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

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

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-

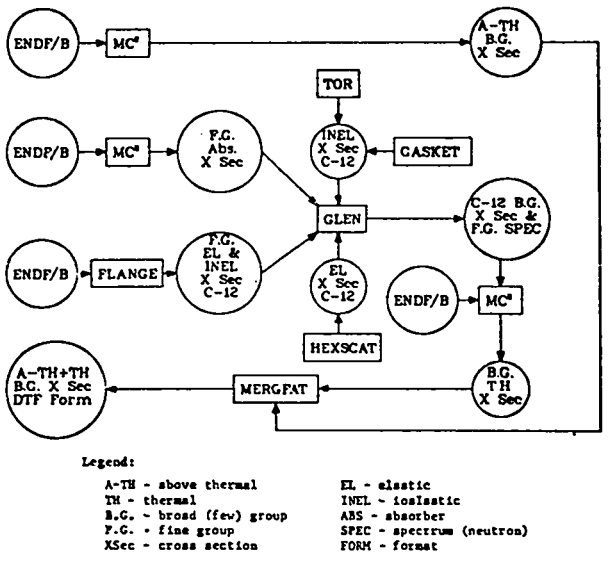

Fig. 1.
Initial data flow systems (several options are shown).

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 Δu = 1/120. Since the second option was found to be too time-consuming and costly without the benefit of considerable increase in

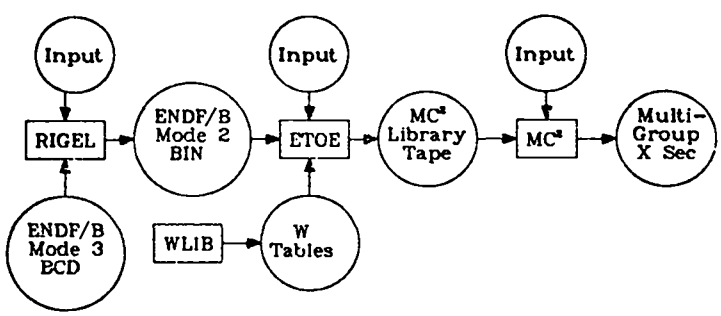

Fig. 2.
MC^2-I library preparation.

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:

$$\left(\overline{\sigma}^{bg}_{c_J}\right)_{\text{resolved}} = \frac{\sum_{j \text{ in } J} \langle \sigma \rangle^{fg}_{j \text{ resolved}} Q_j}{\sum_{j \text{ in } J} Q_j} \quad , \tag{1}$$

where

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

and

$$S_j^{fg} = \sum_{k \neq j} \Sigma_{inel,k}^{fg} \phi_k^{fg} \frac{\langle \Sigma_{inel} \rangle_{k \to j}}{\langle \Sigma_{inel} \rangle_k} + \sum_{k \neq j} \Sigma_{n,2n,k}^{fg} \phi_k^{fg} \frac{2\langle \Sigma_{n,2n} \rangle_{k \to j}}{\langle \Sigma_{n,2n} \rangle_k}$$

$$+ \langle \Sigma_{el} \rangle_{j-1 \to j} \phi_{j-1}^{fg} \quad , \tag{3}$$

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.

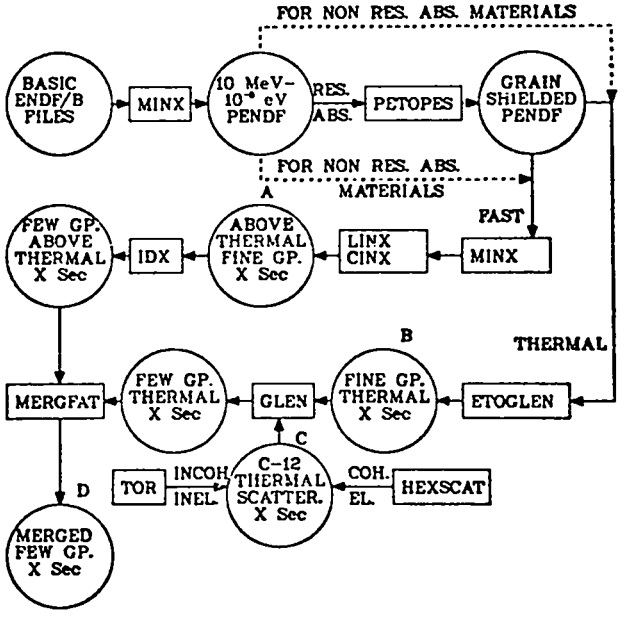

Fig. 3.
Final data flow system.

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-probabil-

ity 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-

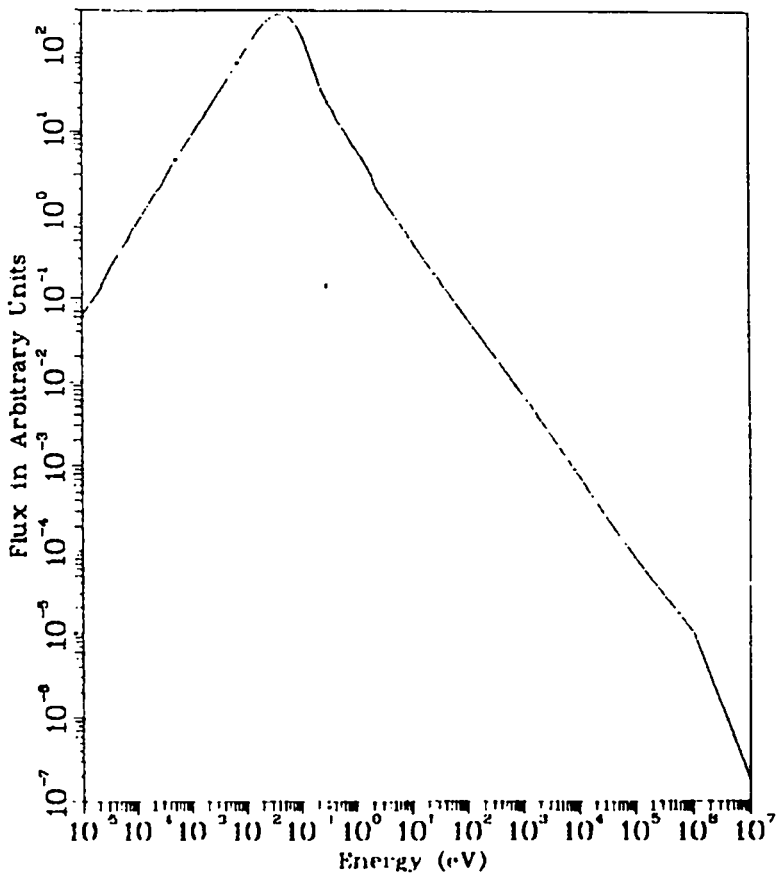

Fig. 4.
Typical HTGR spectrum used for MINX weight function.

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

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

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

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

where

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

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

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

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

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

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

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

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

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

where

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

The escape probability function $\bar{P}_0$ 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

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

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

$\overline{\ell}_F$ = 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

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

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

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

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,

$\bar{\ell}_0 = \frac{4V_0}{S_0}$ = 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:

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

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:

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

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

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

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.

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

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

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

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

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

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.

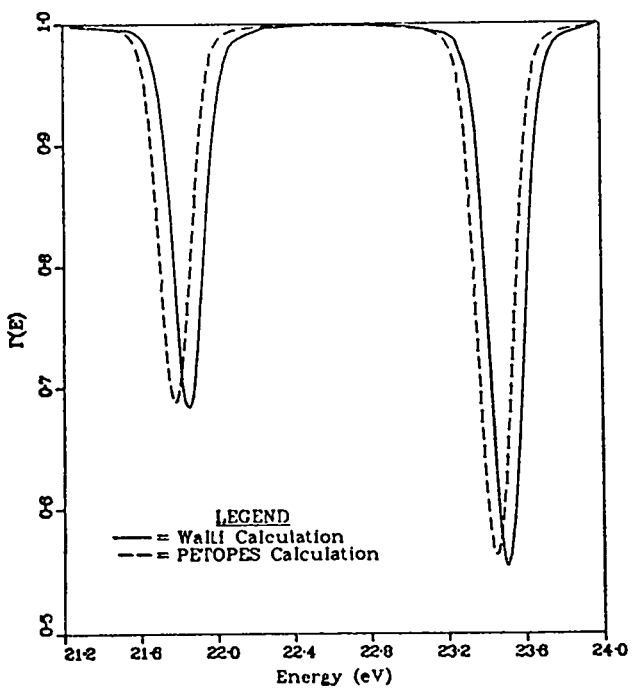

Fig. A-1.
Comparison of Wälti and PETOPES calculations for $\Gamma(E)$ for the 21.8-and 23.5-eV resonances of ^{232}Th at 300 K.

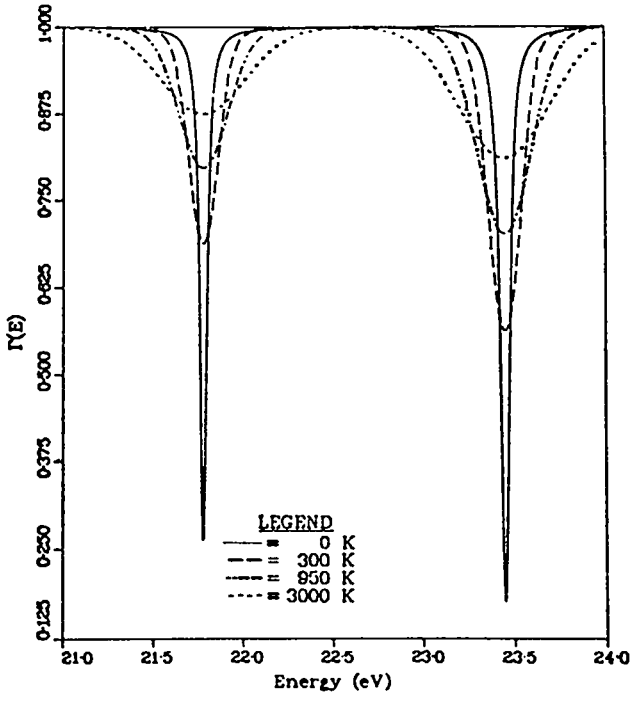

Fig. A-2.
$\Gamma(E)$ for 0, 300, 950, and 3000 K for the 21.8-and 23.5-eV resonances of ^{232}Th.

