A FORTRAN IV TWO-DIMENSIONAL DISCRETE ANGULAR SEGMENTATION TRANSPORT PROGRAM

by Clayton E. Barber

Lewis Research Center
Cleveland, Ohio
A FORTRAN IV TWO-DIMENSIONAL DISCRETE ANGULAR SEGMENTATION TRANSPORT PROGRAM

By Clayton E. Barber

Lewis Research Center
Cleveland, Ohio
A FORTRAN IV TWO-DIMENSIONAL DISCRETE ANGULAR SEGMENTATION TRANSPORT PROGRAM

by Clayton E. Barber

Lewis Research Center

SUMMARY

The development of the discrete angular segmentation method is given for solving the transport equation for a multigroup two-dimensional FORTRAN IV program. Finite-difference equations are derived in some detail and calculational procedures are discussed. The application of scaling and overrelaxation as methods to accelerate the convergence of the neutron fluxes is described. The effect of these methods on convergence rates for selected problems is shown. The program is described, including the input instructions, along with some notes concerning program operation. A sample problem and a FORTRAN IV listing are also included.

INTRODUCTION

A transport program is required because of the inability of diffusion theory to give acceptable results for some problems. For example, the discrepancies between diffusion theory and experiment for some high leakage cores, for the streaming along void regions, and for thin highly absorptive regions makes this need apparent. The discrete angular segmentation method of transport theory has yielded smaller discrepancies than diffusion theory for problems of these types.

The discrete angular segmentation or \( S_n \) method was originally developed by Carlson at Los Alamos Scientific Laboratory (refs. 1 to 3). The \( S_n \) method is basically a numerical, iterative difference method in which the continuous angular distribution of neutron velocities is represented by considering discrete angular directions. This method was originally developed for the calculation of fast-reactor assemblies that usually converged rapidly. It has since been applied to a wide variety of problems. Some of these problems such as those with large regions of long neutron lifetime, thermal reactor systems, or complex two-dimensional problems sometimes converge very slowly.
It is thus desirable to accelerate the flux convergence as much as possible.

The TDSN program was developed in recognition of the need for a FORTRAN IV multigroup two-dimensional $S_n$ transport program that would converge these problems by the application of overrelaxation methods to the flux and source iterations. Various scaling procedures are discussed by Carlson (ref. 4), while other approaches to the problem of accelerating convergence are those of Putnam (ref. 5) and of Blue and Flatt (ref. 6).

The TDSN program will solve either one-dimensional (slab, cylinder, or sphere) or two-dimensional ($x, y$ or $r, z$) problems with either zero-return-current or reflective boundary conditions. The reflective condition for curved boundaries can be either mirror reflection from a plane surface or isotropic reflection. Certain types of geometric symmetries are used to reduce the number of mesh intervals required in two-dimensional problems. Buckling losses for transverse directions can be included. The cross sections can be either $P_1$ or transport corrected $P_0$ with full up- and down-scattering matrices. The array dimensions are completely flexible. Approximately 15,000 storages are required for the unsubscripted variables and the program, including the Lewis monitor system, so that about 17,500 storages are available for subscripted variables on a 32,768 storage computer.

Acceleration parameters may be used, and provisions are included to stop and restart a problem after a specified running time or inner iteration count. On restart, the acceleration parameters, convergence criteria, and output options can be changed.

Output options include activity tables, flux-weighted cross-section averaging, col-collapsed-group cross sections, disadvantage factors, and edits over specified groups and mesh intervals. Input options include using a flat initial flux guess, reading in an initial flux guess, or using a binary card dump of a previous problem as a flux guess. Direction cosines and weights may be included in the input or may be calculated by the program by using a moment modified quadrature procedure.

Another limitation on problems to be run by TDSN, besides size, is that the eigenvalue must be the multiplication factor. Thus, no option for running a calculation with either an external source or a distributed internal source is available.

**SYMBOLS**

- $A_{ik}$: area of mesh-cell surface at $r_i$ and $\vec{z}_k$ normal to $\vec{f}$, cm$^2$
- $C_{ik}$: area of mesh-cell surface at $\vec{r}_i$ and $\vec{z}_k$ normal to $\vec{k}$, cm$^2$
- $c$: number of neutrons emitted per collision, $(\nu \Sigma^f + \Sigma^S)/\Sigma^t$
- $f$: scale factor
G
\hat{i}, \hat{j}, \hat{k}

number of neutron lethargy groups

unit vectors for rectangular coordinates

\( J_{qg}(\vec{\rho}) \)

neutron current in q-direction; net number of neutrons in lethargy group \( g \) per unit area flowing in the q-direction at point specified by \( \vec{\rho} \), neutrons/cm\(^2\)

\( k_{\text{eff}} \)

neutron multiplication factor

\( n \)

order of \( S_n \) quadrature

\( P_\ell(\xi) \)

Legendre polynomial

\( P_{\ell m}(\hat{\Omega}) \)

spherical harmonic

\( P^m_\ell(\xi) \)

associated Legendre polynomial

\( \hat{r} \)

unit vector in radial direction

\( r, z \)

coordinate variables to specify position \( \vec{\rho} = \hat{r}r + z\hat{k} \), cm

\( S(u, \vec{\rho}, \hat{\Omega}) \)

directional source; number of neutrons per unit solid angle per unit volume introduced per unit lethargy into direction \( \hat{\Omega} \) at position \( \vec{\rho} \), neutrons/cm\(^3\)/sr/unit lethargy (see eq. (2))

\( S_g(\vec{\rho}, \hat{\Omega}) \)

total directional source for lethargy group \( g \) (see eq. (10)), neutrons/cm\(^3\)/sr

\( u \)

neutron lethargy

\( V_{ik} \)

volume of mesh cell surrounding point \( \vec{r}_i, \vec{r}_k \), cm\(^3\)

\( w_{m\ell} \)

quadrature weights for angular segment \( m, \ell \)

\( \delta \)

deviation of two results

\( \theta \)

angle between \( \hat{k} \) and \( \hat{\Omega} \)

\( \lambda \)

growth factor for production; ratio of total production resulting from outer iteration to similar production at beginning of iteration

\( \nu \)

average number of neutrons that appear as result of a nuclear fission

\( \xi \)

cosine of \( \theta \)

\( \vec{\rho} \)

position vector, cm

\( \Sigma^a(u, \vec{\rho}) \)

probability per unit distance of neutron travel of a neutron at lethargy \( u \) and position \( \vec{\rho} \) being absorbed by a nucleus, cm\(^{-1}\)

