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

Issued: February 1977
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

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Nuclear Regulatory Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.
Methods for Calculating Group Cross Sections for Doubly Heterogeneous Thermal Reactor Systems

by

M. G. Stamotelatos
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-

![Diagram](image-url)

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

**TABLE I**

<table>
<thead>
<tr>
<th>Group No.</th>
<th>Lower Boundary (eV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$1.83 \times 10^5$</td>
</tr>
<tr>
<td>2</td>
<td>$9.61 \times 10^4$</td>
</tr>
<tr>
<td>3</td>
<td>$1.76 \times 10^4$</td>
</tr>
<tr>
<td>4</td>
<td>3.93</td>
</tr>
<tr>
<td>5</td>
<td>2.38</td>
</tr>
<tr>
<td>6</td>
<td>$4.14 \times 10^{-1}$</td>
</tr>
<tr>
<td>7</td>
<td>$1.00 \times 10^{-1}$</td>
</tr>
<tr>
<td>8</td>
<td>$4.00 \times 10^{-2}$</td>
</tr>
<tr>
<td>9</td>
<td>$5.00 \times 10^{-4}$</td>
</tr>
</tbody>
</table>
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 temperatures (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 MC²-I 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 MC² input file is shown in the diagram of Fig. 2. The RIGEL² code is used to convert ENDF/B data in standard BCD format (Mode 3) to an alternate binary format (Mode 2). The ETOE³ code prepares a library tape for MC² including "W-tables" that are supplied by the WLIB code. Since ETOE provides pointwise elastic-scattering cross sections for MC², temperature must be an input parameter to ETOE which means that a different MC² library tape must be prepared for each temperature. The various MC² libraries are then merged with an auxiliary code, MERMC2, not shown in Fig. 1. There are certain limitations connected with the MC² code, some of which have proved to be so hard to circumvent, unless considerable effort was put in modifying the code, that MC²-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 x 10⁻⁴ eV) energy ranges. Second, the maximum energy value in MC²-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. MC²-I library preparation.](image-url)
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:

\[
\left(\frac{\sigma_{	ext{c},j}}{\sigma_{	ext{c},j}}\right)_{\text{resolved}} = \frac{\sum_{j} <\sigma >_{\text{fg}}^{j} \text{ resolved } Q_{j}}{\sum_{j} Q_{j}},
\]  

(1)

where

\[
Q_{j} = \int_{E_{j}}^{E_{j+1}} S_{j}^{\text{fg}} dE \frac{dE}{\Sigma_{T}(E)},
\]  

(2)

and

\(4\)
\[
S_{fg}^{j} = \sum_{k \neq j} \Sigma_{\text{inel},k}^{fg} \phi_{k}^{fg} \frac{\langle \Sigma_{\text{inel}}^{fg} \rangle_{k+j}}{\langle \Sigma_{\text{inel}}^{fg} \rangle_{k}} + \sum_{k \neq j} \Sigma_{n,2n,k}^{fg} \phi_{k}^{fg} \frac{2\langle \Sigma_{n,2n}^{fg} \rangle_{k+j}}{\langle \Sigma_{n,2n}^{fg} \rangle_{k}} \\
+ \langle \Sigma_{el}^{fg} \rangle_{j-1+j} \phi_{j-1}^{fg}
\]

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 \(^{135}\text{Xe}\) or \(^{149}\text{Sm}\) 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.
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 IDX$^{14}$ 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-
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. The 69-group structure consists of the GAM-I group structure plus a dump group necessary to obtain the correct eigenvalue in lDX. 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 MC² 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 lDX code. This is the file designated by "A" in Fig. 3. Note that file A contains temperature-dependent f-factors for Bondarenko treatment by lDX.

5. The ETOGLEN code is used to retrieve cross-section thermal data (from $5 \times 10^{-4}$ 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
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 method of grain shielding to pointwise (PENDF) cross sections followed by the application of the Levine formalism of "gross" (fuel-rod) space shielding to collapsed grain-shielded fine-group cross sections. The

<table>
<thead>
<tr>
<th>Nuclide</th>
<th>MAT NO.</th>
<th>ENDF/B-VERSION</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. B-10</td>
<td>1155</td>
<td>III</td>
<td>Core</td>
</tr>
<tr>
<td>2. C-12</td>
<td>1165</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>3. O-16</td>
<td>1134</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>4. Si-28</td>
<td>1194</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>5. Xe-135</td>
<td>1294</td>
<td>IV</td>
<td>&quot;</td>
</tr>
<tr>
<td>6. Sm-149</td>
<td>1027</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>7. Th-232</td>
<td>1117</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>8. Pa-233</td>
<td>1119</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>9. Pa-233</td>
<td>1297</td>
<td>IV</td>
<td>&quot;</td>
</tr>
<tr>
<td>10. U-233</td>
<td>1260</td>
<td>IV</td>
<td>&quot;</td>
</tr>
<tr>
<td>11. U-234</td>
<td>1043</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>12. U-235</td>
<td>1157</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>13. U-236</td>
<td>1163</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>14. U-238</td>
<td>1158</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>15. Pu-238</td>
<td>1050</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>16. Pu-239</td>
<td>1264</td>
<td>IV</td>
<td>&quot;</td>
</tr>
<tr>
<td>17. Pu-240</td>
<td>1265</td>
<td>IV</td>
<td>&quot;</td>
</tr>
<tr>
<td>18. Pu-241</td>
<td>1266</td>
<td>IV</td>
<td>&quot;</td>
</tr>
<tr>
<td>19. Pu-242</td>
<td>1161</td>
<td>III</td>
<td>&quot;</td>
</tr>
<tr>
<td>20. B-10</td>
<td>1155</td>
<td>III</td>
<td>reflector</td>
</tr>
<tr>
<td>21. C-12</td>
<td>1165</td>
<td>III</td>
<td>reflector</td>
</tr>
</tbody>
</table>
grain shielding was implemented in the PETOPES code and the gross heterogeneity correction was made in a modified lDX 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 and the GGC-5 codes.

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

$$\sigma_{\text{eff}}(E) = \sigma_1(E) \frac{\Gamma(E)}{1 - r^3[1 - \Gamma(E)]},$$

where

$$\sigma_1(E) = \text{unshielded energy-dependent cross section for the } i\text{-th heavy nuclide;}$$

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

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

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

$$\Gamma(E) = \frac{\overline{\phi}_0}{\overline{\phi}_1} = \frac{1 + \rho Q[1 + W_1(\Sigma_{a,1} + \Sigma_{\text{out},1})]}{1 + \rho Q + W \Sigma_0(\Sigma_{a,0} + \Sigma_{\text{out},0})},$$
where

\[ \rho = \frac{\bar{x}_0}{\bar{x}_1} = \frac{V_0}{V_1} \]

is the volume ratio of regions 0 and 1,

\[ Q = \text{ratio of spatially averaged source densities in regions 0 and 1}, \]

\[ W = 1 + \bar{H}_0(\Sigma_{t,0}) + \bar{H}_1(\Sigma_{t,1}) \]

(6)

\[ \bar{x}_0, \bar{x}_1 = \text{mean chord lengths in regions 0 and 1, respectively}; \]

\[ \bar{x}_j = \frac{4V_j}{\bar{S}_j}, j = 0,1 \]

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

\[ \bar{H}_j(\Sigma_{t,j}) = \frac{1 - P_j}{\bar{x}_j P_j \Sigma_{t,j}}, j = 0,1 \]

(7)

and \( \Sigma_{a,j}, \Sigma_{\text{out},j}, \) 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

\[ \bar{H}_1(0) = \left( \frac{\gamma}{\bar{r}} \right)^2 \left\{ (1 - r^2)^2 \left( 1 + \frac{1}{4} \ln \frac{1+r}{1-r} \right) - \frac{r}{2} (1-r)^2 \right. \]

\[ + \left. \left( \frac{2}{3r} \right)^2 \left[ (1 - r^2)^3 - 3(1-r^3)^2 + 2(1-r^3)(1-r^2)^{3/2} \right] \right\} \]

(8)

where

\[ \gamma = \frac{3r^2}{4(1-r^3)} \]

(9)

The escape probability function \( P_0 \) is given by the expression of Case et al.\textsuperscript{20}

\[ P_0(\Sigma_{t,0}) = \frac{3}{8X^2} [2X^2 - 1 + (1 + 2X) \exp(-2X)] \]

(10)

where
The derivations of these equations and the justifications for the approximations made can be found in Wälti's paper. 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

\[
P^*_{\text{esc}} = P_{\text{esc}} \frac{1 - C}{1 - C(1 - A F F P_{\text{esc}})} ,
\]  

where
\[ P_{\text{esc}} = \text{escape probability from one lump,} \]
\[ C = \text{Dancoff factor (Appendix D), and} \]
\[ \overline{\xi}_F = \text{fuel-rod mean chord length.} \]

Equations for \( P_{\text{esc}} \) for different lump geometries have been derived by many investigators (e.g., see Refs. 20, 21, 22). Wigner\textsuperscript{23} has proposed a "rational" approximation to \( P_{\text{esc}} \) 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\textsuperscript{15} and is given by the following expression

\[ P_{\text{esc}} = \frac{1}{\Sigma_F \overline{\xi}_F} \frac{1}{1 + A} \]

(16)

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

For cylindrical rods, Otter\textsuperscript{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 \( P^*_{\text{esc}} \) is

\[ P^*_{\text{esc}} = \frac{1}{\Sigma_F} \frac{1}{1 + \frac{\Sigma_F}{\Sigma_e}} \]

(17)

where the effective cross section \( \Sigma_e \) is given by

\[ \Sigma_e = \frac{A(1 - C)}{\overline{\xi}_F[1 + C(A - 1)]} \]

(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 $\Sigma_e^{25,26}$. This implies that fuel-rod heterogeneity corrections to homogeneous cross sections can be made by adding $\Sigma_e$ 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 lDX 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,

$P_e = \text{escape probability from one grain for neutrons uniformly and isotropically produced in that homogeneous grain},$

$P_E = \text{escape probability from a homogenized fuel rod for neutrons produced uniformly and isotropically in that fuel rod},$

$f_0 = \text{volume fraction of the grains in one fuel rod},$

$P_F = \text{probability that a neutron incident on a fuel rod collides in that fuel rod},$

$P_M = \text{probability that a neutron leaving a fuel rod collides in the moderator outside that rod},$

$P_0 = \text{probability that a neutron incident on a fuel grain collides in that grain},$

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

$P_E' = \text{neutron escape probability from a fuel rod for neutrons produced in the grains of that fuel rod},$

$P_{ge} = \text{probability that a neutron from the moderator outside any grain will escape from the fuel rod in which that grain is}.$
From these definitions, it immediately follows that

\[ C = 1 - P_M \]  
(19)

and

\[ C_0 = 1 - P_1 \]  
(20)

where

- \( C \) = Dancoff factor of the regular array of fuel rods in the reactor core,
- \( C_0 \) = 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, it also follows that

\[ P_F = \Sigma_F \overline{r}_F P_E \]  
(21)

and

\[ P_0 = \Sigma_0 \overline{r}_0 P_e \]  
(22)

where

- \( \Sigma_0 \) = macroscopic fuel-grain cross section,
- \( \overline{r}_0 = \frac{4V_0}{S_0} \) = mean chord length of a grain of volume \( V_0 \) and surface area \( S_0 \);
  - for a spherical grain of radius \( R \), \( \overline{r}_0 = (4/3)R \).