LASL Identification No. LP-0755

```
      PROGRAM PETOPES (INP,OUT,FSET5=INP,FSET6=OUT,FSET10,FSET11,FSET12,PETOP  1
     1    FILM,FSET9)                                                    PETOP  2
C     PURPOSE OF PROGRAM - TO CONVERT A PENDF TAPE TO SHIELDED PENDF.    PETOP  3
C     PENDF TO PENDF-SHIELDED.                                           PETOP  4
C     --    -- --    -                                                   PETOP  5
      LCM/XSECTT/XT(60000),YT(60000),NPTT                                PETOP  6
      LCM/XSECTE/XE(60000),YE(60000),NPEE                                PETOP  7
      COMMON/CONS/RADC,RADP,VOLC,VOLP,SURC,SURP,PSIP(10),PSIC(10),EMAX,  PETOP  8
     1    EMIN,MT                                                        PETOP  9
      COMMON/CALC/H1TAU,VOLF,Q,SEEP,SEEC,SIGPSP,SIGPSC,TAUTC,TAUXC       PETOP 10
      COMMON/CON1/CONP(10),CONC(10),XSC(9),XSP(9),NMP,NMC,XP(9),XC(9)    PETOP 11
      COMMON/PLOTS/ENG(5000),FAX(5000),GAMX(5000),NX,TITL(5),XLB(5),     PETOP 12
     1    YLB(5)                                                         PETOP 13
      DIMENSION F(10),S(10),J(10),A(8),HOL(7),X(10),Y(10)                PETOP 14
    4 READ (11,15) (A(I),I=1,6),ANEXT1,MCHECK,ANEXT2                     PETOP 15
    5 WRITE (10,15) (A(I),I=1,6),ANEXT1,MCHECK,ANEXT2                    PETOP 16
      IF (MCHECK.EQ.4H  -1)  GO TO 6                                     PETOP 17
      GO TO 4                                                            PETOP 18
    6 END FILE 10                                                        PETOP 19
      REWIND 10                                                          PETOP 20
      REWIND 11                                                          PETOP 21
   15 FORMAT (6A10,A6,A4,A10)                                            PETOP 22
C                                                                        PETOP 23
C     INPUT DEFINITIONS-                                                 PETOP 24
C     RADP - RADIUS OF PARTICLE,E.G. THORIUM CORE  OF THORIUM COATED     PETOP 25
C          PARTICLE IN HTGR                                              PETOP 26
C     RADC - RADIUS OF EFFECTIVE SPHERICAL SHELL,E.G. RADIUS OF EFFECTIVPETOP 27
C          MEDIA SURROUNDING THORIUM CORE IN HTGR FUEL ELEMENT.          PETOP 28
C     VOLP-PARTICLE VOLUME CORRESPONDING TO RADP.                        PETOP 29
C     VOLC-VOLUME CORRESPONDING TO MEDIA SURROUNDING PARTICLE REGION.    PETOP 30
C     SURP-SURFACE AREA OF PARTICLE.                                     PETOP 31
C     SURC-SURFACE AREA OF SURROUNDING MEDIA.                            PETOP 32
C     PSIP- LOG-DEC (MT252) FOR MATERIALS IN PARTICLE REGION,E.G.        PETOP 33
C           FOR THORIUM PSIP= 0.008669.                                  PETOP 34
C     PSIC- -LOG-DEC (MT252) FOR MATERIALS OUTSIDE,PARTICLE REGION. FOR  PETOP 35
C           CARBON,PSIC= 0.1589.                                         PETOP 36
C     NMP=NO OF MATS IN PARTICLE REGION.                                 PETOP 37
C     NMC=NO OF MATS IN OUTER REGION.                                    PETOP 38
C     CONP-ATOMS/CC OF MATS IN PARTICLE REGION.  CONP(1) IS FOR THORIUM  PETOP 39
C     CONC-ATOMS/CC OF MATS OUTSIDE PARTICLE REGION.                     PETOP 40
C     XSP,XP-TOT,POT XSEC FOR MATERIALS WITH CONSTANT XSEC IN PARTICLE   PETOP 41
C          REGION.  XSP(1),XP(1),ARE FOR THORIUM-COMPUTED IN GRANSHL.    PETOP 42
C     XSC,XC-TOT,POT XSEC FOR MATERIALS OUTSIDE PARTICLE REGION.         PETOP 43
C     EMAX-ENERGY BOUNDING RESONANCE REGION FOR PARTICLE SHIELDING,E.G.  PETOP 44
C          EMAX=4.0 KEV FOR THORIUM.                                     PETOP 45
C     EMIN- LOWER BOUND OF RESONANCE REGION,E.G.,EMIN=21EV FOR TH-232.   PETOP 46
C     NOQCAL = 0 FOR FERTILE MATS,E.G. THORIUM IN RES. REGION.           PETOP 47
C            = 1 FOR FISSILE MATS,E.G. U-235 AND U-233 IN THERMAL        PETOP 48
C              REGION ONLY.(NOTE THERMAL REGION MUST BE RUN SOLO         PETOP 49
C              BECAUSE OF THIS)                                          PETOP 50
      READ (5,15) (A(I),I=1,6)                                           PETOP 51
      WRITE (6,15) (A(I),I=1,6)                                          PETOP 52
      READ (5,18) RADP,RADC,EMAX,EMIN                                    PETOP 53
    7 FORMAT (1H0,*  INPUT*//*     RADP = *1PE12.5,* RADC = *1PE12.5,    PETOP 54
     1    * EMAX = *1PE12.5,* EMIN = *1PE12.5,* NOQCAL = *I3)            PETOP 55
   18 FORMAT (6E11.4)                                                    PETOP 56
      READ (5,19) NMP,NMC,NOQCAL                                         PETOP 57
      WRITE (6,7) RADP,RADC,EMAX,EMIN,NOQCAL                             PETOP 58
   19 FORMAT (6I11)                                                      PETOP 59
      READ (5,18) (PSIP(I),I=1,NMP)                                      PETOP 60
      READ (5,18) (PSIC(I),I=1,NMC)                                      PETOP 61
      READ (5,18) (CONP(I),I=1,NMP)                                      PETOP 62
```

```
      READ (5,18) (CONC(I),I=1,NMC)                                    PETOP 63
      READ (5,18) (XSP(I),XP(I),I=1,NMP)                               PETOP 64
      READ (5,18) (XSC(I),XC(I),I=1,NMC)                               PETOP 65
C                                                                      PETOP 66
C     THESE CONSTANTS ARE NEEDED IN SUBROUTINE GRANSHL.                PETOP 67
C                                                                      PETOP 68
      VOLP=4./3.*3.14159*RADP**3                                       PETOP 69
      VOLC=4./3.*3.14159*RADC**3 -VOLP                                 PETOP 70
      SURP=4.*3.14159*RADP**2                                          PETOP 71
      SURC=SURP                                                        PETOP 72
      R=RADP/RADC                                                      PETOP 73
      GAM=3*R**2/(4.0*(1.0-R**3))                                      PETOP 74
      TRM1=(1.0-R**2)**2*(1.0+0.25*ALOG((1.0+R)/(1.0-R)))-             PETOP 75
     1     0.5*R*(1.0-R)**2                                            PETOP 76
      TRM2=(2.0/(3.0*R))**2                                            PETOP 77
      TRM3=(1.0-R**2)**3-3.0*(1.0-R**3)**2+2.0*(1.0-R**3)*(1.0-R**2)**1.PETOP 78
     1 5                                                               PETOP 79
      HITAU=(GAM/R)**2*(TRM1+TRM2*TRM3)                                PETOP 80
      VOLF=VOLC+VOLP                                                   PETOP 81
C                                                                      PETOP 82
C     CALCULATE Q.                                                     PETOP 83
C                                                                      PETOP 84
      SEENUM=0.                                                        PETOP 85
       SIGPSP=0.                                                       PETOP 86
      DO 130 I=1,NMP                                                   PETOP 87
      SEENUM=PSIP(I)*CONP(I)*XP(I)+SEENUM                              PETOP 88
      SIGPSP=CONP(I)*XP(I)+SIGPSP                                      PETOP 89
  130 CONTINUE                                                         PETOP 90
      Q=SEENUM                                                         PETOP 91
      SEEP=SEENUM/SIGPSP                                               PETOP 92
      SEENUM=0.                                                        PETOP 93
      SIGPSC=0.                                                        PETOP 94
      SIGTC=0.                                                         PETOP 95
      DO 140 I=1,NMC                                                   PETOP 96
      SEENUM=PSIC(I)*CONC(I)*XC(I)+SEENUM                              PETOP 97
      SIGPSC=CONC(I)*XC(I)+SIGPSC                                      PETOP 98
      SIGTC=CONC(I)*XSC(I)+SIGTC                                       PETOP 99
  140 CONTINUE                                                         PETOP100
      SEEC=SEENUM/SIGPSC                                               PETOP101
      Q=Q/SEENUM                                                       PETOP102
      TAUTC=4.0*VOLC*SIGTC/SURC                                        PETOP103
      TAUXC=TAUTC                                                      PETOP104
C                                                                      PETOP105
C     END OF Q CALCULATION                                             PETOP106
C                                                                      PETOP107
      TITL(1)=10HGAMMA PLOT                                            PETOP108
      TITL(2)=10H TO COMPAR                                            PETOP109
      TITL(3)=10HE WITH OTH                                            PETOP110
      TITL(4)=10HR METHODS.                                            PETOP111
      XLB(1)=10HENERGY IN                                              PETOP112
      XLB(2)=10HE.V. UNITS                                             PETOP113
   10 READ (11,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ                       PETOP114
      HOL(1)=10H  THIS TAP                                             PETOP115
      HOL(2)=10HF HAS BEEN                                             PETOP116
      HOL(3)=10H CHANGED T                                             PETOP117
      HOL(4)=10HO A PENDF-                                             PETOP118
      HOL(5)=10HSHIELDED F                                             PETOP119
      HOL(6)=10HILE.                                                   PETOP120
   20 FORMAT (6A10,A6,I4,I2,I3,I5)                                     PETOP121
      WRITE (12,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ                      PETOP122
      READ (10,20) DUM                                                 PETOP123
      READ (10,80) ZA,AWR                                              PETOP124
      CALL STORXS                                                      PETOP125
```

```
      NX=0                                                              PETOP126
      PRINT 2020,MAT                                                    PETOP127
 2020 FORMAT (1H1,*WELL,WE MADE IT OUT OF STORXS ONCE,  MAT=*I4)        PETOP128
   30 READ (11,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ                       PETOP129
      IF (MAT.EQ.0)  CALL STORXS                                        PETOP130
      IF (MAT.EQ.0) NX=0                                                PETOP131
 2030 FORMAT (1H ,*WE ARE LOOPING NOW,  MAT=*I4)                        PETOP132
      WRITE (12,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ                       PETOP133
      IF (MAT.EQ.-1) GO TO 2000                                         PETOP134
      IF (MF.NE.3) GO TO 30                                             PETOP135
      IF (MT.EQ.1)  GO TO 31                                            PETOP136
      IF (MT.EQ.2)  GO TO 31                                            PETOP137
      IF (MT.EQ.3)  GO TO 31                                            PETOP138
      IF (MT.EQ.18)  GO TO 31                                           PETOP139
      IF (MT.EQ.102)  GO TO 31                                          PETOP140
      GO TO 30                                                          PETOP141
   31 CONTINUE                                                          PETOP142
      MTXX=MT                                                           PETOP143
      READ (11,40) C1,C2,N1,N2,N3,N4,MAT,MF,MT,NSEQ                     PETOP144
      CALL CXFP (C1,F(1),S(1),J(1))                                     PETOP145
      CALL CXFP (C2,F(2),S(2),J(2))                                     PETOP146
      WRITE (12,50) (F(I),S(I),J(I),I=1,2),N1,N2,N3,N4,MAT,MF,MT,NSEQ   PETOP147
   40 FORMAT (1P2E11.4,4I11,I4,I2,I3,I5)                                PETOP148
   50 FORMAT (2(F8.5,A1,I2),4I11,I4,I2,I3,I5)                           PETOP149
      READ (11,60) NPT,INT,N0,N0,N0,N0,MAT,MF,MT,NSEQ                   PETOP150
      WRITE (12,60) NPT,INT,N0,N0,N0,N0,MAT,MF,MT,NSEQ                  PETOP151
      WRITE (6,100) C1,MAT,MT                                           PETOP152
   60 FORMAT (6I11,I4,I2,I3,I5)                                         PETOP153
      NN1=1                                                             PETOP154
   70 NN2=NN1+2                                                         PETOP155
      READ (11,80) (X(I),Y(I),I=1,3),MAT,MF,MT,NSEQ                     PETOP156
   80 FORMAT (1P6E11.4,I4,I2,I3,I5)                                     PETOP157
      LOOP=0                                                            PETOP158
      DO 85 I=1,3                                                       PETOP159
      E=X(I)                                                            PETOP160
      CALL GRANSHL (E,FACT)                                             PETOP161
      Y(I)=Y(I)*FACT                                                    PETOP162
      LOOP=LOOP+1                                                       PETOP163
   85 CONTINUE                                                          PETOP164
      CALL CXFP (X(1),F(1),S(1),J(1))                                   PETOP165
      CALL CXFP (Y(1),F(2),S(2),J(2))                                   PETOP166
      CALL CXFP (X(2),F(3),S(3),J(3))                                   PETOP167
      CALL CXFP (Y(2),F(4),S(4),J(4))                                   PETOP168
      CALL CXFP (X(3),F(5),S(5),J(5))                                   PETOP169
      CALL CXFP (Y(3),F(6),S(6),J(6))                                   PETOP170
      WRITE(12,90) (F(I),S(I),J(I),I=1,6),MAT,MF,MT,NSEQ                PETOP171
   90 FORMAT (6(F8.5,A1,I2),I4,I2,I3,I5)                                PETOP172
   95 FORMAT (*  M = *I6,*  E = *1PE12.5,*  FACT = *1PE12.5)            PETOP173
  100 FORMAT (1H1,*  TEMPERATURE = *1PE12.5,*  MAT = *I4,*  MT = *I3)   PETOP174
      NN1=NN2+1                                                         PETOP175
      IF (NN1.LE.N4) GO TO 70                                           PETOP176
      READ (11,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ                        PETOP177
      WRITE (12,20) (HOL(I),I=1,7),MAT,MF,MT,NSEQ                       PETOP178
      IF (MTXX.GT.1)  GO TO 30                                          PETOP179
      WRITE (6,200) NX,C1                                               PETOP180
  200 FORMAT (1H1,*  NX= *I6,* FOR TEMP = *1PE12.5//7X,*ENERGY*,15X,    PETOP181
     1    *FACT*13X,*GAMMA*)                                            PETOP182
      WRITE (6,210) (ENG(N),FAX(N),GAMX(N),N=1,NX)                      PETOP183
  210 FORMAT (1P3E18.5)                                                 PETOP184
      WRITE (9) NX,(ENG(N),FAX(N),GAMX(N),N=1,NX),NX                    PETOP185
      GO TO 30                                                          PETOP186
 2000 WRITE (6,2010) MAT                                                PETOP187
 2010 FORMAT (1H1,*  PROCESSING COMPLETE,  MAT = *I4)                   PETOP188
```

```
      CALL GFPLT                                                        PETOPI89
      END                                                               PETOPI90


      SUBROUTINE STORXS                                                 STORX  1
C     STORE TOTAL AND ELASTIC XSEC FOR THORIUM.                         STORX  2
      LCM/XSECTT/XT(60000),YT(60000),NPTT                               STORX  3
      LCM/XSECTE/XE(60000),YE(60000),NPEE                               STORX  4
      DIMENSION A(7)                                                    STORX  5
   10 READ (10,20) (A(I),I=1,7),MAT,MF,MT,NSEQ                          STORX  6
   20 FORMAT (6A10,A6,I4,I2,I3,I5)                                      STORX  7
      IF (MAT.EQ.-1) GO TO 2000                                         STORX  8
      IF (MF.GT.3)  GO TO 300                                           STORX  9
      IF (MF.LT.3) GO TO 10                                             STORX 10
      IF (MT.EQ.1)  GO TO 100                                           STORX 11
      IF (MT.EQ.2)  GO TO 200                                           STORX 12
      GO TO 10                                                          STORX 13
  100 READ (10,30) NPTT                                                 STORX 14
      PRINT 2020,NPTT                                                   STORX 15
      READ (10,20) (A(I),I=1,7)                                         STORX 16
      IF (NPTT.GT.60000) GO TO 130                                      STORX 17
      NPTTS=NPTT                                                        STORX 18
  105 READ (10,40) (XT(I),YT(I),I=1,NPTTS)                              STORX 19
      PRINT 40,(XT(I),YT(I),I=1,99)                                     STORX 20
      PRINT 2040,NPTTS,XT(NPTTS),YT(NPTTS)                              STORX 21
  110 IF (NPTT.EQ.NPTTS)  GO TO 10                                      STORX 22
  120 IF (MT.EQ.0)  GO TO 10                                            STORX 23
      READ (10,20)(A(I),I=1,7),MAT,MF,MT                                STORX 24
      GO TO 120                                                         STORX 25
  130 NPTTS=60000                                                       STORX 26
      GO TO 105                                                         STORX 27
  200 READ (10,30) NPEE                                                 STORX 28
      PRINT 2030,NPEE                                                   STORX 29
      READ (10,20) (A(I),I=1,7)                                         STORX 30
      IF (NPEE.GT.60000) GO TO 230                                      STORX 31
      NPEES=NPEE                                                        STORX 32
  205 READ (10,40)(XE(I),YE(I),I=1,NPEES)                               STORX 33
      PRINT 40,(XE(I),YE(I),I=1,99)                                     STORX 34
      PRINT 2050,NPEES,XE(NPEES),YE(NPEES)                              STORX 35
  210 IF (NPEE.EQ.NPEES)  GO TO 10                                      STORX 36
  220 IF (MT.EQ.0)  GO TO 10                                            STORX 37
      READ (10,20)(A(I),I=1,7),MAT,MF,MT                                STORX 38
      GO TO 220                                                         STORX 39
  230 NPEES=60000                                                       STORX 40
      GO TO 205                                                         STORX 41
  300 READ (10,20) (A(I),I=1,7),MAT,MF,MT,NSEQ                          STORX 42
      IF (MAT.EQ.-1)  GO TO 2000                                        STORX 43
      IF (MAT.NE.0)  GO TO 300                                          STORX 44
      RETURN                                                            STORX 45
   30 FORMAT (55X,I11)                                                  STORX 46
   40 FORMAT (6E11.4)                                                   STORX 47
 2000 WRITE(6,2010) MAT                                                 STORX 48
 2010 FORMAT (1H1,*  SORRY TAPE IS OUT OF TEMPS. MAT=*I4)               STORX 49
 2020 FORMAT (1H1,10X,*XT,YT TABLE*10X,*NPTT=*I11)                      STORX 50
 2030 FORMAT (1H0,10X,*XE,YE TABLE*10X,*NPEE=*I11)                      STORX 51
 2040 FORMAT (1H ,*NPTTS=*I6,4X,*XT(NPTTS)=*,E11.4,4X,*YT(NPTTS)=*E11.4)STORX 52
 2050 FORMAT (1H ,*NPEES=*I6,4X,*XE(NPEES)=*,E11.4,4X,*YE(NPEES)=*E11.4)STORX 53
      RETURN                                                            STORX 54
      END                                                               STORX 55
```

```
      SUBROUTINE GRANSHL (E,FACT)                                        GRANS  1
C     PURPOSE = TO CALCULATE SHIELDING FACTOR FOR TWO REGION PARTICLE.   GRANS  2
C                                                                        GRANS  3
      LCM/XSECTT/XT(60000),YT(60000),NPTT                                GRANS  4
      LCM/XSECTE/XE(60000),YE(60000),NPEE                                GRANS  5
      COMMON/CONS/RADC,RADP,VOLC,VOLP,SURC,SURP,PSIP(10),PSIC(10),EMAX,  GRANS  6
     1    EMIN,MT                                                        GRANS  7
      COMMON/CALC/H1TAU,VOLF,Q,SEEP,SEEC,SIGPSP,SIGPSC,TAUTC,TAUXC       GRANS  8
      COMMON/CON1/CONP(10),CONC(10),XSC(9),XSP(9),NMP,NMC,XP(9),XC(9)    GRANS  9
      COMMON/PLOTS/ENG(5000),FAX(5000),GAMX(5000),NX,TITL(5),XLB(5),     GRANS 10
     1    YLB(5)                                                         GRANS 11
C                                                                        GRANS 12
C     CONDITIONAL RETURNS                                                GRANS 13
C                                                                        GRANS 14
      IF (E.GT.1.0E-10) GO TO 10                                         GRANS 15
      E=0.                                                               GRANS 16
      FACT=0.                                                            GRANS 17
      RETURN                                                             GRANS 18
   10 CONTINUE                                                           GRANS 19
      IF (E.LT.EMAX) GO TO 20                                            GRANS 20
      FACT=1.0                                                           GRANS 21
      RETURN                                                             GRANS 22
   20 CONTINUE                                                           GRANS 23
      QQ=Q                                                               GRANS 24
      IF (E.LT.EMIN) QQ=0.                                               GRANS 25
C                                                                        GRANS 26
C     FIND TOTAL AND ELASTIC CROSS SECTIONS CORRESPONDING TO ENERGY E.   GRANS 27
C                                                                        GRANS 28
      CALL LOCT1(E,ILK,ILOT)                                             GRANS 29
      IHIT=ILOT+1                                                        GRANS 30
      CALL LOCT2(E,ILK,ILOE)                                             GRANS 31
      IHIE=ILOE+1                                                        GRANS 32
      DXTL=XT(ILOT)                                                      GRANS 33
      DYTL=YT(ILOT)                                                      GRANS 34
      DXTH=XT(IHIT)                                                      GRANS 35
      DYTH=YT(IHIT)                                                      GRANS 36
      CALL TERP1 (DXTL,DYTL,DXTH,DYTH,E,ST,2,1)                          GRANS 37
      DXEL=XE(ILOE)                                                      GRANS 38
      DYEL=YE(ILOE)                                                      GRANS 39
      DXEH=XE(IHIE)                                                      GRANS 40
      DYEH=YE(IHIE)                                                      GRANS 41
      CALL TERP1 (DXEL,DYEL,DXEH,DYEH,E,SE,2,2)                          GRANS 42
C                                                                        GRANS 43
      XP(1)=SE                                                           GRANS 44
      XSP(1)=ST                                                          GRANS 45
       SIGPSP1=0.                                                        GRANS 46
      SIGTP=0.                                                           GRANS 47
      SEENUM=0.                                                          GRANS 48
      DO 30 I=1,NMP                                                      GRANS 49
      SEENUM=PSIP(I)*CONP(I)*XP(I)+SEENUM                                GRANS 50
      SIGPSP1=CONP(I)*XP(I)+SIGPSP1                                      GRANS 51
      SIGTP=CONP(I)*XSP(I)+SIGTP                                         GRANS 52
   30 CONTINUE                                                           GRANS 53
      SEEP1=SEENUM/SIGPSP1                                               GRANS 54
      TAUTP=4.0*VOLP*SIGTP/SURP                                          GRANS 55
      TAUXP=TAUTP*(1.0-(1.0-SEEP1/SEEC)*SIGPSP1/SIGTP)                   GRANS 56
      X=3.0*TAUTP/4.0                                                    GRANS 57
      P0TAU=3.0/(8.0*X**3)*(2.0*X**2-1.0+(1.0+2.0*X)*EXP(-2.0*X))        GRANS 58
      H0TAU=(1.0-P0TAU)/(TAUTP*P0TAU)-1.0                                GRANS 59
      W=1.0+H0TAU+H1TAU                                                  GRANS 60
      RHOQ=VOLP/VOLC*QQ                                                  GRANS 61
      UPPER=1.0+RHOQ*(1.0+TAUXC*W)                                       GRANS 62
```

```
      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)                          TERP1   1
C     =====INTERPOLATE ONE PT.=====                                      TERP1   2
C     (X1,Y1) AND (X2,Y2) ARE END PTS. OF THE LINE                       TERP1   3
C     (X,Y) IS INTERPOLATED POINT                                        TERP1   4
C     I=INTERPOLATION CODE                                               TERP1   5
C     NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE    TERP1   6
C            INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.  TERP1   7
C     ERROR STOPS - 301 (X1=X2,DISCONTINUITY)                            TERP1   8
C                   302 (INTERPOLATION CODE IS OUT OF RANGE)             TERP1   9
C                   303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT.) TERP1  10
    5 XA=X1                                                              TERP1  11
      YA=Y1                                                              TERP1  12
      XB=X2                                                              TERP1  13
      YB=Y2                                                              TERP1  14
      XP=X                                                               TERP1  15
      II=I                                                               TERP1  16
      IF ((XB-XA).GT.1.E-10) GO TO 7                                     TERP1  17
      IF (X.EQ.XA) Y=YA                                                  TERP1  18
      PRINT 6,XA,YA,XB,YB,X,Y,I,NERR                                     TERP1  19
    6 FORMAT (1H0,*  ERROR STOP 301 *1P6E12.5,2I3)                       TERP1  20
      RETURN                                                             TERP1  21
    7 CONTINUE                                                           TERP1  22
      IF (II) 10,10,15                                                   TERP1  23
   10 CALL ERROR (302)                                                   TERP1  24
   15 IF (II-5) 20,20,10                                                 TERP1  25
   20 GO TO (25,30,35,60,75), II                                         TERP1  26
   25 YP=YA                                                              TERP1  27
      IF (XP.EQ.XB)  YP=YB                                               TERP1  28
      GO TO 105                                                          TERP1  29
   30 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)                                      TERP1  30
      GO TO 105                                                          TERP1  31
   35 IF (XA) 30,30,40                                                   TERP1  32
   40 IF (XB) 30,30,45                                                   TERP1  33
   45 IF (XP) 50,50,55                                                   TERP1  34
   50 CALL ERROR (303)                                                   TERP1  35
   55 YP=YA+ALOG(XP/XA)*(YB-YA)/ALOG(XB/XA)                              TERP1  36
      GO TO 105                                                          TERP1  37
   60 IF (YA) 30,30,65                                                   TERP1  38
   65 IF (YB) 30,30,70                                                   TERP1  39
   70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))                             TERP1  40
      GO TO 105                                                          TERP1  41
   75 IF (YA) 35,35,80                                                   TERP1  42
   80 IF (YB) 35,35,85                                                   TERP1  43
   85 IF (XA) 70,70,90                                                   TERP1  44
   90 IF (XB) 70,70,95                                                   TERP1  45
   95 IF (XP) 50,50,100                                                  TERP1  46
  100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))                     TERP1  47
  105 Y=YP                                                               TERP1  48
      RETURN                                                             TERP1  49
      END                                                                TERP1  50
```

```
      SUBROUTINE TERP1 (X1,Y1,X2,Y2,X,Y,I,NERR)                          TERP1  1
C     =====INTERPOLATE ONE PT.=====                                      TERP1  2
C     (X1,Y1) AND (X2,Y2) ARE END PTS. OF THE LINE                       TERP1  3
C     (X,Y) IS INTERPOLATED POINT                                        TERP1  4
C     I=INTERPOLATION CODE                                               TERP1  5
C     NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE    TERP1  6
C            INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.  TERP1  7
C     ERROR STOPS - 301 (X1=X2,DISCONTINUITY)                            TERP1  8
C                   302 (INTERPOLATION CODE IS OUT OF RANGE)             TERP1  9
C                   303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT.) TERP1 10
    5 XA=X1                                                              TERP1 11
      YA=Y1                                                              TERP1 12
      XB=X2                                                              TERP1 13
      YB=Y2                                                              TERP1 14
      XP=X                                                               TERP1 15
      II=I                                                               TERP1 16
      IF ((XB-XA).GT.1.E-10) GO TO 7                                     TERP1 17
      IF (X.EQ.XA) Y=YA                                                  TERP1 18
      PRINT 6,XA,YA,XB,YB,X,Y,I,NERR                                     TERP1 19
    6 FORMAT (1H0,*  ERROR STOP 301 *1P6E12.5,2I3)                       TERP1 20
      RETURN                                                             TERP1 21
    7 CONTINUE                                                           TERP1 22
      IF (II) 10,10,15                                                   TERP1 23
   10 CALL ERROR (302)                                                   TERP1 24
   15 IF (II-5) 20,20,10                                                 TERP1 25
   20 GO TO (25,30,35,60,75), II                                         TERP1 26
   25 YP=YA                                                              TERP1 27
      IF (XP.EQ.XB)  YP=YB                                               TERP1 28
      GO TO 105                                                          TERP1 29
   30 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)                                      TERP1 30
      GO TO 105                                                          TERP1 31
   35 IF (XA) 30,30,40                                                   TERP1 32
   40 IF (XB) 30,30,45                                                   TERP1 33
   45 IF (XP) 50,50,55                                                   TERP1 34
   50 CALL ERROR (303)                                                   TERP1 35
   55 YP=YA+ALOG(XP/XA)*(YB-YA)/ALOG(XB/XA)                              TERP1 36
      GO TO 105                                                          TERP1 37
   60 IF (YA) 30,30,65                                                   TERP1 38
   65 IF (YB) 30,30,70                                                   TERP1 39
   70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))                             TERP1 40
      GO TO 105                                                          TERP1 41
   75 IF (YA) 35,35,80                                                   TERP1 42
   80 IF (YB) 35,35,85                                                   TERP1 43
   85 IF (XA) 70,70,90                                                   TERP1 44
   90 IF (XB) 70,70,95                                                   TERP1 45
   95 IF (XP) 50,50,100                                                  TERP1 46
  100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))                     TERP1 47
  105 Y=YP                                                               TERP1 48
      RETURN                                                             TERP1 49
      END                                                                TERP1 50
```

```
      SUBROUTINE ERROR (N)                                               ERROR  1
      IOS=9                                                              ERROR  2
    5 PRINT 10,N                                                         ERROR  3
      WRITE(99,10)                                                       ERROR  4
   10 FORMAT (11H ERROR STOP,I6)                                         ERROR  5
      END                                                                ERROR  6
```

```
      SUBROUTINE LOCT1(X,ILO,LOCT)                                      LOCT1  1
C     BINARY SEARCH ROUTINE WRITTEN BY P. SORAN.  MODIFIED 10-30-73     LOCT1  2
C         TO GIVE RESULTS IDENTICAL TO EARLER LOCT ROUTINE.             LOCT1  3
C         THAT IS.  FIND X SUCH THAT A(LOCT+1).GT.X.GE.A(LOCT), EXCEPT WLOCT1  4
C         X IS EQUAL TO A(N).  IN THAT CASE, LOCT IS SET TO (N-1).      LOCT1  5
C         WHEN X IS NOT BINNABLE, THAT IS WHEN X IS OUTSIDE THE RANGE OFLOCT1  6
C         A-VALUES OR IF A CONTAINS ONLY A SINGLE POINT. THE VALUE LOCT=LOCT1  7
C         IS RETURNED.                                                  LOCT1  8
      LCM/XSECTT/A(60000),YT(60000),N                                   LOCT1  9
      IF(N.EQ.1) GO TO 3001                                             LOCT1 10
      IF(X.LT.A(1)) GO TO 3001                                          LOCT1 11
      IF(X.GT.A(N)) GO TO 3001                                          LOCT1 12
      IF((A(N-1).EQ.A(N)).AND.(X.EQ.A(N))) GO TO 3001                   LOCT1 13
      LOCT=1                                                            LOCT1 14
      IF(A(1).EQ.X) RETURN                                              LOCT1 15
      ILO=1                                                             LOCT1 16
      ISRCH=N                                                           LOCT1 17
1000  IF(ISRCH.LE.ILO+1)GO TO 3000                                      LOCT1 18
      I=(ISRCH+ILO)/2                                                   LOCT1 19
      IF(A(I).LT.X) GO TO 2000                                          LOCT1 20
      ISRCH=I                                                           LOCT1 21
      GO TO 1000                                                        LOCT1 22
2000  ILO=I                                                             LOCT1 23
      GO TO 1000                                                        LOCT1 24
C     X HAS BEEN BINNED.  CONVERT FROM ISRCH TO LOCT HERE.              LOCT1 25
 3000 IF(X.NE.A(ISRCH)) LOCT=ISRCH-1                                    LOCT1 26
      IF(X.EQ.A(ISRCH)) LOCT=ISRCH                                      LOCT1 27
      IF(X.EQ.A(N)) LOCT=N-1                                            LOCT1 28
      RETURN                                                            LOCT1 29
 3001 WRITE (99,10) LOCT                                                LOCT1 30
   10 FORMAT (1H ,I6)                                                   LOCT1 31
      RETURN                                                            LOCT1 32
      END                                                               LOCT1 33


      SUBROUTINE LOCT2(X,ILO,LOCT)                                      LOCT2  1
C     BINARY SEARCH ROUTINE WRITTEN BY P. SORAN.  MODIFIED 10-30-73     LOCT2  2
C         TO GIVE RESULTS IDENTICAL TO EARLER LOCT ROUTINE.             LOCT2  3
C         THAT IS.  FIND X SUCH THAT A(LOCT+1).GT.X.GE.A(LOCT), EXCEPT WLOCT2  4
C         X IS EQUAL TO A(N).  IN THAT CASE, LOCT IS SET TO (N-1).      LOCT2  5
C         WHEN X IS NOT BINNABLE. THAT IS WHEN X IS OUTSIDE THE RANGE OFLOCT2  6
C         A-VALUES OR IF A CONTAINS ONLY A SINGLE POINT. THE VALUE LOCT=LOCT2  7
C         IS RETURNED.                                                  LOCT2  8
      LCM/XSECTE/A(60000),XE(60000),N                                   LOCT2  9
      IF(N.EQ.1) GO TO 3001                                             LOCT2 10
      IF(X.LT.A(1)) GO TO 3001                                          LOCT2 11
      IF(X.GT.A(N)) GO TO 3001                                          LOCT2 12
      IF((A(N-1).EQ.A(N)).AND.(X.EQ.A(N))) GO TO 3001                   LOCT2 13
      LOCT=1                                                            LOCT2 14
      IF(A(1).EQ.X) RETURN                                              LOCT2 15
      ILO=1                                                             LOCT2 16
      ISRCH=N                                                           LOCT2 17
1000  IF(ISRCH.LE.ILO+1)GO TO 3000                                      LOCT2 18
      I=(ISRCH+ILO)/2                                                   LOCT2 19
      IF(A(I).LT.X) GO TO 2000                                          LOCT2 20
      ISRCH=I                                                           LOCT2 21
      GO TO 1000                                                        LOCT2 22
2000  ILO=I                                                             LOCT2 23
      GO TO 1000                                                        LOCT2 24
C     X HAS BEEN BINNED.  CONVERT FROM ISRCH TO LOCT HERE.              LOCT2 25
 3000 IF(X.NE.A(ISRCH)) LOCT=ISRCH-1                                    LOCT2 26
      IF(X.EQ.A(ISRCH)) LOCT=ISRCH                                      LOCT2 27
      IF(X.EQ.A(N)) LOCT=N-1                                            LOCT2 28
      RETURN                                                            LOCT2 29
```

```
 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

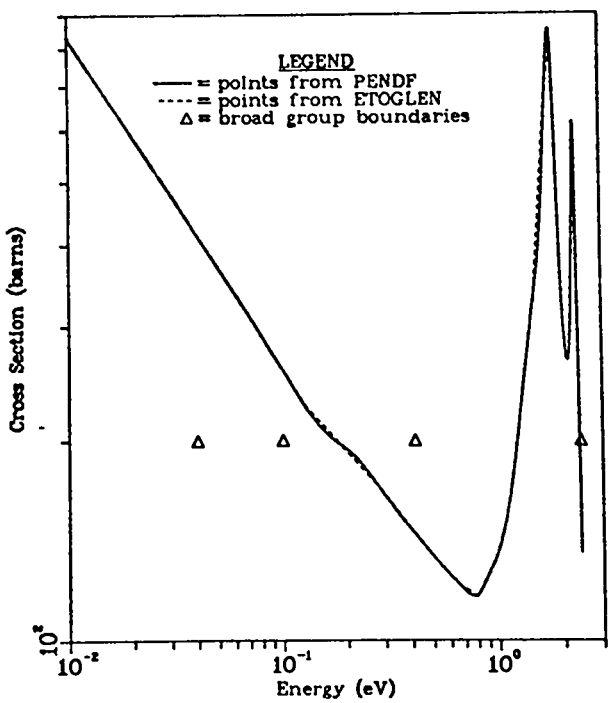

Fig. B-1.
^{233}U fission cross section at 300 K from 0.01 to 2.38 eV.

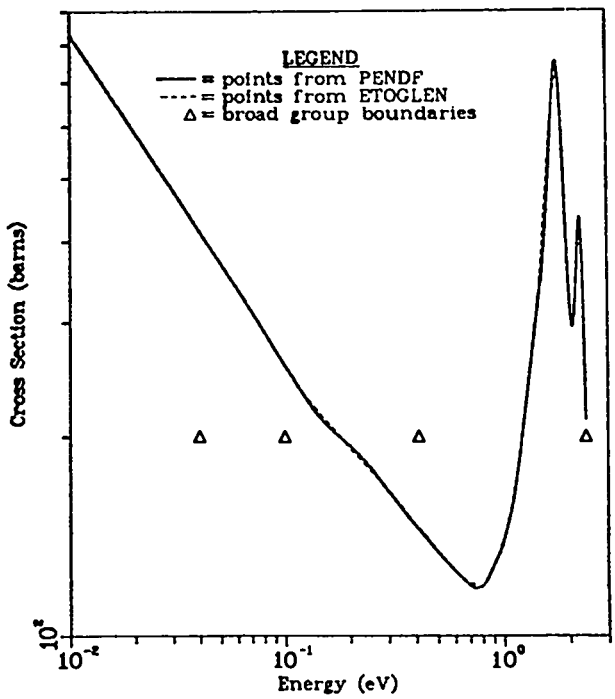

Fig. B-2.
^{233}U fission cross section at 3000 K from 0.01 to 2.38 eV.

LASL Identification No. LP-0756