\( \Sigma^f(u, \vec{\rho}) \)

probability per unit distance of neutron travel of a nuclear fission at lethargy \( u \) and position \( \vec{\rho} \), cm\(^{-1}\)
\( \Sigma^S(u, \bar{\rho}) \) probability per unit distance of neutron travel of neutron at lethargy \( u \) and position \( \bar{\rho} \) being scattered by a nucleus, \( \text{cm}^{-1} \)

\( \Sigma^S(u', u, \bar{\rho}, \hat{\Omega}', \hat{\Omega}) \) kernal for neutron-nucleus scattering per unit distance of neutron travel from lethargy \( u' \) about lethargy \( u \) per unit lethargy through angle \( \cos^{-1}(\hat{\Omega}', \hat{\Omega}) \) at position \( \bar{\rho} \), per cm per unit lethargy

\( \Sigma^t(u, \bar{\rho}) \) total collision probability per unit distance of neutron travel at lethargy \( u \) and position \( \bar{\rho} \), \( \Sigma^t = \Sigma^a + \Sigma^s \), \( \text{cm}^{-1} \)

\( \Sigma^{tr}(u, \bar{\rho}) \) macroscopic transport cross section per unit distance of neutron travel at lethargy \( u \) and position \( \bar{\rho} \), \( \text{cm}^{-1} \)

\( \Phi_g(\bar{\rho}) \) nondirectional scalar neutron flux; number of neutrons in lethargy group \( g \) per unit area at point specified by \( \bar{\rho} \), neutrons/cm²

\( \varphi \) angle between \( \hat{r} \) and projection of \( \hat{\Omega} \) in plane perpendicular to \( \hat{k} \)

\( \chi(u)du \) fission spectrum; probability that a neutron born in fission of a nucleus will be in lethargy interval \( du \) about lethargy \( u \),

\[ \int \chi(u)du = 1 \]

\( \Psi(u, \bar{\rho}, \hat{\Omega}) \) directional flux; number of neutrons per unit lethargy per unit solid angle per unit area at point specified by position vector \( \bar{\rho} \) and flowing in direction \( \hat{\Omega} \), neutrons/cm²/sr/unit lethargy

\( \hat{\Omega} \) direction of neutron flow at position \( \bar{\rho} \)

\( \Omega_r \) direction cosine to radial direction \( \hat{r} \left( \Omega_r = \sqrt{1 - \xi^2 \cos \varphi} \right) \)

\( \Omega_x, \Omega_y, \Omega_z \) rectangular coordinate direction cosines (\( \Omega_z \) represents same cosine as \( \xi \))

\( \hat{\Omega} \cdot \nabla \Psi(u, \bar{\rho}, \hat{\Omega}) \) directional derivative; net flow of neutrons in direction \( \hat{\Omega} \) per unit lethargy per unit solid angle per unit volume away from point specified by position vector \( \bar{\rho} \), neutrons/cm³/sr/unit lethargy

\( \omega \) overrelaxation factor

Subscripts:

\( b \) boundary

\( g \) lethargy group

\( i \) numerical position of mesh cell in first direction, \( r \) or \( x \)

\( k \) numerical position of mesh cell in second direction, \( y \) or \( z \)

\( 4 \)
The time-independent Boltzmann equation of transport theory may be written as

$$\hat{\Omega} \cdot \nabla \psi(u, \bar{\rho}, \hat{\Omega}) + \Sigma^t(u, \bar{\rho}) \psi(u, \bar{\rho}, \hat{\Omega}) = S(u, \bar{\rho}, \hat{\Omega})$$

(1)
\[ S(u, \vec{\rho}, \hat{\Omega}) = \frac{\chi(u)}{4\pi k_{\text{eff}}} \frac{1}{4\pi} \int_{\Omega'} \int_{u'} \nu \Sigma_f(u', \vec{\rho}) \Psi(u', \vec{\rho}, \hat{\Omega}') du' d^2\hat{\Omega}' \]

\[ + \frac{1}{4\pi} \int_{\Omega'} \int_{u'} \Sigma^s(u' - u, \vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega}) \Psi(u', \vec{\rho}, \hat{\Omega}') du' d^2\hat{\Omega}' \]  

\[ (2) \]

\( \Psi(u, \vec{\rho}, \hat{\Omega}) \) denotes the number of neutrons per unit lethargy per unit solid angle per unit area at the point specified by the position vector \( \vec{\rho} \) and flowing in the direction \( \hat{\Omega} \); \( \Sigma^t(u, \vec{\rho}) \) is the total collision probability at lethargy \( u \) per unit length at point \( \vec{\rho} \); and \( S(u, \vec{\rho}, \hat{\Omega}) \) is the source per unit lethargy per unit solid angle per unit volume in direction \( \hat{\Omega} \) at position \( \vec{\rho} \). The equation represents a neutron conservation law stating that the total derivative in the direction \( \hat{\Omega} \) of the neutron flux at position \( \vec{\rho} \) per unit volume per unit lethargy per unit solid angle equals the number of neutrons introduced in that direction less the number which are removed by collisions. Positive solutions of the equation may be obtained for only one value of the multiplication factor \( k_{\text{eff}} \) such that the conservation law will hold. The equation is linear in the flux and of first order in the variables \( \vec{\rho} \) and \( \hat{\Omega} \).

Making the multigroup approximation to the continuous lethargy dependence of equation (1) replaces it by \( G \) equations of similar form; one for each of the lethargy intervals over which the lethargy dependent variables are replaced by suitable averages. Each equation now receives the subscript \( g \) to denote the lethargy group and the equations to be solved become

\[ \hat{\Omega} \cdot \nabla \Psi_g(\vec{\rho}, \hat{\Omega}) + \Sigma^t_g(\vec{\rho}) \Psi_g(\vec{\rho}, \hat{\Omega}) = S_g(\vec{\rho}, \hat{\Omega}) \quad g = 1, \ldots, G \]

\[ (3) \]

and

\[ S_g(\vec{\rho}, \hat{\Omega}) = \frac{x_g}{k_{\text{eff}}} \sum_{g'=1}^{G} \nu \Sigma^f_{g'}(\vec{\rho}) \left[ \frac{1}{4\pi} \int_{\Omega'} \Psi_{g'}(\vec{\rho}, \hat{\Omega}') d^2\hat{\Omega}' \right] \]

\[ + \sum_{g'=1}^{G} \frac{1}{4\pi} \int_{\Omega'} \Sigma^{s}_{g' \rightarrow g}(\vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega}) \Psi_{g'}(\vec{\rho}, \hat{\Omega}') d^2\hat{\Omega}' \]

\[ (4) \]

where \( \Sigma^{s}_{g' \rightarrow g}(\vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega}) \) is the kernel for scattering from the group \( g' \) to group \( g \) through the angle \( \cos^{-1}(\hat{\Omega}' \cdot \hat{\Omega}) \).