The overall neutron escape probability is given by:

\[ P_*^E = P'_E \left[ P_M + (1-P_M)(1-P_F)P_M + \cdots \right] = P'_E \frac{P_M}{1 - (1-P_M)(1-P_F)} \]  
(23)

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

\[ P_*^E = P'_E \frac{1 - C}{1 - C(1 - \Sigma_F \overline{r}_F P_E)} \]  
(24)
The rational approximations for $\mathcal{P}_E$ and $\mathcal{P}_e$ are

\begin{equation}
\mathcal{P}_E = \frac{1}{\sum_F \bar{\chi}/(1 + \frac{\sum_F \bar{\chi}}{A})},
\end{equation}

and

\begin{equation}
\mathcal{P}_e = \frac{1}{\sum_0 \bar{\chi}/(1 + \frac{\sum_0 \bar{\chi}}{a})},
\end{equation}

where $A$ is the rod-geometry-dependent Levine factor\textsuperscript{15} with the recommended value of 1.35 for cylindrical rods. Parameter "a" can be obtained by "ration-alizing" Eq. (10) to give

\begin{equation}
\mathcal{P}^{sph}_e \approx \frac{1}{1 + \frac{9}{16} \sum_0 \bar{\chi}/a},
\end{equation}

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

We can evaluate $\mathcal{P}_E'$ from the series:

\begin{equation}
\mathcal{P}_E' = \mathcal{P}_e [P_1 P_{ge} + (1 - P_1)(1 - P_0)P_1 P_{ge} + \ldots] = \mathcal{P}_e \frac{P_1 P_{ge}}{1 - (1 - P_1)(1 - P_0)},
\end{equation}

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

\begin{equation}
\mathcal{P}_E' = \frac{P_{ge}}{1 + \sum_0 \bar{\chi}/(a + \frac{C_0}{1 - C_0})}.\end{equation}

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

\begin{equation}
\mathcal{P}_E' \approx \frac{\mathcal{P}_E}{1 + \sum_0 \bar{\chi}/(a + \frac{C_0}{1 - C_0})}.
\end{equation}
Equations (24), (25), and (30) can be combined to give:

\[
P_E^* = \frac{1}{1 + \sum F \bar{\xi} \left( \frac{1}{a} + \frac{C_0}{1 - C_0} \right) \left[ 1 + \sum F \bar{\xi}_F \left( \frac{1}{A} + \frac{C}{1 - C} \right) \right]},
\]

(31)

which after neglecting second-order terms yields

\[
P_E^* = \frac{1}{1 + \sum F \bar{\xi}_F (A^* + \frac{C}{1 - C})},
\]

(32)

where

\[
\frac{1}{A^*} = \frac{1}{A} + \frac{\bar{\xi}_0 (1 + \frac{C_0}{1 - C})}{A^*}
\]

(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

\[
P_E^* = \frac{1}{\sigma_F} \frac{1}{\sigma_{\text{eff}}}.
\]

(34)

where

\[
\sigma_{\text{eff}} = \frac{1}{N_F \bar{\xi}_F \left( \frac{1}{A} + \frac{C}{1 - C} \right)}.
\]

(35)

$N_F$ = absorber atomic density in the fuel rod. All the $\sigma$'s are microscopic cross sections per absorber atom. The new quantity $\sigma_{\text{eff}}$ can then replace $\Sigma_e / N_F$ 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\textsuperscript{2}-I or IDX 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:

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

where

\[ \Sigma_g = n \bar{\sigma}_0, \tag{37} \]

\[ \Sigma_f = \Sigma_g + \Sigma_{\text{mod}}, \tag{38} \]

\[ \Sigma_{\text{mod}} = N_1 \sigma_1, \tag{39} \]

and

\[ N_1 = \text{atomic density of fuel-rod moderator outside the grains,} \]
\[ \sigma_1 = \text{fuel-rod moderator microscopic cross section,} \]
\[ n = \frac{f_0/V_0}{V_0} = \text{number of grains per unit volume of the fuel rod,} \]
\[ \bar{\sigma}_0 = \frac{S_0}{4} = \text{average "geometric" cross section of the grains,} \]
\[ m = 3.58. \]

If scattering effects in the fuel grains are considered, parameter "a" should be replaced\textsuperscript{28,29} by group parameter a*:

\[ a^* = \frac{a}{1 - q}, \tag{40} \]

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.
Comparisons and Discussion

The above double-heterogeneity space-shielding methods were used for generating above-thermal few-group $^{232}\text{Th}$, $^{235}\text{U}$, and $^{233}\text{U}$ cross sections for a 3000-MW(TH) HTGR system with fuel rods containing 500- and 200-μm-diameter $\text{ThO}_2$ and UC$_2$ grains, respectively, in a graphite matrix. The most affected in the above-thermal region is the $^{232}\text{Th}$ 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 $^{233}\text{U}$ and $^{235}\text{U}$ absorption cross sections ($^{232}\text{Th}$ is not important in the thermal region) was seen to be considerably less important.

### Table III

<table>
<thead>
<tr>
<th>Temperature (K)</th>
<th>1st Method</th>
<th>2nd Method</th>
<th>MICROX</th>
<th>NGSX</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>6.58</td>
<td>6.72</td>
<td>6.76</td>
<td>6.95</td>
</tr>
<tr>
<td>800</td>
<td>7.82</td>
<td>8.03</td>
<td>8.12</td>
<td>8.28</td>
</tr>
<tr>
<td>1200</td>
<td>8.42</td>
<td>8.65</td>
<td>8.78</td>
<td>8.90</td>
</tr>
</tbody>
</table>
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 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. 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

\[ \sigma_{\text{eff}}^{\text{Th}} = \sigma_{\text{Th}} \frac{V_f}{V_c} \frac{\Gamma(E)}{1 + \frac{V_p}{V_c} \Gamma(E)} , \]  

(A-1)

where \( \sigma_{\text{Th}} \) 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

\[ \Gamma(E) = \frac{1 + \frac{V_p}{V_c} Q (1 + \tau_{x,c} W)}{1 + \frac{V_p}{V_c} Q + \tau_{x,p} W} , \]  

(A-2)

where \( p \) refers to the particle region, \( c \) refers to the moderator region, \( \xi \)'s are

\[ \tau_{x,j} = \tau_t,j \left[ 1 - \left( 1 - \frac{\xi_j}{\xi_{\text{pot}}} \right) \frac{\Sigma_s}{\Sigma_{t,j}} \right] \quad j = p,c \]  

(A-3)
the logarithmic slowing-down decrements for each region, and \( \Sigma_s \) and \( \Sigma_t \) 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 \( \xi \), so that this quantity is energy independent in the moderator region.

\[
\tau_{t,j} = \frac{4v_j}{S_j} \Sigma_{t,j}, \quad j = p, c \quad ,
\]

where \( S \) refers to the surface areas of the regions.

\[
W = 1 + \tilde{H}_0(\tau_{t,p}) + \tilde{H}_1(\tau_{t,c}) .
\]

\[
\tilde{H}_0(\tau_{t,p}) = \frac{1 - \tilde{P}_0(\tau_{t,p})}{\tau_{t,p} P_0(\tau_{t,p})} .
\]

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

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

\[
r = R_0/R_1 ,
\]

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

\[
\gamma = \frac{3r^2}{4(1 - r^3)} .
\]

\[
Q = \frac{\xi_{pot}}{\Sigma_{pot}} \frac{\xi_{p}}{\Sigma_{p}} \frac{\xi_{c}}{\Sigma_{c}} .
\]

21
Also, the cross-section weighted logarithmic decrements for the mixtures in each region are given by

\[ \xi^j = \frac{\sum_k \xi_k^j N_k^j \sigma_{sk}}{\sum_k N_k^j \sigma_{sk}} , \]  

(A-12)

where the \( N_k \) are the concentration and \( \sigma_{sk} \) 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:

\[ \sigma_{\text{eff}} = \sum_{i=1}^{n} N_i \sigma_i \]  

(A-13)

where \( \sigma_{\text{eff}} \) 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 \( \sigma_i \) 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>
<thead>
<tr>
<th>Card No.</th>
<th>Format</th>
<th>Variable</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6A10</td>
<td>A(I)</td>
<td>Title card.</td>
</tr>
<tr>
<td>2</td>
<td>6E11.4</td>
<td>RADP</td>
<td>Radius of particle region.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RADC</td>
<td>Radius of moderator region.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMAX</td>
<td>Upper energy bound of resonance region.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMIN</td>
<td>Lower energy bound of resonance region.</td>
</tr>
<tr>
<td>3</td>
<td>6I11</td>
<td>NMP</td>
<td>No. of materials in particle region.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NMC</td>
<td>No. of materials in moderator region.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOVCAL</td>
<td>Obsolete.</td>
</tr>
<tr>
<td>4</td>
<td>6E11.4</td>
<td>PSIP(I)</td>
<td>NMP values of $\xi_i$ for the materials in particle region. Note $I = 1$ is always material for which grain-shielded cross sections are being produced, e.g., Th.</td>
</tr>
<tr>
<td>5</td>
<td>6E11.4</td>
<td>PSIC(I)</td>
<td>NMC values of $\xi_i$ for the materials in moderator region. Note $I = 1$ is always for the moderating material, e.g., c.</td>
</tr>
<tr>
<td>6</td>
<td>6E11.4</td>
<td>CONP(I)</td>
<td>NMP concentrations for the materials in the particle region. Order same as for PSIP.</td>
</tr>
<tr>
<td>7</td>
<td>6E11.4</td>
<td>CONC(I)</td>
<td>NMC concentrations for the materials in the moderator region. Order same as for PSIC.</td>
</tr>
<tr>
<td>8</td>
<td>6E11.4</td>
<td>XSP(I),XP(I)</td>
<td>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.</td>
</tr>
<tr>
<td>9</td>
<td>6E11.4</td>
<td>XSC(I),XC(I)</td>
<td>NMC values for total and potential cross sections for materials in the moderator region. Order same as for PSIC.</td>
</tr>
</tbody>
</table>
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.
PROGRAM PTOPE5 (INP,OUT,FSET5=INP,FSET6=OUT,FSET10,FSET11,FSET12,OUTS)

PURPOSE OF PROGRAM - TO CONVERT A PENDF TAPE TO SHIELDED PENDF. PENDF TO PENDF SHIELDED.

COMMON/CONS/RADC,RADP,VLCP,VLCP,SURC,SURP,PSIP(10),PSIC(10),EMAX

COMMON/CALC/H1TAU,VOLF,G,SEEPS,SEEPS,REDPS,REDPS,RADC,RADC,EMAX

COMMON/CONF/CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10)

COMMON/CONS/RADC,RADP,VLCP,VLCP,SURC,SURP,PSIP(10),PSIC(10),EMAX

COMMON/CALC/H1TAU,VOLF,G,SEEPS,SEEPS,REDPS,REDPS,RADC,RADC,EMAX

COMMON/CONF/CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10),CONF(10)

DIMENSION F(10),S(10),J(10),A(8),HOL(7),X(10),Y(10)

READ (11,15) (A(I),I=1,6),ANEXT1,MCHECK,ANEXT2

WRITE (6,15) (A(I),I=1,6),ANEXT1,MCHECK,ANEXT2

GO TO 4

END FILE 10

REIND 10

REIND 11

FORMAT (6A10s,6A10,6A10,15E11.4)

READ (5,19) NMP,NMC,NOQCAL

READ (5,18) RADP,EMAX,EMIN

FORMAT (6A11,4)

READ (5,19) NMP,NMC,NOQCAL

WRITE (6,7) RADP,RADC,EMAX,EMIN,NOQCAL

FORMAT (6A11)

READ (5,18) CONF(1),CONF(1)

READ (5,17) CONF(1),CONF(1)

READ (5,16) CONF(1),CONF(1)

READ (5,15) CONF(1),CONF(1)

READ (5,14) CONF(1),CONF(1)

READ (5,13) CONF(1),CONF(1)

READ (5,12) CONF(1),CONF(1)

READ (5,11) CONF(1),CONF(1)

READ (5,10) CONF(1),CONF(1)
READ (5+18) (CONC(I),I=1,NMC)
READ (5+19) (XSP(I),I=1,NMP)
READ (5+19) (XSC(I),I=1,NMC)

C THESE CONSTANTS ARE NEEDED IN SUBROUTINE GRANSHL:

VOLP=4/3*3.14159*RADP**3
VOLC=4/3*3.14159*RADC**3 =VOLP
SURP=4/3*3.14159*RADP**2
SURC=SURP
R=RADP/RADC
GAM=3*R**2/(4.0*(1.0-R**3))
TRM1=(1.0-R**2)**2*(1.0+0.25*ALO)
TRM2=(2.0/(3.0*R))*2
TRM3=(1.0-R**2)**3-3.0*(1.0-R**3)**2+2.0*(1.0-R**3)*(1.0-R**2)**1.5

H1AU=(GAM/R)**2*(TRM1+TRM2*TRM3)