```
      PROGRAM ETOGLEN(INP,OUT,PUN,FILM,FSET11,FSET12,FSET5=INP,          ETOGL  1
     1                FSET6=OUT)                                        ETOGL  2
C     PROGRAM TO GET GLEN INPUT CROSS SECTIONS FROM PENDF TAPE,I.E.      ETOGL  3
C     ENDF/B TO GLEN                                                    ETOGL  4
C     -      -- ----                                                    ETOGL  5
C                                                                       ETOGL  6
      DIMENSION XFISS(200),XCAP(200)                                    ETOGL  7
      DIMENSION EC(200),ED(200),EM(200),INT(10),NPT(10),E(2000),S(2000),ETOGL  8
     1   HOL(10),SM(200),TITL(10),XLAB(10),YLAB(10),ER(2000),EMR(200),  ETOGL  9
     2    SC(200),SR(2000),WI(200),W(2000),WM(200),EWI(200),NECT(10),   ETOGL 10
     3    NEMCT(10),SBD(10),SMBD(10),BMIN(12),BMAX(12),DELU(12)         ETOGL 11
                                                                        ETOGL 12
      DIMENSION XP(500),YP(500)                                         ETOGL 13
                                                                        ETOGL 14
C     E. IS BASIC CALCULATED MESH,ED ARE ADDITIONAL POINTS TO BE ADDED, ETOGL 15
C     E.G. BROAD GROUP MESH BREAK POINTS IN GLEN,RESONANCE PEAKS,       ETOGL 16
C     VALLEYS,ETC..AND EM IS COMBINED MESH.                             ETOGL 17
C     FIRST READ BMIN,BMAX,DELU FOR EACH REGION (UP TO 8) FOR           ETOGL 18
C     CALCULATION OF BASIC E MESH -- DESCENDING ORDER.                  ETOGL 19
C     EMIN IS LOWEST ENERGY BOUND, EMAX IS HIGHEST                      ETOGL 20
C     EC ARE BROAD GROUP BREAK POINTS TO BE ADDED TO PENDF MESH         ETOGL 21
C      FOR INTEGRAL CHECK                                               ETOGL 22
C     EWI,WI ARE ENERGY,WEIGHT FUNCTION PAIRS FOR WEIGHTING             ETOGL 23
C     IF INTEGRAL CHECK                                                 ETOGL 24
C                                                                       ETOGL 25
      READ (5,30) NUMBIN                                                ETOGL 26
      DO 5 N=1,NUMBIN                                                   ETOGL 27
      READ (5,10) BMIN(N),BMAX(N),DELU(N)                               ETOGL 28
    5 CONTINUE                                                          ETOGL 29
   10 FORMAT (6E11.4)                                                   ETOGL 30
      IF (NUMBIN.EQ.1)  GO TO 7                                         ETOGL 31
      NUMED=NUMBIN-1                                                    ETOGL 32
      DO 6 N=1,NUMED                                                    ETOGL 33
      IF (BMIN(N).NE.BMAX(N+1))  BMIN(N)=BMAX(N+1)                      ETOGL 34
    6 CONTINUE                                                          ETOGL 35
    7 NPC=1                                                             ETOGL 36
      WRITE (6,12) NUMBIN                                               ETOGL 37
   12 FORMAT (1H1,17X,I2,20H INPUT ENERGY GROUPS//2X,9HGROUP NO.,       ETOGL 38
     1    2X,14HGROUP MAX (EV),2X,14HGROUP MIN (EV),2X,                 ETOGL 39
     2    14HLETHARGY WIDTH)                                            ETOGL 40
      DO 8 N=1,NUMBIN                                                   ETOGL 41
      WRITE (6,13) N,BMAX(N),BMIN(N),DELU(N)                            ETOGL 42
    8 CONTINUE                                                          ETOGL 43
   13 FORMAT (4X,I3,8X,E11.5,5X,E11.5,5X,E11.5)                         ETOGL 44
      DO 20 N=1,NUMBIN                                                  ETOGL 45
      U=0.                                                              ETOGL 46
      E(NPC)=BMAX(N)                                                    FTOGL 47
   11 U=U+DELU(N)                                                       ETOGL 48
      NPC=NPC+1                                                         ETOGL 49
      E(NPC)=BMAX(N)/EXP(U)                                             ETOGL 50
      IF (E(NPC).LE.BMIN(N))  GO TO 20                                  ETOGL 51
      GO TO 11                                                          ETOGL 52
   20 CONTINUE                                                          ETOGL 53
      E(NPC)=BMIN(N)                                                    ETOGL 54
      EMIN=BMIN(NUMBIN)                                                 ETOGL 55
      EMAX=BMAX(1)                                                      ETOGL 56
      DO 21 N=1,NPC                                                     ETOGL 57
      SR(N)=N                                                           ETOGL 58
   21 CONTINUE                                                          ETOGL 59
C                                                                       ETOGL 60
C     READ ED MESH FROM CARDS.                                          ETOGL 61
C                                                                       ETOGL 62
```

