)                            SAUER 16
      CS=DANCOF                                                           SAUER 17
      RETURN                                                              SAUER 18
      END                                                                 SAUER 19
```

```
      SUBROUTINE BONAL(RAD0,RAD1,NLAT,SIGMAM,CSB)
C     CALCULATES DANCOFF FACTORS BY THE BONALUMI APPROXIMATION
      TAU=0.
      PI=3.141592654
      RADRA=RAD1/RAD0
      VOLRA=RADRA*RADRA-1.
      VOLSQR=(1.+VOLRA)**0.5
      IF(NLAT.EQ.4)TAU=((PI/4.)**0.5*VOLSQR-1.)/VOLRA
      IF(NLAT.EQ.6)TAU=((PI/(3.**0.5*2.))**0.5*VOLSQR-1.)/VOLRA
      IF(TAU)10,10,20
   10 PRINT 15
   15 FORMAT(1H0,*TAU IS ZERO,NLAT VALUE IS WRONG*)
      RETURN
   20 ELBARF=2.*RAD0
      ELBARM=ELBARF*VOLRA
      PROD=SIGMAM*ELBARM
      IF(NLAT.EQ.4)BETA=5.67
      IF(NLAT.EQ.6)BETA=2.125
      DELTAU=PROD/(7.+BETA*PROD)
      TAU1=TAU+DELTAU
      DANCOF=EXP(-TAU*PROD)/(1.+(1.-TAU1)*PROD)
      CSB=DANCOF
      RETURN
      END
```

```
      SUBROUTINE BKLY(X,BIC3)
C     CALCULATES BICKLEY FUNCTIONS OF THE THIRD ORDER
      A0=0.9379388841
      A1=1.194191634
      A2=0.588245154
      A3=0.570337193
      A4=-1.5791166
      A5=4.292469
      B0=0.7276787064
      B1=0.9254690857
      B2=0.4741520763
      B3=0.250820355
      B4=-0.025930075
      B5=0.055707999
      C0=0.4166740874
      C1=0.5295655111
      C2=0.2754273045
      C3=0.1283775092
      C4=0.0119191487
      C5=0.0130209543
      D0=0.2215940159
      D1=-0.09388379097
      D2=0.0147382145
      D3=-0.000857650032
      E0=0.2826723681
      E1=0.2356320335
      F2=0.06340205186
      E3=0.01360032364
```