VOLF=VOLC+VOLP

CALCULATE Q*

SEENUM=0.
SIGPSp=0.
DO 130 I=1,NMP
SEENUM=PSP(I)*CONP(I)*XP(I)*SEENUM
SIGPSp=CONP(I)*XP(I)*SIGPSp
CONTINUE

Q=SEENUM
SEEP=SEENUM/SIGPSp
SEENUM=0.
SIGPSp=0.
SIGTC=0.
DO 140 I=1,NMC
SEENUM=PS(I)*CONC(I)*XC(I)*SEENUM
SIGPSp=CONC(I)*XC(I)*SIGPSp
SIGTC=CONC(I)*XC(I)*SIGTC
CONTINUE

SEEC=SEENUM/SIGPSp
Q=Q/SEENUM
TAUC=4.0*VLOC*SIGTC/SURC
TAXC=TAUTC

END OF Q CALCULATION

TITL(1)=10HGAMMA PLOT
TITL(2)=10H TO COMPAR
TITL(3)=10HE WITH OTH
TITL(4)=10HR METHODS,
XLB(1)=10H ENERGY IN
XLB(2)=10HF. V. UNITS

READ (11,20) (HOL(I),I=1,7) *MAT, MF, MT, NSEQ
HOL(1)=10H THIS TAP
HOL(2)=10H HAS BEEN
HOL(3)=10H CHANGED T
HOL(4)=10H A PENDF=
HOL(5)=10HSMILEDF F
HOL(6)=10HMILE

FORMAT (AA,10,A6,14*I2,13,I5)
WRITE (12,20) (HOL(I),I=1,7) *MAT, MF, MT, NSEQ
READ (10,20) DUM
READ (10,AN) ZA, AWR
CALL STORXS
NX=0
PRINT 2020,MAT
2020 FORMAT (1H1*WELL, WE MADE IT OUT OF STORXS ONCE, MAT=*I4)
30 READ (11,20) (MOL(I),I=1,7),MAT,MF,MT,NSEQ
IF (MAT.EQ.0) CALL STORXS
IF (MAT.EQ.0) NX=0
2030 FORMAT (1H1*WE ARE LOOPING NOW, MAT=*I4)
WRITE (12,20) (MOL(I) ,I=1,7),MAT,MF,MT,NSEQ
IF (MAT.EQ.1) 00 TO 2000
IF (MF,NE,3) GO TO 30
IF (MT.EQ.2) GO TO 31
IF (MT,EQ.4) GO TO 31
IF (MT,EQ.6) GO TO 31
IF (MT,EQ.12) GO TO 31
GO TO 30
31 CONTINUE
MTXX=MT
READ (11,40) (C(I),I=1,8),MAT,MF,MT,NSEQ
CALL CXFP (C(I) )
CALL CXFP (C(I), ,J(I))
WRITE (12,50) (F(I) ,I=1,6),N1,N2,N3,N4,MAT,MF,MT,NSEQ
40 FORMAT (1PE11.4*4111,I4,I2,I3,I5)
50 FORMAT (2(1PE8.5,A12),4111,I4,I2,I3,I5)
READ (11,60) NPT,INT,NO,N0,MAT,MF,MT,NSEQ
WRITE (6,200) (C(I),MAT,MT)
60 FORMAT (6111,14*I2*I3*I5)
70 NN1=1
70 NN2=NN1+2
READ (11,80) (XI(I),Y(I),I=1,3),MAT,MF,MT,NSEQ
80 FORMAT (1PE14.4*4111,I2,I3,I5)
DO 85 I=1,3
E=X(I)
CALL GRANSHL (E,FAC)
Y(I)=Y(I)-FACT
LOOP=LOOP+1
85 CONTINUE
CALL CXFP (X(I),F(1) )
CALL CXFP (Y(I),F(2) )
CALL CXFP (X(I),F(3) )
CALL CXFP (Y(I),F(4) )
CALL CXFP (X(I),F(5) )
CALL CXFP (Y(I),F(6) )
WRITE (12,90) (F(I) ,I=1,6),MAT,MF,MT,NSEQ
90 FORMAT (1PE8.5,A12),4111,I4,I2,I3,I5)
95 FORMAT (1H M = *16* E = 1PE12.5* FACT = 1PE12.5)
100 FORMAT (1H1** TEMPERATURE = 1PE12.5** MAT = 14** MT = 13)
NN1=NN2+1
IF (NN1.LE.N4) GO TO 70
READ (11,20) (MOL(I) ,I=1,7),MAT,MF,MT,NSEQ
WRITE (12,20) (MOL(I) ,I=1,7),MAT,MF,MT,NSEQ
IF (MTXX.GT.1) GO TO 30
WRITE (6,200) (ENG(N),FAI(N),GAMX(N) ,N=1,NX)
200 FORMAT (1H1** NX = *16* FOR TEMP = 1PE12.5/7X*ENERGY*15X, 1 FACT=13X*GAMMA*)
WRITE (6,210) (ENG(N),FAI(N),GAMX(N) ,N=1,NX)
210 FORMAT (1PE18.5)
WRITE (6) NX,ENG(N),FAI(N),GAMX(N) ,N=1,NX)
GO TO 30
2000 WRITE (6,2000) MAT
2010 FORMAT (1H1** PROCESSING COMPLETE, MAT = *I4)
SUBROUTINE STORXS
STORE TOTAL AND ELASTIC XSEC FOR THORIUM.
LCM/XSECTT/XT(60000)/YT(60000)/NPTT
LCM/XSECTE/XE(60000)/YE(60000)/NPEE
DIMENSION A(7)
10 READ (10,20) (A(I),I=1,7),MAT,MT,NSEQ
20 FORMAT (6A6,14,12,13,15)
   IF (MAT,F0,=1) GO TO 2000
   IF (MF,GT,3) GO TO 300
   IF (MF,LT,3) GO TO 10
   GO TO 10
100 READ (10,30) NPTT
   PRINT 2020,NPTT
   READ (10,20) (A(I),I=1,7)
   IF (NPTT,G0,60000) GO TO 10
   NPTTS=NPTT
105 READ (10,40) (XT(I),YT(I),I=1,NPTTS)
   PRINT 40, (XT(I),YT(I),I=1,99)
   PRINT 2040,NPTTS,XT(NPTTS),YT(NPTTS)
110 IF (NPTT,GT,NPTTS) GO TO 10
120 IF (MT,EQ,0) GO TO 10
   READ (10,20) (A(I),I=1,7),MAT,MT
   GO TO 120
130 NPTTS=60000
   GO TO 105
200 READ (10,30) NPEE
   PRINT 2030,NPEE
   READ (10,20) (A(I),I=1,7)
   IF (NPEE,GT,60000) GO TO 230
   NPEES=NPEE
205 READ (10,40) (XE(I),YE(I),I=1,NPEES)
   PRINT 40, (XE(I),YE(I),I=1,99)
   PRINT 2050,NPEES,XE(NPEES),YE(NPEES)
210 IF (NPEE,GT,NPEES) GO TO 10
220 IF (MT,EQ,0) GO TO 10
   READ (10,20) (A(I),I=1,7),MAT,MT
   GO TO 220
230 NPEES=60000
   GO TO 205
300 READ (10,20) (A(I),I=1,7),MAT,MT,NSEQ
   IF (MAT,EQ,=1) GO TO 2000
   IF (MAT,NE,0) GO TO 300
   RETURN
30 FORMAT (55X,I11)
40 FORMAT (25X*MAT=**) MAT
2000 WRITE(6,2010) MAT
2010 FORMAT (1H1*, SORRY TAPE IS OUT OF TEMPS, MAT=**) MAT
2020 FORMAT (1H1*, 10X,*XT,XT TARL=80X,*NPTT=0*I11)
2030 FORMAT (1H1*, 10X,*XE,YE TARL=80X,*NPEE=0*I11)
2040 FORMAT (1H1*,*NPTT=***E11,4X,*XT(NPTTS)=***E11,4X,*YT(NPTTS)=**E11,4)
2050 FORMAT (1H1*,*NPEE=***E11,4X,*XE(NPEES)=***E11,4X,*YE(NPEES)=**E11,4)
RETURN
END
SUBROUTINE GRANSHL (E, FACT)

PURPOSE = TO CALCULATE SHIELDING FACTOR FOR TWO REGION PARTICLE.

LCM/XSECT/XT(60000), YT(60000), NPTT
LCM/XSECT/XT(60000), YE(60000), NPEE
COMMON/CONS/RAUC, RADP, VOLC, VOLP, SURC, SURP, PSIP(10), PSI(10), EMAX,
1 EMIN, MT
COMMON/CALC/HITAU, VOLF, Q, SEEP, SEEC, SIGPSPI, SIGPSC, TAUCC, TAUXC
COMMON/CON1/CONP(10), CON1(10), XSC(9), XSP(9), NMP, NMC, XP(9), XC(9)
COMMON/PLTS/ENG(5000), FAX(5000), GAMX(5000), NA, TITL(5), XLR(5),
1 YLB(5)

CONDITIONAL RETURNS

IF (E.GT.1.0E+10) GO TO 10
E=0,
FACT=0,
RETURN
10 CONTINUE
IF (E.LT.EMAX) GO TO 20
FACT=1.0
RETURN
20 CONTINUE
QQ=Q
IF (E.LT.EMIN) QQ=0.

FIND TOTAL AND ELASTIC CROSS SECTIONS CORRESPONDING TO ENERGY E.

CALL LOC1(E, ILK, IL0T)
IHIT=IL0T+1
CALL LOC2(E, ILK, ILOE)
IHIE=ILOE+1
DXTL=XT(ILOT)
DYTL=YT(ILOT)
DXTH=XT(IHIT)
DYTH=YT(IHIT)
CALL TSEP1(DXTL, DTYL, DXTH, DYTE, E, ST=2, I)
DXEL=XE(ILOE)
DYEL=YE(ILOE)
DXEH=XE(IHIE)
DYEH=YE(IHIE)
CALL TSEP1(DXEL, DYEL, DXEH, DYEH, E, SE=2, I)

XP(I)=SE
XSP(I)=ST
SIGPSPI=0.
SIGTP=0.
SEENUM=0.
DO 30 I=1, NMP
SEEPI=PSIP(I)*CONP(I)*XP(I)*SEENUM
SIGPSPI=SIGPSPI+SEEPI
SIGTP=SIGTP+CONP(I)*XSP(I)*SIGPSPI
30 CONTINUE

SEEPI=SEENUM/SIGPSPI
TAUTP=4.0*VOLP*SIGTP/SURP
TAUXP=TAUTP*(1.0,0,0-SEEPI/SEEC)*SIGPSPI/SIGTP)
X=3.0*TAUTP/4.0
POTAUX=3.0/(8.0*X*X)+2.0*(1.0+2.0*X)*EXP(-2.0*X)
HOTAUX=(1.0*POTAUX)/TAUTP-POTAUX-1.0
W=1.0*HOTAUX*HITAUX
RHOP=VOLP/VOLC*QQ
UPPER=1.0+RHOP*(1.0+TAUXC=0)
UNDER=1,0*RHO*TAUX**W
GAMMA=UPPER/UNDER
FACT=VOLF/VOLC*(GAMMA/(1.0*VOLP/VOLC*GAMMA))
IF (FACT*GT.0.999) GO TO 40
IF (MT.GT.1) GO TO 40
IF (E.LT.EMIN) GO TO 60
IF (NX.GT.1200) GO TO 40
NX=NX+1
ENG(NX)=
FAX(NX)=FACT
GAMX(NX)=GAMMA
40 CONTINUE
45 FORMAT (IH0*, PLOTS GO ONLY TO *1PE12.5, E, V, N)
RETURN
END

SUBROUTINE CXFP(X,F,S,N)
C**********************************************************CXFP 1
C CONVERT X FOR PUNCHING *CXFP 2
C X = FLOATING POINT NUMBER = F*10.0**N *CXFP 3
C F = 0.999995 LE F .LT 9.999995 *CXFP 4
C S = SIGN (HOLLERITH * OR -) OF EXPONENT CXFP 5
C N = EXPONENT *CXFP 6
C*********************************************************CXFP 7
DATA SP/1H+/vSM/1H-/CXFP 8
IF (X*NE.0,0) GO TO 10
F=0,0
S=SP
N=0
RETURN
10 N=ALOG10(ABS(X))
IF (ABS(X)=1.0) 40,20,20
20 F=X/10.0**N
S=SP
IF (ABS(F)=9.999995) 70,30,30
30 F=F/10.0
N=N+1
GO TO 70
40 N=1-N
F=X*10.0**N
S=SM
IF (ABS(F)=9.999995) 70,50,50
50 F=F/10.0
N=N+1
IF (N) 60,60,70
60 S=SP
70 CONTINUE
RETURN
END
SUBROUTINE TERP1 (X1,Y1,X2,Y2,X,Y,NERR)  

I=INTERPOLATION CODE  

NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.  