```
      READ (5,30) NPD                                                    ETOGL 63
   30 FORMAT (6I11)                                                      ETOGL 64
      READ (5,10) (ED(I),I=1,NPD)                                        ETOGL 65
C                                                                        ETOGL 66
C     READ BROAD GROUP ENERGIES.  READ FROM HI TO LO.                    ETOGL 67
C                                                                        ETOGL 68
      READ (5,30) NFGP                                                   ETOGL 69
      READ (5,10) (EC(I),I=1,NFGP)                                       ETOGL 70
C                                                                        ETOGL 71
C  READ IN WEIGHT FCNS.  READ IN E-LO TO E-HI.  MUST BE LOG-LOG INTERP.  ETOGL 72
C                                                                        ETOGL 73
      READ (5,30) NW                                                     ETOGL 74
      READ (5,10) (EWI(N),WI(N),N=1,NW)                                  ETOGL 75
      TITL(1)=10HTHERMAL WE                                              ETOGL 76
      TITL(2)=10HIGHT FUNCT                                              ETOGL 77
      TITL(3)=10HION (GLEN)                                              ETOGL 78
      NWPLT=-NW                                                          ETOGL 79
      XLAB(1)=10HENERGY(EV)                                              ETOGL 80
      YLAB(1)=10HTHERM FLUX                                              ETOGL 81
      CALL PLOJB (EWI,WI,NWPLT,-1,0,0,0,0,1,0,1,0,TITL,30,XLAB,10,YLAB,  ETOGL 82
     1    10)                                                            ETOGL 83
C                                                                        ETOGL 84
C                                                                        ETOGL 85
C     COMBINE E  AND ED TO FORM EM.                                      ETOGL 86
C                                                                        ETOGL 87
      CALL UNION (E,ED,NPC,NPD)                                          ETOGL 88
C                                                                        ETOGL 89
C     REORDER E TO GET EM                                                ETOGL 90
C                                                                        ETOGL 91
      DO 25 N=1,NPC                                                      ETOGL 92
      NN1=NPC-N+1                                                        ETOGL 93
      EM(N)=E(NN1)                                                       ETOGL 94
   25 CONTINUE                                                           ETOGL 95
      JNPC=NPC-1                                                         ETOGL 96
      NNTST=0                                                            ETOGL 97
      DO 35 JI=1,JNPC                                                    ETOGL 98
      DLTST=(EM(JI+1)-EM(JI))/EM(JI)*100.                                ETOGL 99
      IF (DLTST.GT.1.0) GO TO 35                                         ETOGL100
      PRINT 36,JI,EM(JI),EM(JI+1),DLTST                                  ETOGL101
      NNTST=NNTST+1                                                      ETOGL102
   35 CONTINUE                                                           ETOGL103
   36 FORMAT (1H0,* DUPLICATE ENERGIES AT *I4,*,*1PE12.5,* AND *1PE12.5,ETOGL104
     1    *  PCT DIFF = *F6.3)                                          ETOGL105
      IF (NNTST.EQ.0) PRINT 37                                           ETOGL106
   37 FORMAT (1H0,*  THERE ARE NO ENERGY PAIRS WITHIN ONE PERLENT,*)     ETOGL107
      NPM=NPC                                                            ETOGL108
   40 FORMAT (1H0,*  I = *I3,*  E = *1PE12.5)                            ETOGL109
C                                                                        ETOGL110
C     PUNCH ENERGY MESH FOR GLEN.                                        ETOGL111
C                                                                        ETOGL112
      DO 45 N=1,NPM                                                      ETOGL113
      NN1=NPM-N+1                                                        ETOGL114
      E(N)=EM(NN1)                                                       ETOGL115
   45 CONTINUE                                                           ETOGL116
      TITL(5)=10HENERGY MES                                              ETOGL117
      TITL(6)=10HH FOR GLEN                                              ETOGL118
      PUNCH 210,(TITL(I),I=5,6)                                          ETOGL119
      PUNCH 150,(E(N),N=1,NPM)                                           ETOGL120
C                                                                        ETOGL121
C     THERE ARE SEVERAL TEMPERATURES ON TAPE.  READ NOTEMP=NO. OF TEMPS. ETOGL122
C                                                                        ETOGL123
      READ (5,30) NOTEMP,MAT1                                            ETOGL124
      DO 1000 NNT=1,NOTEMP                                               ETOGL125
```