For two-dimensional \( r, z \) cylindrical symmetry, \( \Psi \) depends on two spatial
coordinates, r and z, and two directional coordinates, ξ and φ, and is an even function of both ξ and φ. In sketch (a), r is the perpendicular distance from the axis of symmetry, z is the distance parallel to the axis from the reference plane, ξ is the cosine of θ, which is the angle between the unit vectors \( \hat{k} \) and \( \hat{\Omega} \), and φ is the angle in the plane perpendicular to the axis between the unit vector \( \hat{r} \) and the projection of \( \hat{\Omega} \). References 4 and 7 discuss this geometry as well as other geometries.

The directional derivative \( \hat{\Omega} \cdot \nabla \psi (\rho, \hat{\Omega}) \) in the transport equation (eq. 3) will now be derived. Note that \( \hat{\Omega} \cdot \nabla \psi = d\psi / ds \), where ds is a distance laid off along \( \hat{\Omega} \) and where

\[
\frac{d\psi}{ds} = \frac{\partial \psi}{\partial r} \frac{dr}{ds} + \frac{\partial \psi}{\partial \phi} \frac{d\phi}{ds} + \frac{\partial \psi}{\partial z} \frac{dz}{ds}
\]

From sketch (a); \( dr = (\sin \theta \ ds) \cos \phi \) or

\[
\frac{dr}{ds} = \sin \theta \ \cos \phi = \sqrt{1 - \xi^2} \ \cos \phi
\]

Also,

\[
-r \ d\phi = (\sin \theta \ ds) \sin \phi
\]

or

\[
\frac{d\phi}{ds} = -\frac{\sin \theta \ \sin \phi}{r} = -\frac{\sqrt{1 - \xi^2}}{r} \ \sin \phi
\]

and
\[ dz = \cos \theta \ ds \]

or

\[ \frac{dz}{ds} = \cos \theta = \xi \]

Also note that \( dr/ds \) corresponds to \( \Omega_r \), and \( dz/ds \) corresponds to \( \Omega_z \).

Making these substitutions yields

\[
\frac{d\psi}{ds} = \sqrt{1 - \xi^2} \cos \varphi \frac{\partial \psi}{\partial r} - \frac{\sqrt{1 - \xi^2}}{r} \sin \varphi \frac{\partial \psi}{\partial \varphi} + \xi \frac{\partial \psi}{\partial z}
\]

and equation (3) becomes

\[
\sqrt{1 - \xi^2} \cos \varphi \frac{\partial \psi_g}{\partial r} - \frac{\sqrt{1 - \xi^2}}{r} \sin \varphi \frac{\partial \psi_g}{\partial \varphi} + \xi \frac{\partial \psi_g}{\partial z} + \Sigma g \frac{\partial \psi}{\partial g} = S_g \tag{5}
\]

with

\[
\psi_g = \psi_g(r, z, \xi, \varphi)
\]

The first three terms of equation (5) are not in the usual form for expressing the divergence in cylindrical coordinates. This may be remedied by noting that

\[
\frac{\partial}{\partial \varphi} \left[ \sin \varphi \psi \right] = (\cos \varphi)\psi + \sin \varphi \frac{\partial \psi}{\partial \varphi}
\]

and

\[
\frac{\partial (r\psi)}{\partial r} = \psi + r \frac{\partial \psi}{\partial r}
\]

Using these relations gives

\[
\sin \varphi \frac{\partial \psi}{\partial \varphi} = (\cos \varphi)\psi + \frac{\partial}{\partial \varphi} \left[ \sin \varphi \psi \right] - \cos \varphi \frac{\partial (r\psi)}{\partial r}
\]

and equation (5) becomes
which is a suitable conservative form of the transport equation in cylindrical coordinates.

Two basic boundary conditions can be placed on equation (6): reflective and nonreflective. The axis of the cylinder necessarily has the reflective condition applied. At the outer radius \( r_b \) of the system, the nonreflective or zero-return-current boundary condition is

\[
\Psi(r_b, z, \xi, \varphi) = 0 \quad 0 \leq \varphi \leq \frac{\pi}{2}
\]

where \( \varphi \) is the angle in the plane perpendicular to the cylindrical axis between the unit vector \( \hat{r} \) and the projection of \( \hat{n} \).

At the upper or lower boundary \( z_b, \)

\[
\Psi(r, z_b, \xi, \varphi) = 0
\]

with \(-1 \leq \xi \leq 0\) at the upper boundary and \(0 \leq \xi \leq 1\) at the lower boundary, where \( \xi \) is the cosine of the angle between the unit vectors \( \hat{k} \) and \( \hat{n} \).

For planar reflection, the radial boundary condition is expressed as \( \Psi(r_b, z, \xi, \varphi) = \Psi(r_b, z, \xi, \pi - \varphi) \) and at the axial boundaries as \( \Psi(r, z_b, \xi, \varphi) = \Psi(r, z_b, -\xi, \varphi) \). The planar reflective boundary condition for a curved boundary lacks physical meaning because this condition leaves, for example, \( \Psi(r_b, z, \xi, \pi/2) = 0 \) for \( r, z \) geometry. Another reflective boundary condition (ref. 5) is an isotropic return condition defined by

\[
\Psi(r_b, z, \xi, \varphi) = \frac{\int_{0}^{\pi/2} \int_{-1}^{1} \sqrt{1 - \xi'^{2}} \cos \varphi' \Psi'(r_b, z, \xi', \varphi') \, d\xi' \, d\varphi'}{\int_{0}^{\pi/2} \int_{-1}^{1} \sqrt{1 - \xi'^{2}} \, d\xi' \, d\varphi'}\quad \frac{\pi}{2} \leq \varphi \leq \pi
\]

Mirror and 180° rotational symmetry conditions may be applied to \( x, y \) configurations in addition to the boundary conditions expressed previously for \( r, z \) geometry. Mirror symmetry may also be applied to the \( z \)-direction of \( r, z \) configurations. Mirror symmetry results in conditions expressed in the same manner as the planar reflective boundary conditions for the appropriate values of \( x, y, \) or \( z \) and for the correct range of \( \varphi \) and \( \xi \).
The 180° rotational symmetry condition may be applied if rotating that portion of a system below one of its diagonals through 180° about its midpoint results in the same configuration as that above the diagonal. This condition is defined by

\[ \Psi(x_b - x, 0, \xi, \varphi) = \Psi(x, 0, -\xi, \pi - \varphi) \]

where the origin is at the lower left corner of the configuration and its right boundary is at \( x_b \).