ERROR STOPS - 301 (X1=X2,DISCONTINUITY)  
302 (INTERPOLATION CODE IS OUT OF RANGE)  
303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT.)  
304 (X1 AND X2 ARE END PTS. OF THE LINE)  
305 (X+Y) IS INTERPOLATED POINT  
306 INTERPOLATE ONE PT.  

5 XA=X1  
YA=Y1  
XB=X2  
YB=Y2  
X=XP  
II=I  

IF ((XB-XA) .GT.1.E-10) GO TO 7  
IF (X.EQ.XA) Y=YA  
PRINT 6,XA,YA,XB,YB,X,Y,NERR  

6 FORMAT (1H0,* ERROR STOP 301 *1p6E12,5,2I3)  
RETURN  

7 CONTINUE  
10 CALL ERROR (302)  
15 IF (II-5) 20,20,10  
20 GO TO (25,30,35,60,75), I1  
25 YP=YA  
IF (XP.EQ.XB) YP=YB  
GO TO 105  
30 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)  
GO TO 105  
35 IF (XA) 30,30,40  
40 IF (XB) 30,30,45  
45 IF (XP) 50,50,55  
50 CALL ERROR (303)  
55 YP=YA*ALOG(XP/XA)*(YB-YA)/ALOG(XB/XA)  
GO TO 105  
60 IF (YA) 30,30,65  
65 IF (YB) 30,30,70  
70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))  
GO TO 105  
75 IF (YA) 35,35,80  
80 IF (YB) 35,35,85  
85 IF (XA) 70,70,90  
90 IF (XB) 70,70,95  
95 IF (XP) 50,50,100  
100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))  
105 Y=YP  
RETURN  
END
SUBROUTINE TERP1 (X1,Y1,X2,Y2,X,Y,I,NERR)

====INTERPOLATE ONE PT.===

(X1,Y1) AND (X2,Y2) ARE END PTS. OF THE LINE

(X,Y) IS INTERPOLATED POINT

I=INTERPOLATION CODE

NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE
INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.

ERROR STOPs - 301 (X1=X2=DISCONTINUITY)
302 (INTERPOLATION CODE IS OUT OF RANGE)
303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT.)

5 XA=X1
YA=Y1
XB=X2
YB=Y2
XP=X
II=I

IF ((XB-XA) GT 1.E-10) GO TO 7
IF (X.EQ.XA) Y=YA
PRINT 6,XA,YA,XB,YB,X,Y,I,NERR
6 FORMAT (1H0, 10P3E12,5,2I3)
RETURN

7 CONTINUE
IF (II) 10,10,15
10 CALL ERROR (302)
15 IF (II=5) 20,20,10
20 GO TO (25,30,35,60,75), II
25 YP=YA
IF (XP.EQ.XB) YP=YB
GO TO 105
30 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)
GO TO 105
35 IF (XA) 30,30,40
40 IF (XB) 30,30,45
45 IF (XP) 50,50,55
50 CALL ERROR (303)
55 YP=YA*ALOG(XP/XA)*(YB/YA)/ALOG(XB/XA)
GO TO 105
60 IF (YA) 30,30,65
65 IF (YB) 30,30,70
70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))
GO TO 105
75 IF (YA) 35,35,80
80 IF (YB) 35,35,85
85 IF (XA) 70,70,90
90 IF (XB) 70,70,95
95 IF (XP) 80,80,100
100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))
105 Y=YP
RETURN

END

SUBROUTINE ERROR (N)
IOS=9
5 PRINT 10,N
WRITE(99,I9)
10 FORMAT (1I10,10I6)
END
SUBROUTINE LOCT1(X*ilo*loct)
C BINARY SEARCH ROUTINE WRITTEN BY P. SORAN. MODIFIED 10-30-73
C TO GIVE RESULTS IDENTICAL TO EARLIER LOCT ROUTINE.
C THAT IS, FIND X SUCH THAT A(LOCT+1) GT X GE A(LOCT), EXCEPT
C X IS EQUAL TO A(N). IN THAT CASE, LOCT IS SET TO (N-1).
C WHEN X IS NOT BINNABLE, THAT IS WHEN X IS OUTSIDE THE RANGE OF
C A VALUES OR IF A CONTAINS ONLY A SINGLE POINT, THE VALUE LOCT=LOCT1
C IS RETURNED.
LCM/XSECT/A(60000)*YT(60000)*N
IF(N.EQ.1) GO TO 3001
IF(X.LT.A(I)) GO TO 3001
IF(X.GT.A(N)) GO TO 3001
IF((A(N-1).EQ.A(N)) .AND. (X.EQ.A(N))) GO TO 3001
LOCT=1
IF(A(I).EQ.X) RETURN
ilo=1
ISRCH=N
IF(ISRCH.LE.ILO+1)GO TO 3000
I=(ISRCH+ILO)/2
IF(A(I).LT.X) GO TO 2000
ISRCH=I
GO TO 1000
ILO=I
GO TO 1000
RETURN
WRITE (99,10) LOCT
FORMAT (I18)
RETURN
END

SUBROUTINE LOCT2(X*ilo*loct)
C BINARY SEARCH ROUTINE WRITTEN BY P. SORAN. MODIFIED 10-30-73
C TO GIVE RESULTS IDENTICAL TO EARLIER LOCT ROUTINE.
C THAT IS, FIND X SUCH THAT A(LOCT+1) GT X GE A(LOCT), EXCEPT
C X IS EQUAL TO A(N). IN THAT CASE, LOCT IS SET TO (N-1).
C WHEN X IS NOT BINNABLE, THAT IS WHEN X IS OUTSIDE THE RANGE OF
C A VALUES OR IF A CONTAINS ONLY A SINGLE POINT, THE VALUE LOCT=LOCT2
C IS RETURNED.
LCM/XSECT/A(60000)*XT(60000)*N
IF(N.EQ.1) GO TO 3001
IF(X.LT.A(I)) GO TO 3001
IF(X.GT.A(N)) GO TO 3001
IF((A(N-1).EQ.A(N)) .AND. (X.EQ.A(N))) GO TO 3001
LOCT=1
IF(A(I).EQ.X) RETURN
ilo=1
ISRCH=N
IF(ISRCH.LE.ILO+1)GO TO 3000
I=(ISRCH+ILO)/2
IF(A(I).LT.X) GO TO 2000
ISRCH=I
GO TO 1000
ILO=I
GO TO 1000
RETURN
WRITE (99,10) LOCT
FORMAT (I16)
RETURN
END
SUBROUTINE GFPLT
SUBROUTINE TO PLOT GAMMA AND FACT FOR REPORT.
COMMON/CINS/RADC,RADP,VOLC, VOP, SURC, SURP, PSIP(10), PSIC(10), EMAX,
1 EMIN, MT
COMMON/PLTS/E(4),1250), F(4),1250), G(4),1250), NX, TITL(5), XLBS(5),
1 YLB(5)
DIMENSION YLAB(5), YLAB(5), ZLAB(5), NPT(5), X(5000),
1 Y(5000), Z(5000)
REWIND 9
NT=4
DO 10 N=1, NT
READ (9) NX 
10 CONTINUE
NN1=1
X(1)=EMIN
Y(1)=1,
Z(1)=1,
DO 20 N=1, NT
JP=NPT(N)
20 CONTINUE
NN1=NN1+1
X(1)=EMAX
Y(1)=0,
Z(1)=0,
DO 30 N=1, NT
TP=NPT(N)
30 CONTINUE
NN1=NN1+1
X(1)=EMAX
Y(1)=1,
Z(1)=1,
TITL(1)=10H FACT FOR
TITL(2)=10H0-300-950-
TITL(3)=10H3000 DEG K
XLAB(1)=10H ENERGY IN
XLAB(2)=10HE, V UNITS
YLAB(1)=10H FACT OF F
CALL PLOJRB(X*Y, NN1, 1, 0, 0, 0, 0, 1, 0, 1, 0, TITL, 30, XLAB, 20, YLAB, 10)
TITL(1)=10H GAMMA FOR
YLAB(1)=10H GAMMA OF F
CALL PLOJRB(X*Z, NN1, 1, 0, 0, 0, 0, 1, 0, 1, 0, TITL, 30, XLAB, 20, YLAB, 10)
RETURN
END
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

<table>
<thead>
<tr>
<th>HTGR-EOEC Case No.</th>
<th>Temperature (K)</th>
<th>PENDF Temperature Used for Thermal Cross Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>300</td>
</tr>
<tr>
<td>3</td>
<td>600</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>800</td>
<td>950</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>950</td>
</tr>
<tr>
<td>6</td>
<td>1200</td>
<td>950</td>
</tr>
<tr>
<td>7</td>
<td>1500</td>
<td>950</td>
</tr>
<tr>
<td>8</td>
<td>1700</td>
<td>950</td>
</tr>
<tr>
<td>9</td>
<td>2000</td>
<td>3000</td>
</tr>
<tr>
<td>10</td>
<td>2300</td>
<td>3000</td>
</tr>
<tr>
<td>11</td>
<td>2600</td>
<td>3000</td>
</tr>
<tr>
<td>12</td>
<td>3000</td>
<td>3000</td>
</tr>
</tbody>
</table>
### TABLE B-II

**INPUT SPECIFICATIONS FOR ETOGLEN**

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

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

<table>
<thead>
<tr>
<th>Upper Energy Boundary (eV)</th>
<th>Calculated Using Original PENDF Data</th>
<th>Calculated Using Data on Reduced Mesh</th>
<th>% Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.04</td>
<td>566.6</td>
<td>567.5</td>
<td>0.2</td>
</tr>
<tr>
<td>0.10</td>
<td>324.9</td>
<td>328.5</td>
<td>1.1</td>
</tr>
<tr>
<td>0.414</td>
<td>201.6</td>
<td>205.3</td>
<td>1.8</td>
</tr>
<tr>
<td>2.38</td>
<td>232.9</td>
<td>234.6</td>
<td>0.7</td>
</tr>
</tbody>
</table>

#### T = 3000 K

<table>
<thead>
<tr>
<th>Upper Energy Boundary (eV)</th>
<th>Calculated Using Original PENDF Data</th>
<th>Calculated Using Data on Reduced Mesh</th>
<th>% Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.04</td>
<td>566.7</td>
<td>567.5</td>
<td>0.2</td>
</tr>
<tr>
<td>0.10</td>
<td>324.9</td>
<td>328.6</td>
<td>1.1</td>
</tr>
<tr>
<td>0.414</td>
<td>202.1</td>
<td>205.8</td>
<td>1.8</td>
</tr>
<tr>
<td>2.38</td>
<td>232.2</td>
<td>233.4</td>
<td>0.5</td>
</tr>
</tbody>
</table>

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

**Fig. B-2.**
$^{233}$U fission cross section at 3000 K from 0.01 to 2.38 eV.
PROGRAM ETOGLEN (INP=OUT, PUN=FILM, FSET1=1, FSET2=2, FSET5=INP).