```
C                                                                       ETOGL126
C     FOR EACH TEMP,GET XSEC FOR MT=2,MT=18,MT=102.                     ETOGL127
C                                                                       ETOGL128
      DO 900 NMT=1,3                                                    ETOGL129
      MF1=3                                                             ETOGL130
      MT1=2                                                             ETOGL131
      IF (NMT.EQ.2) MT1=18                                              ETOGL132
      IF (NMT.EQ.3) MT1=102                                             ETOGL133
      IF (NMT.EQ.3.AND.MAT1.EQ.1155) MT1=107                            ETOGL134
   50 READ (11,60) (HOL(I),I=1,7),MAT,MF,MT,NSEQ                       ETOGL135
   60 FORMAT (6A10,A6,I4,I2,I3,I5)                                      ETOGL136
      IF (MAT.LT.0)  GO TO 2000                                         ETOGL137
      IF (MAT.LT.MAT1)  GO TO 50                                        ETOGL138
      IF (MAT.GT.MAT1)  GO TO 2000                                      ETOGL139
      IF (MF.NE.3) GO TO 50                                             ETOGL140
      IF (MT.NE.MT1) GO TO 50                                           ETOGL141
      WRITE (6,2020) MAT,MAT1,MF,MT,MT1,NMT                             ETOGL142
 2020 FORMAT (1H ,*MAT=*I4,3X,*MAT1=*I4,3X,*MF=*I2,3X,*MT=*I3,3X,*MT1=*,ETOGL143
     1    I3,3X,*NMT=*I2)                                              ETOGL144
      READ (11,70) C1,C2,N1,N2,NR,NP                                    ETOGL145
   70 FORMAT(1P2E11.4,4I11)                                             ETOGL146
      TEMDS=C1                                                          ETOGL147
      READ (11,30) (NPT(I),INT(I),I=1,NR)                               ETOGL148
C                                                                       ETOGL149
C     ASSUME THERMAL RANGE IS WITHIN FIRST 2000 PTS ON TAPE.            ETOGL150
C                                                                       ETOGL151
      NPTH=NP                                                           ETOGL152
      IF (NP.GT.2000) NPTH=2000                                         ETOGL153
      READ (11,10) (E(I),S(I),I=1,NPTH)                                 ETOGL154
   80 READ (11,60) (HOL(I),I=1,7),MAT,MF,MT,NSEQ                       ETOGL155
      IF (MT.NE.0) GO TO 80                                             ETOGL156
      IF (E(NPTH).GT.EMAX) GO TO 82                                     ETOGL157
      WRITE (6,81) NPTH,E(NPTH)                                         ETOGL158
   81 FORMAT (1H1,*  EMAX NOT WITHIN *I4,* PTS. LAST ENERGY = *1PE12.5) ETOGL159
      STOP                                                              ETOGL160
   82 CONTINUE                                                          ETOGL161
C                                                                       ETOGL162
C     GET XSEC,SM,CORRESPONDING TO EM.                                  ETOGL163
C                                                                       ETOGL164
      DO 100 I=1,NPM                                                    ETOGL165
      ILO=LOCT(E,EM(I),NPTH)                                            ETOGL166
      IF (ILO.EQ.-1) CALL ERROR(100)                                    ETOGL167
      IHI=ILO+1                                                         ETOGL168
      DO 85 J=1,NR                                                      ETOGL169
      IF (IHI.LE.NPT(J)) GO TO 90                                       ETOGL170
   85 CONTINUE                                                          ETOGL171
      CALL ERROR (200)                                                  ETOGL172
   90 CALL TERP1 (E(ILO),S(ILO),E(IHI),S(IHI),EM(I),CSEC,INT(J))        ETOGL173
      SM(I)=CSEC                                                        ETOGL174
  100 CONTINUE                                                          ETOGL175
  110 FORMAT (1H0,*  I=*I6,*  EM = *1PE11.4,*  SM = *1PE11.4)           ETOGL176
C                                                                       ETOGL177
C     CHECK INTEGRALS AND MAKE COMPARISON PLOTS.                        ETOGL178
C                                                                       ETOGL179
C     CUT OFF MESH POINTS ABOVE EMAX                                    ETOGL180
      KTHRM=0                                                           ETOGL181
      DO 120 N=1,NPTH                                                   ETOGL182
      IF (E(N).GT.EMAX) GO TO 130                                       ETOGL183
      KTHRM=KTHRM+1                                                     ETOGL184
  120 CONTINUE                                                          ETOGL185
  130 CONTINUE                                                          ETOGL186
      TITL(1)=10HETOGLEN VS                                             ETOGL187
      TITL(2)=10H PENDF PTS                                             ETOGL188
```