**SOURCE FUNCTION**

The scattering source (the second term of eq. (4)) may be represented in terms of integrals involving two independent functions of the direction \( \hat{\Omega} \). Expansions in spherical harmonics are employed to permit these integrals to be evaluated approximately. Only the zero and first harmonics of the expansion of \( P_{\ell m}(\hat{\Omega}) \) will be represented. \( P_{\ell}(\xi) \) represents the Legendre polynomials, and \( P_{\ell}^{m}(\xi) \) the associated Legendre polynomials.

The following expressions (from ref. 8)

\[ P_{\ell, 0}(\hat{\Omega}) = P_{\ell}(\xi) = \frac{1}{2^\ell \ell!} \frac{d^\ell}{d\xi^\ell} (\xi^2 - 1)^\ell \quad \ell \geq 0 \]

\[ P_{\ell, m}(\hat{\Omega}) = \sqrt{\frac{2(\ell - m)!}{(\ell + m)!}} P_{\ell}^{m}(\xi) \cos(m\varphi) \quad \ell \geq 1; \ 0 < m \leq \ell \]

\[ P_{\ell, -m}(\hat{\Omega}) = \sqrt{\frac{2(\ell - m)!}{(\ell + m)!}} P_{\ell}^{m}(\xi) \sin(m\varphi) \quad \ell \geq 1; \ 0 < m \leq \ell \]

where

\[ P_{\ell}^{m}(\xi) = \frac{(1 - \xi^2)^{m/2}}{2^\ell \ell!} \frac{d^{\ell+m}}{d\xi^{\ell+m}} (\xi^2 - 1)^\ell \]

can be used to obtain the spherical harmonics

\[ P_{0, 0}(\hat{\Omega}) = P_{0}(\xi) = 1 \]
\[
\begin{align*}
P_{1,0}(\hat{\Omega}) &= P_1(\xi) = \xi \quad \equiv \Omega_z \\
P_{1,1}(\hat{\Omega}) &= \sqrt{1 - \xi^2} \cos \varphi \quad \equiv \Omega_x \\
P_{1,-1}(\hat{\Omega}) &= \sqrt{1 - \xi^2} \sin \varphi \quad \equiv \Omega_y \\
\end{align*}
\]

where

\[
\hat{\Omega} = \sqrt{1 - \xi^2} (\cos \varphi)\hat{\Omega} + \sqrt{1 - \xi^2} (\sin \varphi)\hat{\Omega} + \xi\hat{k}
\]

The harmonics have the orthogonality property

\[
\frac{2\ell + 1}{4\pi} \int P_{\ell',m'}(\hat{\Omega}) P_{\ell,m}(\hat{\Omega}) d^2\hat{\Omega} = \delta_{\ell',\ell} \delta_{m',m}
\]

and the addition formula

\[
P_\ell(\hat{\Omega}' \cdot \hat{\Omega}) = \sum_{m=-\ell}^{\ell} P_{\ell m}(\hat{\Omega}') P_{\ell m}(\hat{\Omega})
\]

The elastic scattering cross section may be expanded as

\[
\Sigma_{g' - g}^{S} = \sum_{L} (2\ell + 1) \Sigma_{g' - g}^{S}\ell\ell(\hat{\Omega}) P_{\ell}(\hat{\Omega}' \cdot \hat{\Omega})
\]

where

\[
\Sigma_{g' - g}^{S}\ell\ell(\hat{\Omega}) = \frac{1}{4\pi} \int \Sigma_{g' - g}^{S} P_{\ell}(\hat{\Omega}' \cdot \hat{\Omega}) d^2\hat{\Omega}
\]

and where \(\Sigma_{g' - g}^{S}(\hat{\Omega}, \hat{\Omega}' \cdot \hat{\Omega})\) is the kernel for elastic scattering from group \(g'\) to group \(g\) through the angle \(\cos^{-1}(\hat{\Omega}' \cdot \hat{\Omega})\) and the flux may be expanded as
\[
\Psi_g(\vec{\rho}, \hat{\Omega}) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} (2\ell + 1)f_{g\ell m}(\vec{\rho})P_{\ell m}(\hat{\Omega})
\]

where

\[
f_{g\ell m}(\vec{\rho}) = \frac{1}{4\pi} \int \Psi_g(\vec{\rho}, \hat{\Omega})P_{\ell m}(\hat{\Omega})d^2\hat{\Omega}
\]

and where \( L \) is equal to infinity. However, for this study, \( L \) will be either 0 or 1.

With these expansions, the elastic scattering source is given by

\[
S_{g}^{L}(\vec{\rho}, \hat{\Omega}) = \sum_{g' = 1}^{G} \left( \frac{1}{4\pi} \right) \int \Sigma_{g' \rightarrow g}^{S}(\vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega})\Psi_{g'}(\vec{\rho}, \hat{\Omega}')d\hat{\Omega}'
\]

\[
= \sum_{g' = 1}^{G} \sum_{\ell=0}^{L} (2\ell + 1)\Sigma_{g' \rightarrow g}^{S\ell}(\vec{\rho}) \sum_{m=-\ell}^{\ell} f_{g'\ell m}(\vec{\rho})P_{\ell m}(\hat{\Omega})
\]

(7)

where \( L = 0 \) for \( P_0 \) scattering and \( L = 1 \) for \( P_1 \) scattering.

If cross sections for inelastic and \( n-2n \) scattering are available, they may be included by adding them to \( \Sigma_{g' \rightarrow g}^{S0} \) to obtain a total scattering matrix defined by

\[
\Sigma^{st}_{g' \rightarrow g}(\vec{\rho}) = \Sigma^{S0}_{g' \rightarrow g}(\vec{\rho}) + \Sigma^{in}_{g' \rightarrow g}(\vec{\rho}) + 2\Sigma^{n-2n}_{g' \rightarrow g}(\vec{\rho})
\]

The group scattering cross section \( \Sigma_{g}(\vec{\rho}) \) becomes

\[
\Sigma_{g}(\vec{\rho}) = \sum_{g' = 1}^{G} \Sigma^{S0}_{g \rightarrow g'}(\vec{\rho}) + \Sigma^{in}_{g \rightarrow g'}(\vec{\rho}) + \Sigma^{n-2n}_{g \rightarrow g'}(\vec{\rho})
\]

and the group scattering removal \( \Sigma^{r}_{g}(\vec{\rho}) \) is similar to \( \Sigma_{g}(\vec{\rho}) \), where the sum does not include \( g' \) equal to \( g \).