1 PROGRAM TO GET GLEN INPUT CROSS SECTIONS FROM PENDF TAPE.

C ENDF/B TO GLEN

C - --- ----

C DIMENSION XFSS(200), XCAP(200)
C DIMENSION FC(200), ED(200), EM(200), INT(10), NPT(10), E(2000), S(2000)
C 1 HOL(10), SM(200), TITL(10), XLAB(10), YLAB(10), ER(200), EMR(200)
C 2 SC(200), SR(200), WI(200), W(2000), WM(200), EWI(200), NECT(10)
C 3 NEMCT(10), SBD(10), SMBD(10), BMIN(12), BMAX(12), DELU(12)

C DIMENSION XP(500), YP(500)

C E. IS BASIC CALCULATED MESH, ED ARE ADDITIONAL POINTS TO BE ADDED.
C E.G. BROAD GROUP MESH BREAK POINTS IN GLEN-RESONANCE PEAKS.
C VALLEYS=FTC, AND EM IS COMMITED MESH.
C FIRST READ BMIN, BMAX, DELU FOR EACH REGION (UP TO 8) FOR
C CALCULATION OF BASIC E MESH = DESCENDING ORDER.
C EMIN IS LOWEST ENERGY BOUND, EMAX IS HIGHEST
C EC ARE BROAD GROUP BREAK POINTS TO BE ADDED TO PENDF MESH
C FOR INTEGRAL CHECK
C EW, WI ARE ENERGY-WEIGHT FUNCTION PAIRS FOR WEIGHTING IF
C INTEGRAL CHECK

C READ (5,30) NUMBIN
C DO 5 N=1,NUMBIN
C READ (5,10) BMIN(N), BMAX(N), DELU(N)
C 5 CONTINUE

10 FORMAT (5I11.4)
C IF (NUMBIN.EQ.1) GO TO 7
C NUMED=NUMBIN-1
C DO 6 N=1, NUMED
C IF (BMIN(N).LE.BMAX(N+1)) BMIN(N)=BMAX(N+1)
C 6 CONTINUE
C 7 NPC=1
C WRITE (6,12) NUMBIN

12 FORMAT (1H1, 17X, 12+20H INPUT ENERGY GROUPS/2X, 9H GROUP NO.
C 1 2X, 14H GROUP MAX (EV), 2X, 14H GROUP MIN (EV), 2X,
C 2 14H ELTHERGY WIDTH)
C DO 8 N=1, NUMBIN
C WRITE (6,13) N, BMAX(N), BMIN(N), DELU(N)
C 8 CONTINUE

13 FORMAT (4X, 13X, 8X, E11.5, 5X, E11.5, 5X, E11.5)
C DO 20 N=1, NUMBIN
C U=0.
C E(NPC)=BMAX(N)
C 11 U=U+DELU(N)
C NPC=NPC+1
C E(NPC)=BMAX(N)/EXP(U)
C IF (E(NPC).LT.BMIN(N)) GO TO 20
C GO TO 11

20 CONTINUE
C E(NPC)=BMIN(N)
C EMIN=BMIN(NUMBIN)
C FMAX=BMAX(1)
C DO 21 N=1, NPC
C SR(N)=N
C 21 CONTINUE

C READ ED MESH FROM CARDS.
READ (5*,30) NPD
30 FORMAT ('6l11')
READ (5*,10) (ED(I)*I=1*NPD)

READ BROAD GROUP ENERGIES. READ FROM HI TO LO.
READ (5*,30) NJFP
READ (5*,10) (EC(I)*I=1 NJFP)

READ IN WEIGHT FCNS. READ IN E-LO TO E-HI. MUST BE LOG-LOG INTERP.*
READ (5*,30) NW
READ (5*,10) (EWI(N)*WI(N)*N=1*NW)
TITL(1)=10HTHERMAL WE
TITL(2)=10HTHERM FUNCT
TITL(3)=10HION (GLEN)
NWPLT=NW
XLAB(1)=10HENERGY(EV)
YLAB(1)=10HTHERM FLUX
CALL PLOJB (EWI*WI*NWPLT,E8,0,0,1,0,1,0,TITL,30,XLAB,10,YLAB)

COMBINE E AND ED TO FORM EM.
CALL UNION (E*ED*NPC,NPD)
REORDER E TO GET EM
DO 25 N=1,NPC
NN1=NPC-N41
EM(N)=E(NN1)
25 CONTINUE
JNPC=NPC-1
NNTST=0
DO 35 JI=1,JNPC
DLTST=(EM(JI+1)-EM(JI))/EM(JI)*100.
IF (DLTST.GT.1.0) GO TO 35
PRINT 36,JI,EM(JI),DLTST
NNTST=NNTST+1
35 CONTINUE
36 FORMAT ('1H0:* DUPLICATE ENERGIES AT *I4,*1PE12.5,* AND *1PE12.5:*')
1 1 PCT DIFF = *F6.3
IF (NNTST*F4.0) PRINT 37
37 FORMAT ('1H0:* THERE ARE NO ENERGY PAIRS WITHIN ONE PERCENT,*)
NPM=NPC
NNTST=0
40 CONTINUE

PUNCH ENERGY MESH FOR GLEN,
DO 45 N=1,NPM
NN1=NPM-N1
E(N)=EM(NN1)
45 CONTINUE
TITL(5)=10ENERGY MES
TITL(6)=10HWF FOR GLEN
PUNCH 210,(TITL(I)*I=5,N6)
PUNCH 150,(E(N),N=1,NPM)

THERE ARE SEVERAL TEMPERATURES ON TAPE. READ NTEMP=NO. OF TEMPS.
READ (5*,30) NTEMP+MAT1
DO 1000 NNT=1,NTEMP
FOR EACH TARGET GET XSEC FOR MT=2, MT=18, MT=102.

DO 900 NMT=1,3
   MF1=3
   MT1=2
   IF (NMT.EQ.2) MT1=18
   IF (NMT.EQ.3) MT1=102
   IF (NMT.EQ.3, AND MAT1.EQ.1155) MT1=107
   50 READ (11,60) (HOL(I), QI=1,7) MAT=MF=M7, NSEQ
   60 FORMAT (A10, A6, I4+IP, I3, I5)
   IF (MAT.EQ.0) GO TO 2000
   IF (MAT.LT.MAT1) GO TO 50
   IF (MAT.GT.MAT1) GO TO 2000
   IF (MF.NE.3) GO TO 50
   IF (MT.NE.MT1) GO TO 50
   WRITE (6,200) MAT=MAT1=MF=MT=MT1=NMT
   1
   3
   2

READ (11,70) C1, C2, N1, N2, NP, NP
   70 FORMAT (1P2, E11, 4) I11
   TEMDS=C1
   READ (11,30) (NPT(I), INT(I), I=1, NR)
   ASSUME THERMAL RANGE IS WITHIN FIRST 2000 PTS ON TAPE.

NPTh=NP
   IF (NP, GT, 2000) NPTh=2000
   READ (11,10) (E(I), S(I), I=1, NPTh)
   10 READ (11,60) (HOL(I), I=1,7), MAT, MF, MT, NSEQ
   IF (MT.NE.0) GO TO 80
   IF (E(NPTh), GT, EMAX) GO TO 82
   WRITE (6,81) NPTh, E(NPTh)
   81 FORMAT (I5, E10.5, *EMAX NOT WITHIN *14,3 PTS, LAST ENERGY = *1PE12.5)
   STOP

CONTINUE
   82

GET XSEC, SM CORRESPONDING TO EM.

DO 100 I=1, NPM
   ILO=LOC(E, EM(I), NPTh)
   IF (ILO.EQ.-1) CALL ERROR(100)
   IH1=ILO+1
   DO 85 J=1, NR
      IF (IH1.LT.E(NPT(J))) GO TO 90
      CONTINUE
      CALL ERROR (200)
      90 CALL TERP (E(ILO), S(ILO), E(IH1), S(IH1), EM(I), CSEC, INT(J))
   SM(I)=CSEC
   100 CONTINUE
   110 FORMAT (1H19, EM = *1PE12.5, SM = *1PE11.4)

CHECK INTEGRALS AND MAKE COMPARISON PLOTS.

CUT OFF MESH POINTS ABOVE EMAX
   KTThRM=0
   DO 120 N=1, NPTh
      IF (E(N), GT, EMAX) GO TO 130
      KTThRM=KTThRM+1
   120 CONTINUE

CONTINUE
   130 CONTINUE

TITL(1)=10HETOGLEN VS
TITL(2)=10H PENDF PTS
REVERSE E-MESH AND ADD BROAD GROUP CUT POINTS

DO 300 NI=NTR
ER(N)=E(N1)
300 CONTINUE
CALL UNION(ER,EC,NPTR,NFGP)
DO 320 N=1,NPTR
IF (ILO,EQ.1) CALL ERROR(200)
CALL TERP(E(ILO),S(ILO),E(IHI),S(IHI),ER(N),CSEC,2)
SR(N)=CSEC
320 CONTINUE
CUT OFF MESH POINTS BELOW EMIN

321 DO 326 N=1,NPTR
IF (ER(N),LT,EMIN) GO TO 327
NSTOR=N
326 CONTINUE
327 CONTINUE
IF (ER(NSTOR),EQ,EMIN) N=NSTOR+1
KTHRM=N PTR
C REORDER LOW TO HIGH
DO 330 N=1,NPTR
E(N)=ER(N)
S(N)=SR(N)
330 CONTINUE
PUT WT FCN ON E AND EM MESHES

340 CONTINUE
DO 350 N=1,NPM
ILO=LOCT(EW,E(N),NW)
IF (ILO,EQ.-1) CALL ERROR(300)
IHI=ILO+1
CALL TERP(EW(ILO),W(ILO),EW(IHI),W(IHI),E(N),WSS,5)
WM(N)=WSS
350 CONTINUE
REVERSE EC MESH

365 CONTINUE
DO 345 K=1,NFGP
SC(K)=EC(K)
345 CONTINUE
DO 346 K=1,NFGP
K=NFGP+K+1
EC(K) = SC(K1)

CONTINUE

FIND BROAD GROUP CUT POINTS IN E AND EM MESHES.

NN1 = 0
DO 370 I = 1, NFGP
DO 360 N = 1, NPM
IF ((EM(N) - EC(I)) .NE. 0.0) GO TO 360
NN1 = NN1 + 1
NECT(NN1) = N
GO TO 370
CONTINUE
CONTINUE
WRITE (6, 3A0) (NECT(N), N = 1, NFGP)

FORMAT (1H100, ** BROAD GROUP CUT POINTS */ /5X, 10I6)

NN1 = 0
DO 400 I = 1, NFGP
DO 390 N = 1, NPM
IF (E(N) .NE. EC(I)) GO TO 390
NECT(NN1) = N
GO TO 400
CONTINUE
CONTINUE
WRITE (6, 3A0) (NECT(I), I = 1, NFGP)
WRITE (6, 405) (TITL(I), I = 3, 4), NNT

FORMAT (1H1**, GLEN POINTS FOR *2A10**, TEMP NUMBER *I2, 1 /* PT, NO., ENERGY*8X, 2 *CROSS SECTION**X*, WEIGHT FUNCTION*)

FORMAT (1H1**, PENDF POINTS FOR *2A10**, TEMP NUMBER *I2, 1 /* PT, NO., ENERGY*8X, 2 *CROSS SECTION**X*, WEIGHT FUNCTION*)

WRITE (6, 406) (N, EM(N), SM(N), WM(N), N = 1, NPM)
WRITE (6, 407) (TITL(I), I = 3, 4), NNT

FORMAT (1H1**, PENDF POINTS FOR *2A10**, TEMP NUMBER *I2, 1 /* PT, NO., ENERGY*8X, 2 *CROSS SECTION**X*, WEIGHT FUNCTION*)

WRITE (6, 406) (N, E(N), SM(N), WM(N), N = 1, NPM)

GET BROAD GROUP XSEC FOR ROTH PENDF AND GLEN DATA.