```
      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
      SMBO(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)                                  TERP1  1
C     ****INTERPOLATE ONE PT.*****                                          TERP1  2
C     (X1,Y1) AND (X2,Y2) ARE END PTS. OF THE LINE                          TERP1  3
C     (X,Y) IS INTERPOLATED POINT                                           TERP1  4
C     I=INTERPOLATION CODE                                                  TERP1  5
C     NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE       TERP1  6
C            INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.     TERP1  7
C     ERROR STOPS - 301 (X1=X2,DISCONTINUITY)                               TERP1  8
C                   302 (INTERPOLATION CODE IS OUT OF RANGE)                TERP1  9
C                   303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT.)    TERP1 10
    5 XA=X1                                                                 TERP1 11
      YA=Y1                                                                 TERP1 12
      XB=X2                                                                 TERP1 13
      YB=Y2                                                                 TERP1 14
      XP=X                                                                  TERP1 15
      II=I                                                                  TERP1 16
      IF (XA.EQ.XB) CALL ERROR (301)                                        TERP1 17
      IF (II) 10,10,15                                                      TERP1 18
   10 CALL ERROR (302)                                                      TERP1 19
   15 IF (II-5) 20,20,10                                                    TERP1 20
   20 GO TO (25,30,35,60,75), II                                            TERP1 21
   25 YP=YA                                                                 TERP1 22
      IF (XP.EQ.XB)  YP=YB                                                  TERP1 23
      GO TO 105                                                             TERP1 24
   30 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)                                         TERP1 25
      GO TO 105                                                             TERP1 26
   35 IF (XA) 30,30,40                                                      TERP1 27
   40 IF (XB) 30,30,45                                                      TERP1 28
   45 IF (XP) 50,50,55                                                      TERP1 29
   50 CALL ERROR (303)                                                      TERP1 30
   55 YP=YA*ALOG(XP/XA)*(YB-YA)/ALOG(XB/XA)                                 TERP1 31
      GO TO 105                                                             TERP1 32
   60 IF (YA) 30,30,65                                                      TERP1 33
   65 IF (YB) 30,30,70                                                      TERP1 34
   70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))                                TERP1 35
      GO TO 105                                                             TERP1 36
   75 IF (YA) 35,35,80                                                      TERP1 37
   80 IF (YB) 35,35,85                                                      TERP1 38
   85 IF (XA) 70,70,90                                                      TERP1 39
   90 IF (XB) 70,70,95                                                      TERP1 40
   95 IF (XP) 80,50,100                                                     TERP1 41
  100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))                        TERP1 42
  105 Y=YP                                                                  TERP1 43
      RETURN                                                                TERP1 44
      END                                                                   TERP1 45
```

```
      SUBROUTINE ERROR (N)                                                  ERROR  1
      IOS=9                                                                 ERROR  2
    5 PRINT 10,N                                                            ERROR  3
      WRITE(99,10)                                                          ERROR  4
   10 FORMAT (11H ERROR STOP,I6)                                            ERROR  5
      END                                                                   ERROR  6
```