```
      F0=1.012074180
      F1=-0.000325432
      F2=-1.1646323
      F3=1.3873864
      F4=-4.4655208
      X2=X*X
      X3=X2*X
      X4=X3*X
      X5=X4*X
      IF(X)10,20,20
   10 PRINT 15
   15 FORMAT(1H0,*X IS LESS THAN ZERO,TRY AGAIN*)
      RETURN
   20 IF(X-0.1)30,40,40
   30 SUM=A0+A1*X+A2*X2+A3*X3+A4*X4+A5*X5
      BIC3=0.7366554521/SUM
      RETURN
   40 IF(X-0.4)50,60,60
   50 SUM=B0+B1*X+B2*X2+B3*X3+B4*X4+B5*X5
      BIC3=0.5714977571/SUM
      RETURN
   60 IF(X-1.0)70,80,80
   70 SUM=C0+C1*X+C2*X2+C3*X3+C4*X4+C5*X5
      BIC3=0.3272473766/SUM
      RETURN
   80 IF(X-2.5)90,100,100
   90 BIC3=(D0+D1*X+D2*X2+D3*X3)/(E0+E1*X+E2*X2+E3*X3)
      RETURN
  100 Y=1./(X+3.25)
      SUM=F0+F1*X+F2*X2+F3*X3+F4*X4
      BIC3=1.268445824*EXP(-X)/(Y**0.5*SUM)
      RETURN
      END
```

REFERENCES

1. B. J. Toppel, A. L. Rago, and D. M. O'Shea, "MC^2, A Code to Calculate Multigroup Cross Sections," Argonne National Laboratory report ANL-7318 (1967).

2. O. Ozer, Ed., "Description of the ENDF/B Processing Codes and Retrieval Subroutines," Brookhaven National Laboratory report BNL-50300 (ENDF-11) (1971).

3. D. M. Green and T. A. Pitterle, "ETOE, A Program for ENDF/B to MC^2 Data Conversion," Atomic Power Development Associates, Inc. report APDA-219 (ENDF-120) (1968).

4. W. W. Clendenin, "Calculation of Thermal Neutron Diffusion Length and Group Cross Sections: The GLEN Program," Los Alamos Scientific Laboratory report LA-3893 (1968).

5. H. C. Honeck and D. R. Finch, "FLANGE-II (Version 71-1), A Code to Process Thermal Neutron Data from an ENDF/B Tape," Savannah River Research Laboratory report DP-1278 (1971).

6. J. U. Koppel, T. R. Triplett, and Y. D. Naliboff, "GASKET, A Unified Code for Thermal Neutron Scattering," General Atomic report GA-7417 (1967).

7. W. W. Clendenin, "Calculation of Thermal Neutron Scattering Cross Sections for Crystalline Materials: The TOR Program," Los Alamos Scientific Laboratory report LA-3823 (1967).

8. Y. D. Naliboff and J. U. Koppel, "HEXSCAT, Coherent Elastic Scattering of Neutrons by Hexagonal Lattices," General Atomic report GA-6026 (1964).

9. K. D. Lathrop, "DTF-IV, A FORTRAN-IV Program for Solving the Multigroup Transport Equation with Anisotropic Scattering," Los Alamos Scientific Laboratory report LA-3373 (1965).

10. P. Wälti and P. Koch, "MICROX, A Two-Region Flux Spectrum Code for the Efficient Calculation of Group Cross Sections," General Atomic report GA-A10827 (1972).

11. C. R. Weisbin, P. D. Soran, R. E. MacFarlane, D. R. Harris, R. J. LaBauve, J. S. Hendricks, J. E. White, and R. B. Kidman, "MINX, A Multigroup Interpretation of Nuclear X-Sections from ENDF/B," Los Alamos Scientific Laboratory report LA-6486-MS (ENDF-237) (1976).

12. I. I. Bondarenko, Ed., Group Constants for Nuclear Reactor Calculations (Consultants Bureau, New York, 1964).

13. P. Wälti, "Evaluation of Grain Shielding Factors for Coated Fuel Particles," Nucl. Sci. Eng. 45, 321 (1971).

14. R. W. Hardie and W. W. Little, Jr., "1DX, A One-Dimensional Diffusion Code for Generating Effective Nuclear Cross Sections," Battelle-Northwest Laboratory report BNWL-954 (1969).

15. M. M. Levine, "Resonance Integral Calculations for ^{238}U Lattices," Nucl. Sci. Eng. 16, 271 (1963).

16. R. E. MacFarlane and R. B. Kidman, "LINX and BINX: CCCC Utility Codes for the MINX Multigroup Processing Code," Los Alamos Scientific Laboratory report LA-6219-MS (1976).

17. R. B. Kidman and R. E. MacFarlane, "CINX: Collapsed Interpretation of Nuclear X-Sections," Los Alamos Scientific Laboratory report LA-6287-MS (1976).

18. C. A. Stevens and C. V. Smith, "GAROL, A Computer Program for Evaluating Resonance Overlap," General Atomic report GA-6637 (1965).

19. D. R. Mathews, P. K. Koch, J. Adir, and P. Walti, "GGC-5, A Computer Program for Calculating Neutron Spectra and Group Constants," General Atomic report GA-8871 (1971).

20. K. M. Case, F. DeHoffman, and G. Placzek, Introduction to the Theory of Neutron Diffusion (U. S. Government Printing Office, Washington, D.C. 1953).

21. I. Carlvik and B. Pershagen, "The Dancoff Correction in Various Geometries," Aktiebolaget Atomenergi report AE-16 (1959).

22. I. Carlvik, "Dancoff Correction in Square and Hexagonal Lattices," Aktiebolaget Atomenergi report AE-257 (1966).

23. E. P. Wigner, E. Creutz, H. Jupnik, and T. Snyder, "Resonance Absorption of Neutrons by Spheres," J. Appl. Phys. 26, 260 (1955).

24. J. M. Otter, "Escape Probability Approximations in Lumped Resonance Absorbers," Atomics International report NAA-SR-9744 (1964).

25. G. I. Bell,"Theory of Effective Cross Sections," Los Alamos Scientific Laboratory report LA-2322 (1959).

26. L. Dresner, Resonance Absorption in Nuclear Reactors (Pergamon Press, New York, 1960).

27. G. I. Bell and S. Glasstone, Nuclear Reactor Theory (Van Nostrand Reinhold Company, New York 1970).

28. M. G. Stamatelatos, "Cross-Section Space Shielding in Doubly Heterogeneous HTGR Systems," Los Alamos Scientific Laboratory report LA-6157-MS (1975).

29. M. G. Stamatelatos, "Rational Approximations for Cross-Section Space-Shielding in Doubly Heterogeneous Systems," Nucl. Sci. Eng. 61, 543 (1976).

30. A. Sauer, "Approximate Escape Probabilities," Nucl. Sci. Eng. 16, 329 (1963).

31. R. Bonalumi, "Systematic Approximations of Neutron Fast-Collision Probabilities," Energia Nucleare 12, 1 (1965).

☆ U.S. GOVERNMENT PRINTING OFFICE 1977—777-018/24