For cylindrical geometry, \( \Psi_g(\vec{\rho}, \hat{\Omega}) \) is even in \( \phi \). The only \( P_{\ell m}(\hat{\Omega}) \) that is
odd in \( \varphi \), yielding \( f_{g\ell m} = 0 \), is \( P_{1,-1}(\hat{\Omega}) \). Using the definition of \( f_{g\ell m} \) gives

\[
f_{g, 0, 0(\vec{\rho})} = \frac{1}{4\pi} \int \Psi_g(\vec{\rho}, \hat{\Omega}) P_{0, 0}(\hat{\Omega}) d^2\hat{\Omega}
\]
or, since

\[
P_{0, 0}(\hat{\Omega}) = 1
\]

\[
f_{g, 0, 0(\vec{\rho})} = \frac{1}{4\pi} \int \Psi_g(\vec{\rho}, \hat{\Omega}) d^2\hat{\Omega}
\]

This expression is just the integral of the directional flux over all directions, or the scalar flux. Therefore, make the identification

\[
\Phi_g(\vec{\rho}) \equiv f_{g, 0, 0(\vec{\rho})}
\]

For \( f_{g, 1, 0} \) the definition of \( f_{g\ell m} \) results in

\[
f_{g, 1, 0(\vec{\rho})} = \frac{1}{4\pi} \int \Psi_g(\vec{\rho}, \hat{\Omega}) P_{1, 0}(\hat{\Omega}) d^2\hat{\Omega}
\]
or

\[
f_{g, 1, 0(\vec{\rho})} = \frac{1}{4\pi} \int \Psi_g(\vec{\rho}, \hat{\Omega}) \xi d^2\hat{\Omega}
\]

where \( \xi = \Omega_z \) for cylindrical geometry. Therefore,

\[
J_{g_\rho}(\vec{\rho}) \equiv f_{g, 1, 0(\vec{\rho})}
\]

and similarly for \( f_{g, 1, 1} \), the identification

\[
J_{g_r}(\vec{\rho}) \equiv f_{g, 1, 1(\vec{\rho})}
\]

results.

Making these substitutions into equation (7) results in the scattering source, for \( L = 0 \),

\[
S^0_g(\vec{\rho}, \hat{\Omega}) = \sum_{g' = 1}^{G} \Sigma_{g'}^{st}(\vec{\rho}) \Phi_{g'}(\vec{\rho})
\]  

(8)
and, for \( L = 1 \)

\[
S^1_g(\vec{r}, \Omega) = \sum_{g'=1}^{G} \left[ \sum_{g'=g}^{G} (\rho) \Phi_{g'}(\vec{r}) + 3 \Sigma_{g'-g}^{g'}(\rho) \left[ \xi J_{g'}(\vec{r}) + \sqrt{1 - \xi^2} (\cos \phi) J_{g'}(\vec{r}) \right] \right]
\]

and equation (4) becomes

\[
S_g(\vec{r}, \Omega) = \frac{\chi_g}{k_{\text{eff}}} \sum_{g'=1}^{G} \nu \Sigma^f_{g'-g}(\rho) \Phi_{g'}(\vec{r}) + S^f_g(\vec{r}, \Omega)
\]

Equation (9) may be generalized for other geometries by recognizing that \( \xi = \Omega \),

and that \( \sqrt{1 - \xi^2} \cos \phi = \Omega \). Then equation (9) becomes

\[
S^1_g(\vec{r}, \Omega) = \sum_{g'=1}^{G} \left[ \sum_{g'=g}^{G} (\rho) \Phi_{g'}(\vec{r}) + 3 \Sigma_{g'-g}^{g'}(\rho) \sum_q \Omega_q J_{g'}(\vec{r}) \right]
\]

for each direction \( q \) of the coordinate system.

The source \( S_g(\vec{r}, \Omega) \) may also be written in a form that separates contributions to the source from within the group and from other groups. To do this, let

\[
\begin{align*}
F(\vec{r}) &= \sum_{g'=1}^{G} \nu \Sigma^f_{g'}(\rho) \Phi_{g'}(\vec{r}) \\
F^t(\vec{r}) &= \sum_{g'=1}^{G} \Sigma_{g'-g}^{g'}(\rho) \Phi_{g'}(\vec{r}) \\
F^1(\vec{r}, \Omega) &= 3 \sum_{g'=1}^{G} \Sigma_{g'-g}^{g'}(\rho) \sum_q \Omega_q J_{g'}(\vec{r})
\end{align*}
\]

(12)
Then

\[
S_g(\vec{\rho}, \hat{\Omega}) = \frac{\chi_g}{k_{\text{eff}}} F(\vec{\rho}) + T^t_g(\vec{\rho}) + T^1_g(\vec{\rho}, \hat{\Omega}) + \Sigma_{g-g}(\vec{\rho}) \Phi_g(\vec{\rho}) + 3\Sigma_{g-g}(\vec{\rho}) \sum_q \Omega_q J_q(\vec{\rho})
\]  

(13)

In many problems, anisotropic scattering is adequately taken into account by either the diagonal or the full transport approximation. For either of these approximations, anisotropic scattering is accounted for by an appropriate modification of the cross sections, and on the use of these modified values, the scattering is treated as if it were isotropic:

\[
S_g(\vec{\rho}) = \frac{\chi_g}{k_{\text{eff}}} F(\vec{\rho}) + T^t_g(\vec{\rho}) + \Sigma_{g-g}(\vec{\rho}) \Phi_g(\vec{\rho})
\]  

(14)

The transport approximation may be regarded as replacing an anisotropic scattering distribution with an isotropic component and a \( \delta \)-function forward scattering component. In the diagonal transport approximation (ref. 9), \( \Sigma^t_g \) is replaced by \( \left( \Sigma^t_g - \Sigma^{s1}_g \right) \), and \( \Sigma_{g-g}^{st} \) is replaced by \( \left( \Sigma_{g-g}^{st} - \Sigma^{s1}_g \right) \), where

\[
\Sigma^{s1}_g = \sum_{g'=1}^{G} \Sigma^{s1}_{g-g'}
\]

and \( \Sigma^{tr}_g \) is defined to be \( \Sigma^t_g - \Sigma^{s1}_g \).