NBG = NFGP - 1
DO 430 N = 1, NBG
TOP = 0,
DEM = 0,
NE1 = NECT(N)
NE2 = NECT(N + 1) - 1
NEM1 = NECT(N)
NEM2 = NECT(N + 1) - 1
DO 410 J = NE1, NE2
TOP = TOP + (E(J + 1) - E(J)) * (W(J + 1) * SM(J) + W(J) * SM(J)) / 2
DEM = DEM + (E(J + 1) - E(J)) * (W(J + 1) * WM(J) + W(J) * WM(J)) / 2
CONTINUE

FORMAT (1H1**, PENDF POINTS FOR *2A10**, TEMP NUMBER *I2, 1 /* PT, NO., ENERGY*8X, 2 *CROSS SECTION**X*, WEIGHT FUNCTION*)

WRITE (6, 440) (TITL(I), I = 3, 4), NNT

FORMAT (1H1, *BROAD GROUP CUT POINTS */ /5X, 10I6)
1  */ RROAD GROUP ENERGY XSEC FROM PENDF XSEC FROM GLEN DATA*)  ETogl315
450 WRITE (6,450) (EC(I),SBD(I),SMRD(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)=10H IUN (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,-37,0,1,1,TITL(40),XLAB,20,  ETogl344
1 YLAB,20)  ETogl345
C C REORDER FOR GLEN PUNCH  ETogl346
C C DO 140 N=1,NPM  ETogl347
NN1=NPM-N+1  ETogl348
E(N)=EM(NN1)  ETogl349
S(N)=SM(NN1)  ETogl350
140 CONTINUE  ETogl351
C C PUNCH FOR GLEN  ETogl352
C C TITL(1)=10H ELASTIC  ETogl353
TITL(2)=10HCROSS SEC  ETogl354
TITL(3)=10HION FOR MA  ETogl355
TITL(4)=10H MATERIAL *  ETogl356
IF (NMT.NE.1) GO TO 141  ETogl357
PUNCH 200,(TITL(1)*I=1*4),MAT1,TEMDS  ETogl358
PUNCH 150,(S(N),N=1,NPM)  ETogl359
150 FORMAT (1P4E24.12)  ETogl360
200 FORMAT (4A10,I4,** TEMP=*1PE11.4,**DEG K*)  ETogl361
210 FORMAT (4A10)  ETogl362
141 CONTINUE  ETogl363
IF (NMT.NE.2) GO TO 143  ETogl364
DO 142 N=1,NPM  ETogl365
XFISS(N)=S(N)  ETogl366
142 CONTINUE  ETogl367
143 CONTINUE  ETogl368
IF (NMT.NE.3) GO TO 147  ETogl369
DO 144 N=1,NPM  ETogl370
XCAP(N)=S(N)  ETogl371
144 CONTINUE  ETogl372
DO 145 N=1,NPM  ETogl373
145 CONTINUE  ETogl374

$S(N) = XFISS(N) \times XCAP(N)$

\texttt{CONTINUE}

\textbf{TITL}(1) = \texttt{ISO ABSORPT}
\textbf{PUNCH} 200, (\textbf{TITL}(I) \times I = 1, 4) \times \texttt{MAT1} \times \texttt{TEMDS}
\textbf{PUNCH} 150, (S(N) \times N = 1, NPM)
\textbf{XNU} = 1.0
\textbf{IF} (\texttt{MAT1} = 0.1157) \quad \textbf{XNU} = 2.4188
\textbf{IF} (\texttt{MAT1} = 0.1260) \quad \textbf{XNU} = 2.498
\texttt{DO} 146 N = 1, NPM
\textbf{S(N)} = \textbf{XNU} \times XFISS(N)
\texttt{CONTINUE}

\textbf{TITL}(1) = \texttt{NUFISSN}
\textbf{PUNCH} 200, (\textbf{TITL}(I) \times I = 1, 4) \times \texttt{MAT1} \times \texttt{TEMDS}
\textbf{PUNCH} 150, (S(N) \times N = 1, NPM)

\texttt{CONTINUE}

\textbf{REVERSE EC MESH FOR NEXT PASS.}

\textbf{DO} 211 K = 1, NFGP
\textbf{SC(K)} = \textbf{EC(K)}
\texttt{CONTINUE}

\textbf{DO} 212 K = 1, NFGP
\textbf{K1} = NFGP - K + 1
\textbf{EC(K)} = \textbf{SC(K1)}
\texttt{CONTINUE}

\texttt{STOP}

\texttt{WRITE (6, 2010) MAT1, MAT}
\textbf{FORMAT} (1H1, 6) \texttt{SORRY, MAT = *14, NOT ON TAPE, LAST MAT = *14)
\texttt{STOP}
\texttt{END}

\textbf{FUNCTION} \texttt{LOCT(E, EK, N)}
\textbf{C} \texttt{BRACKETS EK IN E SO THAT EK GE E(LOCT) AND EK LT E(LOCT+1)}
\textbf{C} \texttt{IF EK CANNOT BE BRACKETED, LOCT=-1}
\textbf{DIMENSION E(1)}
\textbf{RETURN} \texttt{LOCT=-1} \texttt{IF ARRAY HAS ONLY ONE PT. (AS FOR NPTS IN XSEC).}

\textbf{50 FORMAT} (1H1, 1P8E15.5)
\textbf{IF} (N*LE.1) \texttt{GO TO 10}
\textbf{M} = N - 1
\texttt{DO} 5 I = 1, M
5 \textbf{IF} (EK GE E(I)) \texttt{AND} (EK LT E(I+1)) \texttt{GO TO 15}
\textbf{IF} (E(N-1) GE E(N)) \texttt{AND} (EK EQ E(N)) \texttt{GO TO 10}
\textbf{IF} (E(N) NE EK) \texttt{GO TO 10}
\textbf{LOCT} = M
\texttt{RETURN}
10 \textbf{LOCT} = -1
\textbf{PRINT 50, (E(I) \times I = 1, N), EK}
\texttt{RETURN}
15 \textbf{LOCT} = I
\texttt{RETURN}
\texttt{END}
SUBROUTINE TERPl (X1,Y1,X2,Y2,X,Y,II)

===INTERPOLATE ONE PT.====

(X1,Y1) AND (X2,Y2) ARE END PTS. OF THE LINE

(X,Y) IS INTERPOLATED POINT

I=INTERPOLATION CODE

NOTE - IF A NEGATIVE OR ZERO ARGUMENT OF A LOG IS DETECTED, THE INTERPOLATION IF AUTOMATICALLY CHANGED FROM LOG TO LINEAR.

ERROR STOPS - 301 (X1=X2=DISCONTINUITY)

302 (INTERPOLATION CODE IS OUT OF RANGE)

303 (ZERO OR NEGATIVE ARGUMENT FOR INTERPOLATED PT*)

5 XA=X1
YA=Y1
XB=X2
YB=Y2
XP=X
II=I

IF (XA.EQ.XB) CALL ERROR (301)
IF (II) 10,10,15
10 CALL ERROR (302)
15 IF (II-5) 20,20,10
20 GO TO (25,35,60,75)*II
25 YP=YA
IF (XP.EQ.XB) YP=YB
GO TO 105
30 YP=YA*(XB-XA)*(YB-YA)/(XB-XA)
GO TO 105
35 IF (XA) 30,30,40
40 IF (XB) 30,30,45
45 IF (XP) 50,50,55
50 CALL ERROR (303)
55 YP=YA*ALOG(XP/XA)*YB-YA)/ALOG(XB/XA)
GO TO 105
60 IF (YA) 30,30,65
65 IF (YB) 30,30,70
70 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))
GO TO 105
75 IF (YA) 35,35,80
80 IF (YB) 35,35,85
85 IF (XA) 70,70,90
90 IF (XB) 70,70,95
95 IF (XP) 40,50,100
100 YP=YA*EXP(ALOG(XP/XA)*ALOG(YB/YA)/ALOG(XB/XA))
105 Y=YP
RETURN
END

SUBROUTINE ERROR (N)
IOS=9
5 PRINT 10,N
WRITE(99,N)
10 FORMAT (11H ERROR STOP=16)
END
SUBROUTINE UNION (XU, X, NPU, NP)

FUNCTION OF SUBROUTINE
UNION COMPUTES THE UNION OF INDEPENDENT VARIABLE SETS X(IP), IP=1, NPU, AND PLACES THE UNION INTO XU(IP2), IP2=1, NPU.

DIMENSION XU(2000), KU(200), X(200)

ADD A SET X TO AN EXISTING UNION SET XU
DO 106 IP=1, NPU
   KU(IP)=0
106 CONTINUE
DO 103 IP=1, NPU
   IF (X(IP), LT, XU(NPU)) GO TO 120
   IF (X(IP), GT, XU(1)) GO TO 130
   DO 104 IP=1, NPU
      IF (X(IP), EQ, XU(IP1)) GO TO 140
      IF (IP1, EQ, NPU) GO TO 105
      IF (X(IP), LT, XU(IP1), AND, X(IP), GT, XU(IP1+1)) GO TO 150
105 CONTINUE
104 CONTINUE
HERE NPU IS INCREMENTED BY ONE AND A POINT IS ADDED TO THE LEFT
   NPU=NPU+1
   XU(NPU)=X(IP)
   KU(NPU)=1
   CONTINUE
   GO TO 103
HERE CONTROLS ARE SET TO ADD A POINT ON THE RIGHT
   KONREL=1
   NPMOV=NPU
   GO TO 170
HERE NPU IS NOT INCREMENTED BY ONE
   CONTINUE
   KU(IP1)=1
   GO TO 103
HERE NPU IS INCREMENTED BY ONE AND CONTROLS ARE SET TO ADD A POINT BETWEEN POINTS IP1 AND IP1+1
   KONREL=2
   NPMOV=NPU-IP1
   GO TO 170
HERE WE INCREMENT NPU BY ONE AND MOVE THE LEFT MOST NPMOV POINTS SET ONE POSITION TO THE LEFT
   NPU=NPU+1
   DO 171 IP2=1, NPMOV
      XU(NPU-IP2+1)=XU(NPU-IP2)
      KU(NPU-IP2+1)=KU(NPU-IP2)
   171 CONTINUE
HERE A NEW POINT IS ADDED
   NPADD=NPU-NPMOV
   XU(NPADD)=X(IP)
   KU(NPADD)=1
   CONTINUE
103 CONTINUE
102 RETURN
END
APPENDIX C

MERGFAT

A code to merge fast and thermal cross section sets.

Multigroup cross-section data sets for energy groups above the thermal boundary energy (2.58 eV for the HTGR) are generated by the lDX 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 (\(\chi\)), 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**

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

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6E12.5</td>
<td>GPEN(I) NOBG values of lower group bounds in eV.</td>
</tr>
<tr>
<td>6</td>
<td>6E12.5</td>
<td>XI(I) NOBG values of X.</td>
</tr>
<tr>
<td>7</td>
<td>1216</td>
<td>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.</td>
</tr>
<tr>
<td>8</td>
<td>A6,2X,A10,A6</td>
<td>NUCLE Nuclide identifier assigned in lDX input (see Ref. 14).</td>
</tr>
<tr>
<td></td>
<td>MODER</td>
<td>Moderator, absorber identifier. Use word &quot;MODERAT&quot; for moderator and &quot;ISOTOPE&quot; for absorber.</td>
</tr>
<tr>
<td></td>
<td>MATID</td>
<td>ENDF/B MAT number.</td>
</tr>
</tbody>
</table>