```
      SUBROUTINE UNION (XU,X,NPU,NP)                                    UNION  1
C                                                                       UNION  2
C  FUNCTION OF SUBROUTINE                                               UNION  3
C     UNION1 COMPUTES THE UNION OF INDEPENDENT VARIABLE SETS X(IP),IP=1,UNION  4
C          XU(IP1),IP1=1,NPU, AND PLACES THE UNION INTO XU(IP2),IP2=1,NPUNION  5
C  STORAGE                                                              UNION  6
      DIMENSION XU(2000),KU(2000),X(200)                                UNION  7
C                                                                       UNION  8
C     ADD A SET X   TO AN EXISTING UNION SET XU                         UNION  9
      DO 106 IP=1,NPU                                                   UNION 10
      KU(IP)=0                                                          UNION 11
  106 CONTINUE                                                          UNION 12
      DO 103 IP=1,NP                                                    UNION 13
      IF (X(IP).LT.XU(NPU)) GO TO 120                                   UNION 14
      IF (X(IP).GT.XU(1)) GO TO 130                                     UNION 15
      DO 104 IP1=1,NPU                                                  UNION 16
      IF (X(IP).EQ.XU(IP1)) GO TO 140                                   UNION 17
      IF (IP1.EQ.NPU) GO TO 105                                         UNION 18
      IF (X(IP).LT.XU(IP1).AND.X(IP).GT.XU(IP1+1)) GO TO 150            UNION 19
  105 CONTINUE                                                          UNION 20
  104 CONTINUE                                                          UNION 21
C                                                                       UNION 22
C     HERE NPU IS INCREMENTED BY ONE AND A POINT IS ADDED TO THE LEFT   UNION 23
  120 NPU=NPU+1                                                         UNION 24
      XU(NPU)=X(IP)                                                     UNION 25
      KU(NPU)=1                                                         UNION 26
  121 CONTINUE                                                          UNION 27
      GO TO 103                                                         UNION 28
C                                                                       UNION 29
C     HERE CONTROLS ARE SET TO ADD A POINT ON THE RIGHT                 UNION 30
  130 KONREL=1                                                          UNION 31
      NPMOV=NPU                                                         UNION 32
      GO TO 170                                                         UNION 33
C                                                                       UNION 34
C     HERE NPU IS NOT INCREMENTED BY ONE .                              UNION 35
  140 CONTINUE                                                          UNION 36
      KU(IP1)=1                                                         UNION 37
      GO TO 103                                                         UNION 38
C                                                                       UNION 39
C     HERE NPU IS INCREMENTED BY ONE AND CONTROLS ARE SET TO ADD A POINTUNION 40
C          BETWEEN POINTS IP1 AND IP1+1                                 UNION 41
  150 KONREL=2                                                          UNION 42
      NPMOV=NPU-IP1                                                     UNION 43
      GO TO 170                                                         UNION 44
C                                                                       UNION 45
C     HERE WE INCREMENT NPU BY ONE AND MOVE THE LEFT-MOST NPMOV POINTS IUNION 46
C          SET ONE POSITION TO THE LEFT                                 UNION 47
  170 NPU=NPU+1                                                         UNION 48
      DO 171 IP2=1,NPMOV                                                UNION 49
      XU(NPU-IP2+1)=XU(NPU-IP2)                                         UNION 50
      KU(NPU-IP2+1)=KU(NPU-IP2)                                         UNION 51
  171 CONTINUE                                                          UNION 52
C                                                                       UNION 53
C     HERE A NEW POINT IS ADDED                                         UNION 54
      NPADD=NPU-NPMOV                                                   UNION 55
      XU(NPADD)=X(IP)                                                   UNION 56
      KU(NPADD)=1                                                       UNION 57
  172 CONTINUE                                                          UNION 58
  103 CONTINUE                                                          UNION 59
  102 RETURN                                                            UNION 60
      END                                                               UNION 61
```

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,        MERGF  1
     1    FSET9,FSET10)                                                 MERGF  2
C     MERGFAT IS A VERSION OF JUMRLFAT THAT ACCEPTS FAST XSEC AS         MERGF  3
C     OUTPUT BY THE HONEDX VERSION OF THE 1DX CODE.                      MERGF  4
C     REVISIONS MADE AT LASL BY LABAUVE,NOV75.                           MERGF  5
C                                                                        MERGF  6
      DIMENSION TOTP0(70),TOTP1(70),TOTRA(70)                            MERGF  7
      DIMENSION NUCID(20),N2N(20),SK2KF(70,70),DIAGSM(70),CHKSM(70),     MERGF  8
     1    DIFF(70),CAPA(70),FISA(70),STR(70),AVNU(70),SINTRA(70,70),     MERGF  9
     2    SCAP(70),SS(70),SABS(70),HOL(70),KK(70),SSN2N(70,70),          MERGF 10
     3    XNUSIG(70),A(2000),P0(70,70),P1(70,70),P2(70,70),P3(70,70),    MERGF 11
     4    XS(70),TITLE(12),TOTN2N(70),SNP(70),SND(70),SNT(70),SNHE3(70),MERGF 12
     5    SNA(70),SN2A(70)                                               MERGF 13
      DIMENSION XI(50),GPEN(50),TTL(12),VEL(50),FACAP(50),FAFIS(50),     MERGF 14
     1  AOEN(20),TOTIN(70)                                               MERGF 15
      DIMENSION KG1(20),KG2(20)                                          MERGF 16
      DIMENSION V(50),HLT(10)                                            MERGF 17
      NTPF=9                                                             MERGF 18
      NTPT=10                                                            MERGF 19
 5002 READ(8,5001)(HLT(I),I=1,9)                                         MERGF 20
 5001 FORMAT(9A8)                                                        MERGF 21
      PRINT 600                                                          MERGF 22
  600 FORMAT (1H1,20X,10HINPUT DATA ///)                                 MERGF 23
      READ (8,5) LENG,NOELU,LTL,LNGUP                                    MERGF 24
      PRINT 601,LENG,NOELU,LTL,LNGUP                                     MERGF 25
```

```
  601 FORMAT (1H010X7HLENG = ,I6,10H, NDELU = ,I6,8H, LTL = ,I6,          MERGF 26
     110H, LNGUP = ,I6)                                                  MERGF 27
      READ( 8,70  )(TITLE(I),I=1,7)                                      MERGF 28
      PRINT 602,(TITLE(I),I=1,7)                                         MERGF 29
  602 FORMAT (1H05X,8A10)                                                MERGF 30
      READ (8,71) NOBG,NOI,IOPT                                          MERGF 31
      PRINT 603,NOBG,NOI,IOPT                                            MERGF 32
  603 FORMAT (1H010X7HNOBG = ,I12,8H, NOI = ,I12,9H, IOPT = ,I12)        MERGF 33
      NMAT=NOI                                                           MERGF 34
   71 FORMAT (3I12)                                                      MERGF 35
      KGROPS=NOBG                                                        MERGF 36
   70 FORMAT (8A10)                                                      MERGF 37
       PUNCH 1000, (TITLE(I),I=1,12)                                     MERGF 38
      READ ( 8,3) (GPEN(I),I=1,KGROPS)                                   MERGF 39
      PRINT 604                                                          MERGF 40
  604 FORMAT (4H0  I3X7HGPEN(I))                                         MERGF 41
      PRINT 605,(I,GPEN(I),I=1,KGROPS)                                   MERGF 42
  605 FORMAT (I3,1PE12.5)                                                MERGF 43
      AVEN1=(GPEN(KGROPS-1)+GPEN(KGROPS))/2.0                            MERGF 44
      AVEN2=(GPEN(KGROPS))/2.0                                           MERGF 45
      FAXT=0.                                                            MERGF 46
      READ ( 8,3) (XI(I),I=1,KGROPS)                                     MERGF 47
      PRINT 606                                                          MERGF 48
  606 FORMAT (4H0  I6X5HXI(I))                                           MERGF 49
      PRINT 605,(I,XI(I),I=1,KGROPS)                                     MERGF 50
      V(1)=1.0E+07                                                       MERGF 51
      DO 4000 K=1,KGROPS                                                 MERGF 52
 4000 V(K+1)=GPEN(K)                                                     MERGF 53
      DO 72 K=1,KGROPS                                                   MERGF 54
   72 VEL(K)=0.007*(V(K)**0.5+V(K+1)**0.5)                               MERGF 55
C      *** NOTE *** THESE ARE JUST AVERAGE BROAD GROUP VELOCITIES.        MERGF 56
      LBL1=6H SORS                                                       MERGF 57
      LBL2=6H VELS                                                       MERGF 58
      I1=1                                                               MERGF 59
      M1=1                                                               MERGF 60
   73 I2=I1+5                                                            MERGF 61
      IF (I2.LE.KGROPS) GO TO 74                                         MERGF 62
      PUNCH 75,(XI(I),I=I1,KGROPS)                                       MERGF 63
   75 FORMAT (6F12.6)                                                    MERGF 64
      GO TO 76                                                           MERGF 65
   74 PUNCH 87,(XI(I),I=I1,I2),LBL1,M1                                   MERGF 66
      M1=M1+1                                                            MERGF 67
      I1=I2+1                                                            MERGF 68
      GO TO 73                                                           MERGF 69
   76 CONTINUE                                                           MERGF 70
      I1=1                                                               MERGF 71
      M1=1                                                               MERGF 72
   77 I2=I1+5                                                            MERGF 73
      IF (I2.LE.KGROPS) GO TO 78                                         MERGF 74
      PUNCH 79,(VEL(I),I=I1,KGROPS)                                      MERGF 75
   79 FORMAT (1P6E12.2)                                                  MERGF 76
      GO TO 1180                                                         MERGF 77
   78 PUNCH 81,(VEL(I),I=I1,I2),LBL2,M1                                  MERGF 78
      M1=M1+1                                                            MERGF 79
      I1=I2+1                                                            MERGF 80
      GO TO 77                                                           MERGF 81
 1180 CONTINUE                                                           MERGF 82
   87 FORMAT (  6F12.6,A6,I2)                                            MERGF 83
   81 FORMAT (1P6E12.2,A6,I2)                                            MERGF 84
    5 FORMAT (12I6)                                                      MERGF 85
      MATNO=1                                                            MERGF 86
      READ (8,5) NDKS,(KG1(N),KG2(N),N=1,NDKS)                           MERGF 87
      PRINT 607,NDKS                                                     MERGF 88
```

```
  607 FORMAT (1H010X7HNDKS = ,I6)                                          MERGF 89
      PRINT 608                                                            MERGF 90
  608 FORMAT (4H0  N3X6HKG1(N)7X6HKG2(N))                                  MERGF 91
      PRINT 609,(N,KG1(N),KG2(N),N=1,NDKS)                                 MERGF 92
  609 FORMAT (I3,2I6)                                                      MERGF 93
      DO 999 M=1,NMAT                                                      MERGF 94
      REWIND NTPF $ REWIND NTPT                                            MERGF 95
      READ (8,4100) NUCLE,MODER,MATID                                      MERGF 96
 4100 FORMAT (A6,2X,A10,A6)                                                MERGF 97
      DO 500 N=1,NDKS                                                      MERGF 98
      K1=KG1(N)                                                            MERGF 99
      NOBG=KG2(N)                                                          MERGF100
      LTABL=LENG+6                                                         MERGF101
         NOBG2 = NOBG + 1                                                  MERGF102
      FAFIS(N)=1.0                                                         MERGF103
 2102 FACAP(N)=1.                                                          MERGF104
   85 FORMAT (15H              G=03,3H L=03,13H NUCLIDE NO.=05,1H 3A10)    MERGF105
 4110 IF (N.EQ.1) READ (NTPF,4115) NUCID(M),FID,(TTL(KK),KK=1,11)          MERGF106
 4115 FORMAT (13A6)                                                        MERGF107
      IF (NUCID(M).NE.NUCLE) GO TO 4110                                    MERGF108
 1085 FORMAT (A6,F6.2,11A6)                                                MERGF109
      DO 3070 K=K1,NOBG                                                    MERGF110
      DO 3070 KF=K1,NOBG                                                   MERGF111
      P0(K,KF)=0.0                                                         MERGF112
      SINTRA(K,KF)=0.0                                                     MERGF113
      SSN2N(K,KF)=0.0                                                      MERGF114
 3070 CONTINUE                                                             MERGF115
      IF (N.EQ.1) GO TO 3000                                               MERGF116
      GNU=AVNU(K1)                                                         MERGF117
      MTBL=2*LNGUP+5                                                       MERGF118
      MGPS=NOBG-K1+1                                                       MERGF119
      MVLG=MTBL*MGPS                                                       MERGF120
      DO 3010 L=1,2                                                        MERGF121
 4120 READ (NTPT,4130) MODES,MATCH                                         MERGF122
 4130 FORMAT (22X,A10,15X,A6)                                              MERGF123
      IF (MODES.EQ.MODER.AND.MATCH.EQ.MATID) GO TO 4135                    MERGF124
      GO TO 4120                                                           MERGF125
115        .03                                                            MERGF126
 4135 CONTINUE                                                             MERGF127
      IF (L.EQ.1) MADD=0                                                   MERGF128
      IF (L.EQ.2) MADD=MVLG                                                MERGF129
      READ (NTPT,3) (A(M+MADD),M=1,MVLG)                                   MERGF130
 3010 CONTINUE                                                             MERGF131
      DO 3020 MG=1,MGPS                                                    MERGF132
      DO 3030 MGM=1,MGPS                                                   MERGF133
      KK1=K1+MG-1                                                          MERGF134
      KK2=K1+MGM-1                                                         MERGF135
      LCK=9-MG+(MGM-1)*(MTBL+1)                                            MERGF136
      P0(KK1,KK2)=A(LCK)                                                   MERGF137
      P1(MG,MGM)=A(LCK+MVLG)                                               MERGF138
 3030 CONTINUE                                                             MERGF139
      PRINT 3040,KK1                                                       MERGF140
 3040 FORMAT (1H0,*  SCATT. MATRIX FOR GP *I3)                             MERGF141
      PRINT 3050,(P0(KK1,KK2),KK2=K1,NOBG)                                 MERGF142
      PRINT 3050,(P1(MG,MGM),MGM=1,MGPS)                                   MERGF143
      LCK=(MG-1)*MTBL+1                                                    MERGF144
      SABS(KK1)=A(LCK)                                                     MERGF145
      XNUSIG(KK1)=A(LCK+1)                                                 MERGF146
      STR(KK1)=A(LCK+2)                                                    MERGF147
      AVNU(KK1)=GNU                                                        MERGF148
      IF (GNU.LE.0.0) GNU=1.E-10                                           MERGF149
      FISA(KK1)=XNUSIG(KK1)/GNU                                            MERGF150
      CAPA(KK1)=SABS(KK1)-FISA(KK1)                                        MERGF151
      P1TOT=0.0                                                            MERGF152
```

```
3050 FORMAT (1P8E15.5)                                                  MERGF153
     DO 3060 MGM=1,MGPS                                                 MERGF154
     P1TOT=P1TOT+P(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 MERGF215
C FIRST L GROUPS IN WHICH ALL CASES ARE CONSIDERED.  ADDITIONAL VALUES AMERGF216
C ADDED INTO THE L -TH GROUP.                                          MERGF217
      DO 90 K=1,NOBG                                                   MERGF218
      MN=LENG+K                                                        MERGF219
      IF (MN.GT.NOBG+1) GO TO 90                                       MERGF220
      DO 91 KF=MN,NOBG2                                                MERGF221
   91 SK2KF(K,MN-1)=SK2KF(K,MN-1)+SK2KF(K,KF)                          MERGF222
   90 CONTINUE                                                         MERGF223
      MK=NOBG-LENG+2                                                   MERGF224
      DO 51 K1=MK,NOBG                                                 MERGF225
   51 SK2KF(K1,NOBG)=SK2KF(K1,NOBG)+SK2KF(K1,NOBG2)                    MERGF226
  300 CONTINUE                                                         MERGF227
      LTABL=LENG+LNGUP+7                                               MERGF228
      NXC=LTABL*NOBG                                                   MERGF229
      NCX6=NXC+LTABL                                                   MERGF230
      DO 360 JT=1,NCX6                                                 MERGF231
  360 A(JT)=0.                                                         MERGF232
      DO 361 K=1,NOBG                                                  MERGF233
      J1=(K-1)*LTABL+1                                                 MERGF234
      J2=J1+1                                                          MERGF235
      J3=J1+2                                                          MERGF236
      J4=J3+1                                                          MERGF237
      J5=J4+1                                                          MERGF238
      J6=J5+1                                                          MERGF239
      J7=J6+1                                                          MERGF240
      A(J1)=FISA(K)                                                    MERGF241
      A(J4)=CAPA(K)                                                    MERGF242
      A(J5)=SARS(K)                                                    MERGF243
      A(J6)=XNUSIG(K)                                                  MERGF244
      A(J7)=STR(K)                                                     MERGF245
      DO 362 KF= K,NOBG2                                               MERGF246
      J8=(KF-1)*LTABL+KF+7-K+1+LNGUP                                   MERGF247
      A(J8)=SK2KF(K,KF)                                                MERGF248
      IF ((KF-K).GE.  LENG) GO TO 361                                  MERGF249
  362 CONTINUE                                                         MERGF250
      IF (FID.NE.6H 12.00) GO TO 361                                   MERGF251
      IF(K.LT.KG1(NDKS)) GO TO 361                                     MERGF252
      KUP1=K+1                                                         MERGF253
      IF(KUP1.GT.NOBG) GO TO 361                                       MERGF254
      DO 3561 KF=KUP1,NOBG                                             MERGF255
      J9=LTABL*(K-1)+8+LNGUP-(KF-K)                                    MERGF256
      A(J9)=P0(KF,K)                                                   MERGF257
 3561 CONTINUE                                                         MERGF258
  361 CONTINUE                                                         MERGF259
  250 CONTINUE                                                         MERGF260
      PUNCH 85,NOBG,LTABL,MATNO,NUCID(M),(HOL(I),I=1,2)                MERGF261
      NP1=1                                                            MERGF262
      NBR=1                                                            MERGF263
   63 NP2=NP1+5                                                        MERGF264
      IF (NP2.LE.NXC ) GOTO 67                                         MERGF265
      PUNCH 65,(A(NP),NP=NP1,NP2),NUCID(M),NBR                         MERGF266
   68 FORMAT (1P6E12.5)                                                MERGF267
      GO TO 64                                                         MERGF268
   67 PUNCH 65,(A(NP),NP=NP1,NP2),NUCID(M),NBR                         MERGF269
      IF (NP2.EQ.NXC ) GOTO 64                                         MERGF270
      NBR=NBR+1                                                        MERGF271
      NP1=NP2+1                                                        MERGF272
      GO TO 63                                                         MERGF273
   64 CONTINUE                                                         MERGF274
   65 FORMAT (1P6E12.5,A6,I2)                                          MERGF275
      DO 103 K=1,NOBG                                                  MERGF276
```

```
  103 KK(K)=K                                                                    MERGF277
      IF(M.NE.1)GOTO9                                                            MERGF278
      WRITE(6,1000)(TITLE(I),I=1,7)                                              MERGF279
 1000 FORMAT(8A10)                                                               MERGF280
    9 CONTINUE                                                                   MERGF281
      WRITE(6,85) NOBG,LTABL,MATNO,NUCID(M),(HOL(I),I=1,2)                       MERGF282
C CHECK TO ADD UP SK2KF                                                          MERGF283
C  THE DIAGONAL SUM OF THE DOWN SCATTERING AND SELF SCATTERING ADDED TO          MERGF284
C ABSORPTION MUST EQUAL THE TRANSPORT CROSS SECTION.                             MERGF285
      DO 400 K=1,NOBG                                                            MERGF286
      DIAGSM(K)=0.                                                               MERGF287
      KK=K+LENG-1                                                                MERGF288
      IF (KK.GT.NOBG) KK=NOBG                                                    MERGF289
      DO 450 KF=K,KK                                                             MERGF290
  450 DIAGSM(K)=DIAGSM(K)+SK2KF(K,KF)                                            MERGF291
      CHKSM(K)=DIAGSM(K)+SABS(K)-TOTN2N(K)                                       MERGF292
      DIFF(K)=CHKSM(K)-STR(K)                                                    MERGF293
      WRITE(6,460)                                                               MERGF294
  460 FORMAT (20H0TRANSPORT           18HCHECKSUM                                MERGF295
     1    10HDIFFERENCE)                                                        MERGF296
      WRITE(6,470) STR(K),CHKSM(K),DIFF(K)                                       MERGF297
  470 FORMAT(E12.4,7X,E12.4,8X,E12.4)                                            MERGF298
      IF(DIFF(K).LE..1*STR(K))GOTO400                                            MERGF299
      WRITE(6,480)                                                               MERGF300
  480 FORMAT(31H0DIFFERENCE EXCEEDS 10 PER CENT)                                 MERGF301
  400 CONTINUE                                                                   MERGF302
      WRITE(6,85) NOBG,LTABL,MATNO,NUCID(M),(HOL(I),I=1,2)                       MERGF303
      DO 701 I=1,NOBG                                                            MERGF304
      K=LTABL*(I-1)                                                              MERGF305
      DO 701 J=1,LTABL                                                           MERGF306
  701 SK2KF(I,J)=A(K+J)                                                          MERGF307
      KA=1 $ KB=8                                                                MERGF308
  724 KC=MIN0(KB,NOBG)                                                           MERGF309
      WRITE(6,720) (K,K=KA,KC)                                                   MERGF310
      WRITE(6,721)                                                               MERGF311
      DO 722 J=1,LTABL                                                           MERGF312
  722 WRITE (6,723) J,(SK2KF(I,J),I=KA,KC)                                       MERGF313
      KA=KA+8 $ KB=KB+8                                                          MERGF314
      IF(KA.LE.NOBG) GO TO 724                                                   MERGF315
  720 FORMAT (1H0,5X,8(2X,5HGROUP,I3,3X))                                        MERGF316
  721 FORMAT (1H )                                                               MERGF317
  723 FORMAT (I4,1P8E13.5)                                                       MERGF318
  100 CONTINUE                                                                   MERGF319
      MATNO=MATNO+1                                                              MERGF320
  999 CONTINUE                                                                   MERGF321
      IF(HLT(1).NE.8HLASTDECK) GO TO 5002                                        MERGF322
      END                                                                        MERGF323
```