A second method (ref. 2) of developing the transport approximation is to use \( \Sigma^{tr}_g \) as defined by a cross-section program to replace \( \Sigma^t_g \). Then \( \Sigma_{g-g}^{st} \) is replaced by \( \Sigma^{tr}_g - \Sigma^{a}_g - \Sigma^r_g \), where

\[
\Sigma^r_g = \sum_{g'=1}^{G} \left( \Sigma^{s0}_{g-g'} + \Sigma^{in}_{g-g'} + \Sigma^{n-2n}_{g-g'} \right)
\]

\[g' \neq g\]

where \( \Sigma^a_g \) is the cross section for neutron absorption including absorptions that result in fission. Under these circumstances, \( \Sigma^{st}_{g-g} \) may become negative.
For an adjoint problem (refs. 7 and 10), the source is developed differently resulting in

\[
S_{g}^{*}(\rho, \hat{\Omega}) = \nu \Sigma_{g}^{f} F_{g}^{*}(\rho) + T_{g}^{t*}(\rho) + T_{g}^{1*}(\rho, \hat{\Omega}) + \Sigma_{g \rightarrow g}^{st}(\rho) \Phi_{g}^{*}(\rho) + 3 \Sigma_{g \rightarrow g}^{s1}(\rho) \sum_{q} \Omega_{q} J_{qg}^{*}(\rho)
\]

(15)

where

\[
F_{g}^{*}(\rho) = \sum_{g' = 1}^{G} \frac{\chi_{g'}}{k_{\text{eff}}} \Phi_{g'}^{*}(\rho)
\]

(16a)

\[
T_{g}^{t*}(\rho) = \sum_{g' = 1}^{G} \Sigma_{g \rightarrow g'}^{st}(\rho) \Phi_{g'}^{*}(\rho)
\]

(16b)

\[
T_{g}^{1*}(\rho, \hat{\Omega}) = 3 \sum_{g' = 1}^{G} \Sigma_{g \rightarrow g'}^{s1}(\rho) \sum_{q} \Omega_{q} J_{qg}^{*}(\rho)
\]

(16c)

SPATIAL MESH AND DIRECTIONAL SEGMENTATION

The transport equation, as given for cylindrical symmetry by equation (6), can be solved by performing integrations over the direction variable \( \hat{\Omega} \) such as those involved in the source function (eq. (4)). In obtaining numerical solutions, these integrations may be performed by means of mechanical quadrature in which the continuous variable \( \hat{\Omega} \) is represented by a set of discrete directions. The conversion of the transport equation into difference form also requires that the space coordinates represented by the continuous variable \( \rho \) be divided into a set of intervals.

For the space dimensions, the division is arbitrary, and for the \( r, z \) cylindrical geometry, a set of \( r_{i} \) is used to divide \( r \) into \( I \) intervals and a set of \( z_{k} \) to divide \( z \).
into K intervals. An i,k mesh results over the volume of the cylinder. In general, the mesh intervals are selected so that the surfaces of the resulting mesh cells coincide as much as possible with the natural bounding surfaces of the configuration being represented. All the mesh cells are considered to be uniform in material and free of any internal boundaries. The radial and axial bounding areas of the mesh cells are given by

\[ A_{ik} = 2\pi r_i \Delta z_k \quad \text{and} \quad C_{ik} = \pi \left(r_{i+1}^2 - r_i^2\right) \]

and the volume of the mesh cell by

\[ V_{ik} = \pi \left(r_{i+1}^2 - r_i^2\right) \Delta z_k. \]

For the direction variables \( \xi \) and \( \varphi \), a two-dimensional partition of the surface of a unit sphere centered at each mesh cell is required. Since \( \psi \) and \( S \) are both even functions of \( \varphi \), it is only necessary to consider the range of \( \varphi \) from zero to \( \pi \). The directional variable \( \xi \) is also even and, if no \( z \) variation exists, \( \xi \) can be considered only for the range from zero to 1 and only the first quadrant of the unit sphere needs to be used. Even with \( z \) variation, the section of the unit sphere with \( \xi \) between -1 and zero will have the same form of mechanical quadrature as the first quadrant, so that only the first quadrant needs to be considered in devising the quadrature scheme. In the past, a system of Gaussian quadrature (refs. 4 and 5) or a moment modified quadrature system (ref. 3) has been used. More recently projection-invariant quadrature sets have been proposed (refs. 4 and 7).

Whatever quadrature system is to be used, certain relations should be required. The most important of these are

\[ \sum_{m, \ell} w_{m\ell} = 1 \]  \hspace{1cm} (17a)

\[ \sum_{m, \ell} w_{m\ell} \Omega_{m\ell} = 0 \]  \hspace{1cm} (17b)

\[ \sum_{m, \ell} w_{m\ell} \Omega_{m\ell}^2 = \frac{1}{3} \]  \hspace{1cm} (17c)

In some quadrature sets (i.e., the Double-P method and the projection-invariant Set B, ref. 4), the first moment relation

\[ \sum_{m, \ell} w_{m\ell} |\Omega_{m\ell}| = \frac{1}{2} \]  \hspace{1cm} (17d)
also holds. Except for equation (17b), these are all forms of a general condition

\[ \sum_{m, \ell} w_{m\ell} | \Omega_{m\ell}^{q} | = \frac{1}{q + 1} \]

where \( q \) is the order of the moment.

In the moment modified sets, the first quadrant of the unit sphere (and others if required) is divided into \( n/2 \) bands of latitude by arcs of latitude of constant cosine \( \xi \). These arcs are defined by dividing the continuous variable \( \xi \) into a set of \( \xi_m \). These bands are determined so that the areas of the bands, as \( \xi \) decreases, are \( na, (n - 2)a, \ldots, 4a, 2a \). The sum of these areas must be \( \pi \), the area of the first quadrant, or

\[ \sum_{m=1}^{n/2} 2ma = \pi \]

where

\[ \sum_{m=1}^{n/2} 2ma = \frac{n}{2} \left( \frac{n + 2}{2} \right) a \]

Thus, \( a = 4\pi/n(n + 2) \). Each band is now divided by arcs of longitude (\( \phi \) constant) into sections of area \( a \) with the bands partitioned into \( L = n, n - 2, \ldots, 4, 2 \) sections. Here, the continuous variable \( \phi \) is divided into a set of boundaries \( \phi_{m\ell} \) so that

\( \phi_{m,0} = \pi, \phi_{m,L/2} = \frac{\pi}{2}, \) and \( \phi_{m,L} = 0 \)

as shown in sketch (b) for \( n = 6 \).