The "fast" data file output by the lDX 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
   FSET9,FSET10)  MERGF 2
C MERGFAT IS A VERSION OF JUMRLFAT THAT ACCEPTS FAST XSEC AS
C OUTPUT BY THE MONEBDX VERSION OF THE lDX CODE.
C REVISIONS MADE AT LASL BY LABAUVE,NOV75.
C
DIMENSION TOTP0(70),TOTP1(70),TOTRA(70)
DIMENSION NUCID(20),N2N(20),SK2KF(70),70),*DIAG5N(70),*CHKSM(70),
1  DIFF(70),*CAPA(70),*FISA(70),*STR(70),*AVNU(70),*SINTRA(70),70),
2  SCAP(70),*SS(70),*SABS(70),*HOL(70),*KK(70),*SSN2N(70),70),
3  XNUS(70),*A(2000),*P0(70),70),*P1(70),70),*P2(70),70),*P3(70),70),
4  XS(70),*TITLE(12),*TOTN2N(70),*SNP(70),*SND(70),*SNT(70),*SNHE3(70),
5  SNA(70),*SN2A(70)
DIMENSION XS(50),*GPEN(50),*TTL(12),*VEL(50),*FACAP(50),*FAFIS(50),
1  AOEN(20),*TOTIN(70)
DIMENSION KG1(20),*KG2(20)
DIMENSION V(V(54),*HLT(10)
NTPF=9  MERGF 14
NTPT=10  MERGF 15
5002 READ(8)5001,(HLT(I),I=1,9)  MERGF 16
5001 FORMAT(9(9A8))  MERGF 17
PRINT 600  MERGF 18
600 FORMAT (I120X10HINPUT DATA ///)  MERGF 19
READ (8) LENG,NOELU,LNLGUP  MERGF 20
PRINT 601,ENG,NODELUS,LNLGUP  MERGF 21
```

49
601 FORMAT (1H0,10X$) NDELU = $16,8H, LTL = $16,  
      110H, LUNIP = $16, READ (8,70) (TITLE(I), I=1,7) 
      PRINT 602, (TITLE(I), I=1,7)  
602 FORMAT (1H0,5X$) READ (8,71) NOBG, NOI, IOPT  
      PRINT 603, NOBG, NOI, IOPT  
603 FORMAT (1H0,10X$) NOI = $12,8H, NOI = $12,9H, IOPT = $12) 
      NMAT=NOI  
71 FORMAT (3$I2) 
    KGROPS=NOIG 
70 FORMAT (1A10) 
      PUNCH 1000, (TITLE(I), I=1,12) 
      READ (8,3) (OPEN(I), I=1,KGROPS) 
      PRINT 604  
604 FORMAT (1H0,5X$) PRINT 605, (I, OPEN(I), I=1,KGROPS)  
7 FORMAT (1H0,10X$) PRINT 606  
606 FORMAT (1H0,5X$) PRINT 607, (I, XI(I), I=1,KGROPS)  
      V(I)=1.0*0.7  
      DO 4700 K=1,KGROPS  
4700 FORMAT (6H0)  
      V(K+1)=GPEN(K)  
      DO 72 K=1,KGROPS  
      VEL(K)=0.07*(V(K)*0.5+V(K+1)*0.5) 
500 FORMAT (A12,6F12.6,1A6)  
      VEL(K)=0.07*(V(K)*0.5+V(K+1)*0.5) 
500 FORMAT (A12,6F12.6,1A6)  
      --- NOTF --- THESE ARE JUST AVERAGE BROAD GROUP VELOCITIES, 
C     12=I1+5 
      IF (I2.LE.KGROPS) GO TO 74 
      PUNCH 75, (XI(I), I=1,KGROPS) 
75 FORMAT (6$F12.6) 
      GO TO 76 
      PUNCH 87, (XI(I), I=1,I2), LRL1,M1 
      M1=M1+1 
      I1=I2+1 
      GO TO 73 
76 CONTINUE 
      I1=1 
      M1=1 
57 I2=I1+5 
      IF (I2.LE.KGROPS) GO TO 78 
      PUNCH 79, (VEL(I), I=1,KGROPS) 
79 FORMAT (1$A12,2) 
      GO TO 1100 
    78 PUNCH 81, (VEL(I), I=1,I2), LRL2,M1 
      M1=M1+1 
      I1=I2+1 
      GO TO 77 
1100 CONTINUE 
87 FORMAT (6$F12.6,1A6,12) 
81 FORMAT (1$A12,2,1A6,12) 
5 FORMAT (12$I10) 
      MATNO=1 
      READ (8,5) NDKS, (KG1(N), KG2(N), N=1,NDKS) 
      PRINT 607, NDKS
607 FORMAT (1H10X7HNDKS = $16)
 PRINT 608
608 FORMAT (4H0 N3X6HKG1(N)7X6HKG2(N))
 PRINT 609 (N,NKG1(N),NKG2(N),N=1,NDKS)
609 FORMAT (I3,2I6)
 DO 999 M=1,NMAT
 REWIND NTPF $ REWIND NTPT
 READ (8,A100) NUCLE,MODER, MATID
4100 FORMAT (A6,2X,A10,A6)
 DO 500 N=1,NDKS
 K1=NKG1(N)
 NOBG=NKG2(N)
 LTAG=ENG*6
 NOBG2=NOBG+1
 FAFIS(N)=1,0
 FACAP(N)=1.

2102 IF (CACP(N)=19)
 RETURN

85 FORMAT (15H G=03,13H L=03,13H NUCLIDE NO.,=05,1H 3A10)
4110 IF (N.EQ,4115) READ (NTPF,A115) NUCID(M),FID(TT,(TTL(KK),KK=1,11)
4115 FORMAT (13A6)
 IF (NUCID(M),NE,NUCLE) GO TO 4110
1085 FORMAT (A6,F6.2,11A6)
 DO 3070 K=K1,NOBG
 DO 3070 KF=K1,NOBG
 P0(K,KF)=0.0
 SINTRA(K,KF)=0.0
 SSN2N(K,KF)=0.0
 CONTINUE
3070 CONTINUE
 IF (N.EQ,1) GO TO 3000
 GNU=AVNU(K1)
 MTBL=2*ENGS+5
 MGPS=NOBG-K1+1
 MVLG=MTBL*MGPS
 DO 3010 L=1,2
4120 READ (NTPF,A130) MODE*MATCH
4130 FORMAT (23X,A10,A6)
 IF (MODE,NE,MODER,AND, MATCH,NE, MATID) GO TO 4135
 GO TO 4120
115 CONTINUE
4135 CONTINUE
 IF (L.EQ,1) MADD=0
 IF (L.EQ,2) MADD=MVLG
 READ (NTPF,3) (A(M*MADD),M=1,MVLG)
3010 CONTINUE
 DO 3020 MG=1,MGPS
 DO 3030 MGM=1,MGPS
 KK1=K1+MG=1
 KK2=K1+MG=1
 LCK=9-MG*(MG-1)*(MTBL+1)
 P0(KK1,KK2)=A(LCK)
 PL(MG,MGM)=A(LCK*MVLG)
3030 CONTINUE
 PRINT 3040,KK1
3040 FORMAT (1H10X SCATT, MATRIX FOR GP *)
 PRINT 3050,P0(KK1,KK2)*KK2=K1,NOBG)
 PRINT 3050,P1(MG,MGM)=MG=1,MGPS
 LCK=MG=1*MTBL+1
 SABS(KK1)=A(LCK)
 XNUSIG(KK1)=A(LCK+1)
 STR(KK1)=A(LCK+2)
 AVNU(KK1)=GNU
 IF (GNU,0,E,0) GNU=1,E=10
 FISA(KK1)=XNUSIG(KK1)/GNU
 CAPA(KK1)=SABS(KK1)=FISA(KK1)
PITOT=0.0

DO 3060 MGM=IsMQPS
P1TOT=P1TOT+P*(MG*MGM)
CONTINUE
PRINT 3050,P1TOT
P0(KK1*KK1)=P0(KK1*KK1)+P1TOT
STR(KK1)=STR(KK1)+P1TOT
CONTINUE
GO TO 3050
CONTINUE
PRINT 3000,NUCID(M),FID,(TTL(KK)*KK=1,11)
1 FORMAT (8A10)
DO 10 K=1,NOBG
KF=K-1
IF(K*EQ.1) KF=NOBG2
READ (NTPF,3) FISA(K),SABS(K),XNUSIG(K),STR(K),P0(K,K),P0(K,F,K)
READ (NTPF,3) ZILCH
IF(ZILCH*EQ.0,0) GO TO 20
PRINT 19,ZILCH
19 FORMAT (1H10, ZILCH = *1PE12.5,* READ ERROR*)
CONTINUE
3 FORMAT (6F12.5)
CAPA(K)=SARS(K)*FISA(K)
IF(FISA(K)*EQ.0.0) AVNU(K)=XNUSIG(K)/FISA(K)
CONTINUE
3500 CONTINUE
500 CONTINUE
L0D=-1
DO 95 K=1,NOBG
T0TP0(K)=0
T0TP1(K)=0
T0TP2(N(K))=0
S0(K)=0
XNUSIG(K)=FISA(K)*AVNU(K)
DO 80 KF=K,NOBG2
T0TP2(N(K))=T0TP2(N(K))+SINTRA(K,KF)
IF(L0D*EQ.-1) SK2KF(K,KF)=P0(K,K,F,K)+SINTRA(K,K,F,K)+S02N(K,K,F,K)*2.
S0(K)=S0(K)+SK2KF(K,KF)
T0TP2(N(K))=T0TP2(N(K))+S02N(K,K,F,K)
80 CONTINUE
CAPA(K)=CAPA(K)*SNP(K)*SNA(K)
SABS(K)=CAPA(K)*FISA(K)
95 CONTINUE
L0D1=L0D+2
H0L(2)=10H ET0G=ENDF
GO TO (110*120*130*140*150)*L0D1
110 H0L(1)=10H IS0T0RP T
H0L(2)=10H RNSP TABLE
GO TO 160
120 H0L(1)=10H P=0 TABLE
GO TO 160
130 H0L(1)=10H P=1 TABLE
GO TO 160
140 H0L(1)=10H P=2 TABLE
GO TO 160
150 H0L(1)=10H P=3 TABLE
160 CONTINUE
L0D=L0D+1
L=LENG
N=LENG-1
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.

DO 90 K=1,NOBG
  MN=LENG*K
  IF (MN+GT,NOBG+1) GO TO 90
  DO 91 KF=MN+NOBG
  SK2KF(K,MN+1)=SK2KF(K,MN+1)*SK2KF(K,KF)
  90 CONTINUE
  MK=NOBG+LENG+2
  DO 51 K=MK,NOBG
     Q1=SK2KF(K,NOBG)+SK2KF(K,SKF(K)-1)+SK2KF(K,KF)
     51 CONTINUE
  MK=NOBG+LENG+2
  DO 360 J7=1,NCX6
     360 CONTINUE
  LTABL=LENG+LNGUP*7
  NCX=LTABL*NOBG
  NCX6=NCX+LTABL
  DO 360 J7=1,NCX6
     360 CONTINUE
  A(J7)=0.
  DO 361 J=1,NOBG
     361 CONTINUE
  IF (KFL=LENG) GO TO 361
  CONTINUE
  IF (FID,NE,6H12,00) GO TO 361
  IF (KF=K+1) GO TO 361
  KUP1=K+1
  IF (KUP1,GT,NOBG) GO TO 361
  DO 361 KF=KUP1,NOBG
     361 CONTINUE
  PUNCH 85,N0BG,LTABL,MA1N0,NUCID(M),MOL(I),I=1,2
  NP1=1
  NBR=1
  63 NP2=NP1+5
  IF (NP2,LE,NCX) GOTO 67
  PUNCH 65,(A(NP),NP=NP1,NP2,NUCID(M),NBR
  68 FORMAT (1P6E12.5,9A6,1Z)
  GO TO 64
  67 PUNCH 65,(A(NP),NP=NP1,NP2,NUCID(M),NBR
  IF (NP2,LE,NCX) GOTO 64
  NBR=NBR+1
  NP1=NP2+1
  GO TO 63
  64 CONTINUE
  65 FORMAT (1P6E12.5,9A6,1Z)
  DO 103 K=1,NOBG

103  KK(K)=K
   IF(M.NE.I) GOTO 9
   WRITE(6,1000)(TITLE(I),I=1,7)
1000  FORMAT(6A10)
   9 CONTINUE
   WRITE(6,95) NOBG,LTABL,NUCID(M),HOL(I),I=1,2
   C  CHECK TO ADD UP SK2KF
   C  THE DIAGONAL SUM OF THE DOWN SCATTERING AND SELF SCATTERING ADDED TO
   C  ABSORPTION MUST EQUAL THE TRANSPORT CROSS SECTION.
   DO 400 K=1,NOBG
      DIAGSM(K)=0.
      KK=K+1
      IF (KK.GT.NOBG) KK=NOBG
      DO 450 KF=K,NOBG
         450 DIAGSM(K) = DIAGSM(K)*SK2KF(K,KF)
      END
      CHKSM(K) = DIAGSM(K)*SABS(K)-TOTN2N(K)
      DIFF(K) = CHKSM(K)-STR(K)
      WRITE(6,460)
460  FORMAT (200,TRANSPORT,18HCHECKSUM
        1 10MDIFFERENCE)
      WRITE(6,470) STR(K),CHKSM(K),DIFF(K)
470  FORMAT(E12.4,E12.4,E12.4)
      WRITE(K),LE,1*STR(K) GOTO 400
      WRITE(6,480)
480  FORMAT(31HMDIFFERENCE EXCEEDS 10 PER C\%NT)
      CONTINUE
      WRITE(6,45) NOBG,LTABL,NUCID(M),HOL(I),I=1,2
      DO 701 I=1,NOBG
         K=LTABL*(I-1)
         DO 701 J=1,LTABL
            701 SK2KF(I,J)=A(K*J)
         END
      KA=A s KB=A
      KC=MIN0(KA,NOBG)
      WRITE(6,721) (K,K=KA,KC)
      WRITE(6,721)
      DO 722 J=1,LTABL
         722 WRITE (6,73) J,(SK2KF(I,J),I=KA,KC)
            KA=KA+8 s KB=KB+8
            IF(KA.LE.NOBG) GO TO 724
            WRITE(6,773) J,(SK2KF(I,J),I=KA,KC)
999  CONTINUE
   MATNO=MATNO+1
   IF(HLT(1).NE.8HLASTDECK) GOTO 5002
END
APPENDIX D

DANCOFF FACTOR FOR A REGULAR ARRAY OF CYLINDRICAL FUEL RODS

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

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

\[ C = \frac{1}{\alpha_0} \int_{0}^{\alpha_0} \frac{1}{2r} \int_{-r}^{r} \frac{K_3(t)}{K_1(0)} \, dt \, dr , \]  

(D-1)

where \( K_3 \) is the Bickley function, \( t \) is the optical length between rods, \( r \) is the radius of one rod, and \( \alpha_0 \) 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 \( \bar{\lambda} \) in the moderator through

\[ \bar{\lambda} = \frac{4V}{S} = \frac{2d}{\sqrt{3} \pi} \left( \frac{d^2}{2} - \pi y^2 \right) . \]  

(D-2)

Sauer has found a good approximation for \( C \):

\[ C = \frac{e^{-t \Sigma \bar{\lambda}}}{1 + (1-t) \Sigma \bar{\lambda}} , \]  

(D-3)

where \( \Sigma \) is the moderator cross section and, for a hexagonal lattice

\[ t = \frac{\pi}{2} y \left( \frac{1 - 2y}{\sqrt{3} - \pi y^2} - 0.12 \right) . \]  

(D-4)

Bonalumi 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.
Bonaiumi has, therefore, suggested the following modification:

\[ C = \frac{e^{-t \Sigma t}}{1 + \frac{1 - t_1 \Sigma t}{t_1}} \]  

(D-5)

where

\[ t_1 = t + \frac{\Sigma t}{7 + \beta \Sigma t} \]  

(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 CYLINDERS DANCP 2
C INPUT QUANTITIES DANCP 3
C NALF AND NRAO DETERMINE AN INTEGRATION GRID FOR THE CARLVIK DANCP 4
C INTEGRATION, THEY ARE BOTH TAKEN TO BE 128, DANCP 5
C LATTICE AND 6 FOR A HEXAGONAL LATTICE, DANCP 6
C IF IOPTC=0, ALL THREE METHODS ARE COMPARED, IF IOPTC=1, DANCP 7
C THE CARLVIK ROUTINE IS USED, IF IOPTC=2, THE BONALUMI DANCP 8
C APPROXIMATION ONLY IS USED, IF IOPTC=3, THE SAUER DANCP 10
C APPROXIMATION ONLY IS USED, DANCP 11
C RAD0 IS THE PIN RADIUS IN CM, DANCP 12
C RAD1 IS THE MODERATOR RADIUS IN THE THREE-REGION MODEL, DANCP 13
C GAPWID IS THE GAP WIDTH (CM) AROUND THE PIN, DANCP 14
C RAD1S IS THE MODERATOR RADIUS WHEN THE GAP WIDTH IS NOT DANCP 15
C EXPLICITLY GIVEN AS IN THE SAUER OR THE BONALUMI APPROXIMATIONS, DANCP 16
C SIGF IS THE MACROSCOPIC FUEL-PIN CROSS SECTION (1/CM), DANCP 17
C DENSF IS THE ATOMIC CONCENTRATION OF THE FUEL PIN, DANCP 18
C SIGMAM IS THE MODERATOR MACROSCOPIC CROSS SECTION (1/CM), DANCP 19
READ 5,NLAT,NALF,NRAO,IOPTC,RAD1S DANCP 20
5 FORMAT(4T10,2E10.4) 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) NRAO=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,NRAO,IOPTC,RAD1S DANCP 28
15 FORMAT(1H0,*NLAT = *,NALF = *,NRAO = *,IOPTC = *) DANCP 29
114,* RAD1S = *,E10.4) DANCP 30
PRINT 16,RAD0,RAD1,GAPWID,SIGMAM,SIGF,DENSF DANCP 31
16 FORMAT(1H0,*RAD0 = *,E12.6,* RAD1 = *,E12.6,* GAPWID = *,E12.6/* DANCP 32
10 SIGMAM = *,E12.6,* SIGF = *,E12.6,* DENSF = *,E12.6) DANCP 33
IF(IOPTC).EQ.20,20,30 DANCP 34
```
DO 25 IOPTC=1,3
CALL DANCOFF(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC,CS,
1CSB,C,IOPTC,RAD1S)
CALL LEVINE(SIGF,DENSF,C,RAD0,IOPTC)
25 CONTINUE
GO TO 50
CALL DANCOFF(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC,CS,
1CSB,C,IOPTC,RAD1S)
CALL LEVINE(SIGF,DENSF,C,RAD0,IOPTC)
50 CONTINUE
END