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      RADIS 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,RADIS                                  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(6E12.6)                                                     DANCP 27
      PRINT 15,NLAT,NALF,NRAD,IOPTC,RADIS                                DANCP 28
   15 FORMAT (1H1,*NLAT = *,I4,* NALF = *,I4, * NRAD = *,I4,* IOPTC = *,DANCP 29
     1I4,* RADIS = *,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                                                    DANCP 35
      CALL DANCOFF(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC,CS,         DANCP 36
     1CSB,C,IOPTC,RAD1S)                                                 DANCP 37
      CALL LEVINE(SIGF,DENSF,C,RAD0,IOPTC)                               DANCP 38
   25 CONTINUE                                                           DANCP 39
      GO TO 50                                                           DANCP 40
   30 CALL DANCOFF(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC,CS,         DANCP 41
     1CSB,C,IOPTC,RAD1S)                                                 DANCP 42
      CALL LEVINE(SIGF,DENSF,C,RAD0,IOPTC)                               DANCP 43
   50 CONTINUE                                                           DANCP 44
      END                                                                DANCP 45
```

```
      SUBROUTINE LEVINE(SIGF,DENSF,C,RAD0,IOPTC)                         LEVIN  1
C     CALCULATES THE EFFECTIVE GEOMETRIC SHIELDING CROSS SECTION BY THE  LEVIN  2
C     LEVINE METHOD AND USING THE OTTER APPROXIMATION FOR THE LEVINE     LEVIN  3
C     FACTOR                                                             LEVIN  4
      ELBARF=2.*RAD0                                                     LEVIN  5
      TAUF=SIGF*ELBARF                                                   LEVIN  6
      IF(TAUF.LE.0.) PRINT 20                                            LEVIN  7
   20 FORMAT(1H0,*TAUF IS G. OR E. TO ZERO*)                             LEVIN  8
      IF(TAUF.GE.2.) GO TO 30                                            LEVIN  9
      ALEVI=1.5013+0.14879*TAUF**0.5-0.17226*TAUF                        LEVIN 10
      GO TO 40                                                           LEVIN 11
   30 ALEVI=1.+1./TAUF-1./TAUF**3.                                       LEVIN 12
   40 CONTINUE                                                           LEVIN 13
      PRINT 50,ALEVI                                                     LEVIN 14
   50 FORMAT(1H0,*THE LEVINE FACTOR EQUALS *,1PE12.4)                    LEVIN 15
      QDEN=ELBARF*(1.+(ALEVI-1.)*C)                                      LEVIN 16
      QNUM=ALEVI*(1.-C)                                                  LEVIN 17
      EFXSEC=QNUM/QDEN/DENSF                                             LEVIN 18
      IF(IOPTC.EQ.1) ANAME=10H(CARLVIK)                                  LEVIN 19
      IF(IOPTC.EQ.2) ANAME=10H(BONALUMI)                                 LEVIN 20
      IF(IOPTC.EQ.3) ANAME=10H (SAUER)                                   LEVIN 21
      PRINT 10,ANAME,EFXSEC                                              LEVIN 22
   10 FORMAT(1H0,*EFFECTIVE SIGMA WITH DANCOFF FACTOR *,A10,* = *,       LEVIN 23
     1 1PE12.4)                                                          LEVIN 24
      RETURN                                                             LEVIN 25
      END                                                                LEVIN 26
```

```
      SUBROUTINE DANCOFF(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC,CS,   DANCO  1
     1                   CSB,C,IOPTC,RAD1S)                              DANCO  2
C     CALCULATES DANCOFF FACTORS BY THE ORIGINAL METHOD AS USED BY       DANCO  3
C     CARLVIK, BY THE SAUER APPROXIMATION AND BY THE BONALUMI            DANCO  4
C     APPROXIMATION AND COMPARES THE RESULTS OF THE THREE                DANCO  5
      IF(IOPTC)10,10,20                                                  DANCO  6
   10 PRINT 15                                                           DANCO  7
   15 FORMAT(1H0,*IOPTC IS ZERO OR NEGATIVE,TRY AGAIN*)                  DANCO  8
      RETURN                                                             DANCO  9
```

```
   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=3.14159265                                                      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)                       BONAL  1
C     CALCULATES DANCOFF FACTORS BY THE BONALUMI APPROXIMATION          BONAL  2
      TAU=0.                                                            BONAL  3
      PI=3.141592654                                                    BONAL  4
      RADRA=RAD1/RAD0                                                   BONAL  5
      VOLRA=RADRA*RADRA-1.                                              BONAL  6
      VOLSQR=(1.+VOLRA)**0.5                                            BONAL  7
      IF(NLAT.EQ.4)TAU=((PI/4.)**0.5*VOLSQR-1.)/VOLRA                   BONAL  8
      IF(NLAT.EQ.6)TAU=((PI/(3.**0.5*2.))**0.5*VOLSQR-1.)/VOLRA         BONAL  9
      IF(TAU)10,10,20                                                   BONAL 10
   10 PRINT 15                                                          BONAL 11
   15 FORMAT(1H0,*TAU IS ZERO,NLAT VALUE IS WRONG*)                     BONAL 12
      RETURN                                                            BONAL 13
   20 ELBARF=2.*RAD0                                                    BONAL 14
      ELBARM=ELBARF*VOLRA                                               BONAL 15
      PROD=SIGMAM*ELBARM                                                BONAL 16
      IF(NLAT.EQ.4)BETA=5.67                                            BONAL 17
      IF(NLAT.EQ.6)BETA=2.125                                           BONAL 18
      DELTAU=PROD/(7.+BETA*PROD)                                        BONAL 19
      TAU1=TAU+DELTAU                                                   BONAL 20
      DANCOF=EXP(-TAU*PROD)/(1.+(1.-TAU1)*PROD)                         BONAL 21
      CSB=DANCOF                                                        BONAL 22
      RETURN                                                            BONAL 23
      END                                                               BONAL 24
```

```
      SUBROUTINE BKLY(X,BIC3)                                           BKLY   1
C     CALCULATES BICKLEY FUNCTIONS OF THE THIRD ORDER                   BKLY   2
      A0=0.937938841                                                    BKLY   3
      A1=1.194191634                                                    BKLY   4
      A2=0.588245154                                                    BKLY   5
      A3=0.570337193                                                    BKLY   6
      A4=-1.5791166                                                     BKLY   7
      A5=4.292469                                                       BKLY   8
      B0=0.7276787064                                                   BKLY   9
      B1=0.9254690857                                                   BKLY  10
      B2=0.4741520763                                                   BKLY  11
      B3=0.250820355                                                    BKLY  12
      B4=-0.02593007 5                                                  BKLY  13
      B5=0.055707999                                                    BKLY  14
      C0=0.4166740874                                                   BKLY  15
      C1=0.5295655111                                                   BKLY  16
      C2=0.2754273045                                                   BKLY  17
      C3=0.1283775092                                                   BKLY  18
      C4=0.0119191487                                                   BKLY  19
      C5=0.0139209543                                                   BKLY  20
      D0=0.2215940159                                                   BKLY  21
      D1=-0.09388379097                                                 BKLY  22
      D2=0.01473882145                                                  BKLY  23
      D3=-0.000857650032                                                BKLY  24
      E0=0.2826723681                                                   BKLY  25
      E1=0.2356320335                                                   BKLY  26
      F2=0.06340205186                                                  BKLY  27
      E3=0.01360032364                                                  BKLY  28
```

```
      F0=1.012074180                                                    BKLY  29
      F1=-0.000325432                                                   BKLY  30
      F2=-1.1646323                                                     BKLY  31
      F3=1.3873864                                                      BKLY  32
      F4=-4.4655208                                                     BKLY  33
      X2=X*X                                                            BKLY  34
      X3=X**2*X                                                         BKLY  35
      X4=X3*X                                                           BKLY  36
      X5=X4*X                                                           BKLY  37
      IF(X)10,20,20                                                     BKLY  38
   10 PRINT 15                                                          BKLY  39
   15 FORMAT(1H0,*X IS LESS THAN ZERO,TRY AGAIN*)                       BKLY  40
      RETURN                                                            BKLY  41
   20 IF(X-0.1)30,40,40                                                 BKLY  42
   30 SUM=A0+A1*X+A2*X2+A3*X3+A4*X4+A5*X5                               BKLY  43
      BIC3=0.7366554521/SUM                                             BKLY  44
      RETURN                                                            BKLY  45
   40 IF(X-0.4)50,60,60                                                 BKLY  46
   50 SUM=B0+B1*X+B2*X2+B3*X3+B4*X4+B5*X5                               BKLY  47
      BIC3=0.5714977571/SUM                                             BKLY  48
      RETURN                                                            BKLY  49
   60 IF(X-1.0)70,80,80                                                 BKLY  50
   70 SUM=C0+C1*X+C2*X2+C3*X3+C4*X4+C5*X5                               BKLY  51
      BIC3=0.3272473766/SUM                                             BKLY  52
      RETURN                                                            BKLY  53
   80 IF(X-2.5)90,100,100                                               BKLY  54
   90 BIC3=(D0+D1*X+D2*X2+D3*X3)/(E0+E1*X+E2*X2+E3*X3)                  BKLY  55
      RETURN                                                            BKLY  56
  100 Y=1./(X+3.25)                                                     BKLY  57
      SUM=F0+F1*X+F2*X2+F3*X3+F4*X4                                     BKLY  58
      BIC3=1.2488445824*EXP(-X)/(Y**0.5*SUM)                            BKLY  59
      RETURN                                                            BKLY  60
      END                                                               BKLY  61
```

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