With each area is associated an average value \( \bar{\xi}_m \) of \( \xi \) and \( \bar{\phi}_{m\ell} \) of \( \phi \). The average direction \((\bar{\xi}_m, \bar{\phi}_{m\ell})\) is chosen so that condition (17c) holds (ref. 3). In the quadrature system being considered, this direction is obtained by defining \( \bar{\phi}_{m\ell} \) to be the simple average.
and with equal weights $w_{m, \ell}$ (representing the areas), while $\bar{\xi}_m$ can be defined by

$$\bar{\xi}_m = \frac{1}{2} (\xi_m + \xi_{m-1}) - \frac{\sqrt{n^2 + 2n}}{\sqrt{n^2 + 2n - 2}} \quad m = 1, 2, \ldots, \frac{n}{2}$$

and

$$\bar{\xi}_m = -\bar{\xi}_{n-m+1} \quad m = \frac{n}{2} + 1, \ldots, n$$

where

$$\xi_m = 1 - \frac{4m(m + 1)}{n(n + 2)} \quad m = 0, \ldots, \frac{n}{2}$$

and

$$\xi_m = -\xi_{n-m} \quad m = \frac{n}{2} + 1, \ldots, n$$

Projection-invariant quadratures attempt to perform the quadrature subject to symmetry requirements on the discrete directions dividing the surface of the unit sphere. The symmetry requirements are rotational invariance about the midpoint of an octant of the unit sphere and reflection invariance about the spherical arcs connecting the midpoint of the octant and its vertices. With these requirements satisfied, the results of integrations over $\hat{\Omega}$ no longer depend on the orientation of the unit sphere with respect to the space axes as they do for the moment modified quadrature.

If the areas to be associated with the discrete directions to obtain the weighting factors $w_{m, \ell}$ are laid out on the unit sphere and are made subject to the invariance requirements, a diagram for an octant such as sketch (c) for $n = 6$ results.
This figure is representative of the Set A quadrature of reference 4. The Set B quadrature of reference 4 is obtained from the Set A quadrature for \( n + 2 \) by removing the corner directions. After \( n = 4 \), the total number of directions for Set B will be more than that for the corresponding Set A quadrature. Sketch (d) represents an octant for \( n = 6 \) of Set B quadrature. (For a further discussion of projection-invariant methods see refs. 4 and 7.) Appendix B lists these types of quadrature sets for a number of \( S_n \) orders.

**TRANSPORT DIFFERENCE EQUATION FOR \( r, z \) GEOMETRY**

Equation (6) can be put into difference form, as is done in reference 7, or another neutron conservation equation for the mesh cell \( i, k \) and quadrature segment \( m, \ell \) can be developed (refs. 4 and 10). The first term of equation (6) represents the loss of neutrons per unit volume in the \( r \)-direction from a differential spatial-directional element. The second term represents a similar loss in the \( \phi \)-direction and the third represents a loss in the \( z \)-direction. Thus, the first term, following the procedure of references 4 and 10, may be written in difference form as

\[
\frac{w_{m, \ell} \Omega_{r_{m, \ell}}}{V_{i,k}} \left( A_{i+1,k} \psi_{g,i+1,k,m,\ell} - A_{i,k} \psi_{g,i,k,m,\ell} \right)
\]

which represents the net flow per unit volume out of the mesh cell \( i, k \) through the radial surfaces. In this term, \( w_{m, \ell} \) is the solid angle of the quadrature segment about the discrete direction \( m, \ell \), \( \Omega_{r_{m, \ell}} \) is the direction cosine of the angle between direction \( m, \ell \) and the radial direction yielding the component of the directional flux normal to the radial surfaces, \( A_{i+1,k} \) and \( A_{i,k} \) are the surface areas at \((r_{i+1,k}, \bar{z}_k)\) and \((r_i, \bar{z}_k)\), and \( \psi_{g,i+1,k,m,\ell} \) and \( \psi_{g,i,k,m,\ell} \) represent the average directional flux over these surfaces.

The third term is similar for the \( z \)-direction and becomes

\[
\frac{w_{m, \ell} \Omega_{z_{m, \ell}}}{V_{i,k}} \left( C_{i,k+1} \psi_{g,i,k+1,m,\ell} - C_{i,k} \psi_{g,i,k,m,\ell} \right)
\]
representing the net flow per unit volume out of the mesh cell through the axial surfaces.

The second term of equation (6) represents the angular redistribution of neutrons since it corresponds to the loss per unit volume in the \( \varphi \)-direction from the quadrature segment but not a loss from the mesh cell. In difference form this term may be written as

\[
\frac{\alpha_m \ell \psi_{g,i,k,m,\ell} - \alpha_m \ell-1 \psi_{g,i,k,m,\ell-1}}{V_{ik}}
\]

where \( \psi_{g,i,k,m,\ell} \) and \( \psi_{g,i,k,m,\ell-1} \) represent average fluxes for the quadrature segment boundaries of direction \( m, \ell \). The quantities \( \alpha_m \ell \) and \( \alpha_m \ell-1 \) are to be determined so that the net angular redistribution in the \( \varphi \)-direction for any \( m \) will be zero.

Making these substitutions into equation (6) and multiplying through by \( V_{ik} \) result in the transport difference equation:

\[
w_{m \ell} \Omega_{m \ell} (A_{i+1,k} \psi_{g,i+1,k,m,\ell} - A_{ik} \psi_{g,i,k,m,\ell}) + \alpha_m \ell \psi_{g,i,k,m,\ell} - \alpha_m \ell-1 \psi_{g,i,k,m,\ell-1} + w_{m \ell} \beta_{m \ell} (C_{i,k+1} \psi_{g,i,k+1,m,\ell} - C_{ik})
\]

\[
= w_{m \ell} S_{g,i,k,m,\ell} V_{ik} \tag{18}
\]

The \( \alpha_{m \ell} \) coefficients must now be evaluated. Because of the cylindrical symmetry, the range of \( \varphi \) to be considered is from zero to \( \pi \) and there cannot be angular redistribution to values of \( \varphi \) outside this range. Therefore, the first and last values of \( \alpha_{m \ell} \) must be zero, imposing the conditions

\[
\alpha_{m,0} = \alpha_{m,L} = 0
\]

Another condition can be determined by examining a configuration that results in constant directional flux. For such a configuration, the net flow will be zero, or from equation (18),

\[
w_{m \ell} \Omega_{m \ell} (A_{i+1,k} - A_{ik}) + (\alpha_{m \ell} - \alpha_{m,\ell-1}) + w_{m \ell} \beta_{m \ell} (C_{i,k+1} - C_{ik}) = 0
\]

For cylindrical geometry \( C_{i,k+1} = C_{ik} \) and the third term is zero. This agrees with equation (6) since it indicates no angular redistribution in the \( \xi \)-direction.