SUBROUTINE LEVINE(SIGF,DENSF,C,RAD0,IOPTC)
CALCULATES THE EFFECTIVE GEOMETRIC SHIELDING CROSS SECTION BY THE
LEVINE METHOD AND USING THE OTTER APPROXIMATION FOR THE LEVINE
FACTOR
ELBARF=2.*RAD0
TAUF=SIGF*ELBARF
IF(TAUF.LE.0.) PRINT 20
20 FORMAT(1X,0,TAUF IS G, OR E, TO ZERO*)
IF(TAUF.GE.2.) GO TO 30
ALEVI=1.0134*1.4879*TAUF**0.5-0.17226*TAUF
GO TO 40
30 ALEVI=1.0+1.*TAUF=1.*TAUF**3.
40 CONTINUE
PRINT 50,ALEVI
50 FORMAT(1X,ALEVI)
CONTINUE
PEN
SUBROUTINE DANCOFF(NLAT,RAD0,RAD1,GAPWID,\ SIGMAM,NALF,NRAD,CC,CS,
1CSB,C,IOPTC,RAD1S)
CALCULATES DANCOFF FACTORS BY THE ORIGINAL METHOD AS USED BY
CARLVIK, BY THE SAUER APPROXIMATION AND BY THE BONALUMI
APPROXIMATION AND COMPARES THE RESULTS OF THE THREE
IF(IOPTC).LT.10*10*20
10 PRINT 15
15 FORMAT(1X,0,IOPTC IS ZERO OR NEGATIVE, TRY AGAIN*)
RETURN
END
SUBROUTINE CARLVK(NLAT,RAD0,RAD1,GAPWID,SIGMAM,NALF,NRAD,CC)
C CALCULATES DANCOFF FACTORS BY THE ORIGINAL METHOD AS IMPLEMENTED
C
PI=3.141592654
GAM=0.0
IF(NLAT<6) GAM=PI/60.0
CNST1=(1./PI)**0.5
CNST2=(3.**0.5)**0.5
IF(NLAT<6) CNST1=CNST1*CNST2
PITCH=RAD0/PITCH
E=(RAD0+GAPWID)/PITCH
E2=E**2
CONST=2.0/(PI*NALF*NRAD)
SIG =SIGMAM*PITCH
CC=0.0
ISIG=10./SIG+1.
NRAD=MINO(100,ISIG)
I=2*NRAD
DZ=R/NRAD
DALF=PI/(NLAT*NALF)
ALF=0.5*DALF
DO 60 N=1,NALF
ALF=ALF+DALF
CAG=COS(ALF*GAM)
DX=COS(GAM)/CAG
DY=SIN(ALF)/CAG
T=SIN(ALF*GAM)/CAG
Z=R-0.5*DZ
DO 50 I=1,I1
Z=Z+DZ
X=Z*T-SORT(E2-Z*Z)
F=CAG-Z
IF(F.GE.E) GO TO 10
IF(F.LE.E) GO TO 40
X=X-2.0*SORT(E2-F**2)
60 CONTINUE
50 CONTINUE
10 CONTINUE
SUBROUTINE SAUER(RAD0,RAD1,NLAT,SIGMAM,CS)
C
CALCULATES DANCEFF FACTORS BY THE SAUER APPROXIMATION
PI=3.141592654
RADRA=RAD0/RAD1
VOLRA=VOLRA/RADRA
VOLSQR=(1.*VOLRA)**0.5
IF(NLAT,FQ.4)TAU=((PI/(4.*0.5*VOLSQR-1.))/VOLRA-0.08
IF(NLAT,FQ.6)TAU=((PI/(3.*0.5*2.*))**0.5*VOLSQR-1.)/VOLRA-0.12
IF(TAU)<0.10)
10 PRINT 15
15 FORMAT(1H0,*TAU IS ZERO,NLAT IS WRONG*)
RETURN
20 ELBARF=2.*RAD0
ELBARM=E®BARF*VOLRA
PROD=SIGMAM*ELBARM
DANCOF=EXP(-TAU*PROD)/(1.+((1.-TAU)*PROD)
CS=DANCOF
RETURN
END
SUBROUTINE BONAL(RADO, RAD1, NLAT, SIGMA, CSB)
C CALCULATES DANCOFF FACTORS BY THE BONALUMI APPROXIMATION
TAU = 0.
PI = 3.141592654
RADRA = RAD1/RADO
VOLRA = RADRA * RADRA - 1.
VOLSQR = (1.0 + VOLRA)**0.5
IF(NLAT.EQ.4) TAU = ((PI/4.0)**0.5 * VOLSQR - 1.0) / VOLRA
IF(NLAT.EQ.6) TAU = ((PI/(3.0**0.5) * VOLSQR - 1.0) / VOLRA
IF(TAU).LT.10.0, 10, 20
10 PRINT 15
15 FORMAT(1X, 0. ** TAU IS ZERO, NLAT VALUE IS WRONG*)
RETURN
20 ELBARM = 2.0 * RADO
   ELBARM = ELBARM * VOLRA
   PROD = SIGMA * ELBARM
   IF(NLAT.EQ.4) BETA = 5.67
   IF(NLAT.EQ.6) BETA = 2.125
   DELTAU = PROD / (7.0 + BETA * PROD)
   TAU1 = TAU + DELTAU
   DANCOF = EXP(-TAU*PROD) / (1.0*(1.0 + TAU1)*PROD)
   CSB = DANCOF
   RETURN
END

SUBROUTINE BKLY(X, BIC3)
C CALCULATES BICKLEY FUNCTIONS OF THE THIRD ORDER
A0 = 0.9379348841
A1 = 1.194191634
A2 = 0.588245154
A3 = 0.570337193
A4 = 1.5701166
A5 = 4.92269
B0 = 0.7276787664
B1 = 0.9254650857
B2 = 0.4741520763
B3 = 0.2508258355
B4 = 0.029330079
B5 = 0.055707999
C0 = 0.4166740874
C1 = 0.5295653111
C2 = 0.2754273045
C3 = 0.1253775092
C4 = 0.0119191487
C5 = 0.0130209543
D0 = 0.221540159
D1 = 0.0984379097
D2 = 0.014732145
D3 = 0.005857650032
E0 = 0.282473681
E1 = 0.2354370335
F2 = 0.0630205186
E3 = 0.01340032364
FO = 1,012074180
F1 = -4.000325432
F2 = -1.1646323
F3 = 1.3073944
F4 = -4.4653208
x2 = x * x
x3 = x2 * x
x4 = x3 * x
x5 = x4 * x
IF (x) 10, 20, 20
10 PRINT 15
15 FORMAT (1H0, "* x is less than zero, try again!")
RETURN
20 IF (x = 0, 1) 30, 40, 40
30 SUM = A0 + A1 * x + A2 * x2 + A3 * x3 + A4 * x4 + A5 * x5
BIC3 = 0.7366554521/SUM
RETURN
40 IF (x = 0.4) 50, 60, 60
50 SUM = B0 + B1 * x + B2 * x2 + B3 * x3 + B4 * x4 + B5 * x5
BIC3 = 0.514977571/SUM
RETURN
60 IF (x = 1.0) 70, 80, 80
70 SUM = C0 + C1 * x + C2 * x2 + C3 * x3 + C4 * x4 + C5 * x5
BIC3 = 0.3275473766/SUM
RETURN
80 IF (x = 2.5) 90, 100, 100
90 BIC3 = (D0 + D1 * x + D2 * x2 + D3 * x3) / (E0 + E1 * x + E2 * x2 + E3 * x3)
RETURN
100 Y = 1 / (X + 3, 25)
SUM = F0 + F1 * x + F2 * x2 + F3 * x3 + F4 * x4
BIC3 = 1.2 * 4.58824 * EXP (-X) / (Y = 0.5 * SUM)
RETURN
END
REFERENCES