The conditions on \( \alpha_{m \ell} \) thus reduce to
\[ \alpha_m, 0 = \alpha_m, L = 0 \]
\[ \alpha_m, \ell - \alpha_m, \ell - 1 = -w_m, \ell \Omega r_m, \ell (A_{i+1}, k - A_{i k}) \]

Another expression for \( \alpha_m, \ell \) can be found by noting that with

\[ \alpha_m, 0 = 0 \]

and

\[ \alpha_m, 1 = -w_m, 1 \Omega r_m, 1 (A_{i+1}, k - A_{i k}) \]
\[ \alpha_m, 2 = -\left(w_m, 2 \Omega r_m, 2 + w_m, 1 \Omega r_m, 1\right) (A_{i+1}, k - A_{i k}) \]

Substituting \( \alpha_m, 2 \) and \( \alpha_m, 1 \) into (19) gives

\[ \alpha_m, 3 = -\left(w_m, 3 \Omega r_m, 3 + w_m, 2 \Omega r_m, 2 + w_m, 1 \Omega r_m, 1\right) (A_{i+1}, k - A_{i k}) \]

or, in general,

\[ \alpha_m, \ell = -\left(w_m, \ell \Omega r_m, \ell + w_m, \ell-1 \Omega r_m, \ell-1 + \cdots + w_m, 1 \Omega r_m, 1\right) (A_{i+1}, k - A_{i k}) \]  

(20)

The equations represented by (18) (one equation for each mesh cell \( i, k \) and discrete direction \( m, \ell \) and group \( g \)) do not couple different values of \( \Omega r_m, \ell \), while the energy groups are coupled only through the source term. This system of equations can be solved by putting equation (18) into the form of a recursion relation, and the fluxes are calculated in turn. A complete set of fluxes can be obtained by starting the procedure at each geometric boundary and applying the recursion relation until the opposite boundary is reached. In carrying out this procedure, only those fluxes with directional components in the direction in which the recursion relation is being applied are obtained. The procedure must be initialized with respect to direction in addition to the initialization at the boundaries. For this, the direction \( \varphi_m, 0 = \pi \) is used for which \( \alpha_m, 0 = 0 \) and results in the procedure being started at the outer radial boundary.

This procedure can be applied by putting the recursion relation into a form in which one unknown value of the flux can be determined from a set of previously calculated values. Thus, the number of unknowns in equation (18) must be reduced by three. One of two different methods is usually used. One method is known as the 'step model'.
(or step-function method) and the other as the "diamond model" (or line-segment method).

In the step model, the average flux \( \psi_{g, i, k, m, l} \) for the mesh cell and quadrature segment is assumed to apply also at the mesh and quadrature boundaries yet to be reached in the recursion sweep. Thus, for a sweep from \( i \) to \( i+1 \) and \( k \) to \( k+1 \) the condition becomes

\[
\psi_{g, i, k, m, l} = \psi_{g, i+1, k, m, l} = \psi_{g, i, k+1, m, l} = \psi_{g, i, k, m+1, l}
\]

For the diamond model, the average flux is assumed to be the average of the opposite boundary values or

\[
\psi_{g, i, k, m, l} = \frac{1}{2} (\psi_{g, i+1, k, m, l} + \psi_{g, i, k+1, m, l})
\]

\[
= \frac{1}{2} (\psi_{g, i, k+1, m, l} + \psi_{g, i, k, m+1, l})
\]

\[
= \frac{1}{2} (\psi_{g, i, k, m, l} + \psi_{g, i, k, m, l-1}) \quad (21)
\]

The diamond model is used in TDSN since it generally leads to accurate results.

Using the diamond model for a recursion sweep from \( i \) to \( i+1 \) and from \( k \) to \( k+1 \) equation (18) becomes

\[
w_{m \ell} \Omega_{r m \ell} \left[ A_{i+1, k} (2 \psi_{g, i, k, m, l} - \psi_{g, i, k, m, l-1}) - \alpha_{m \ell} (\psi_{g, i, k, m, l} + \psi_{g, i, k, m, l-1}) \right]
\]

\[
+ w_{m \ell} \Omega_{Z m \ell} \left[ C_{i, k+1} (2 \psi_{g, i, k, m, l} - \psi_{g, i, k, m, l-1}) - C_{i k} \psi_{g, i, k, m, l} \right]
\]

or

\[
2w_{m \ell} \Omega_{r m \ell} A_{i+1, k} \psi_{g, i, k, m, l} - w_{m \ell} \Omega_{r m \ell} (A_{i+1, k} + A_{i k}) \psi_{g, i, k, m, l}
\]

\[
+ 2 \alpha_{m \ell} (\psi_{g, i, k, m, l} + \psi_{g, i, k, m, l-1})
\]

\[
+ w_{m \ell} \Omega_{Z m \ell} (C_{i, k+1} + C_{i k}) \psi_{g, i, k, m, l} - w_{m \ell} \Omega_{Z m \ell} (C_{i, k+1} + C_{i k}) \psi_{g, i, k, m, l}
\]

\[
+ w_{m \ell} \Sigma^t_{g, i, k} V_{ik} \psi_{g, i, k, m, l} = w_{m \ell} S_{g, i, k, m, l} - V_{ik}
\]

or
since $C_{i,k+1} = C_{ik}$.

A symmetrical relation for the first four terms similar to that of the fifth and sixth terms can be obtained when equation (19) is used in the form

$$w_{m\ell} \Omega_{r_{m\ell}} A_{i+1,k} = \alpha_{m, \ell-1} - \alpha_{m\ell} + w_{m\ell} \Omega_{r_{m\ell}} A_{ik}$$

to substitute for $w_{m\ell} \Omega_{r_{m\ell}} A_{i+1,k}$ in the first term. Further symmetry can be obtained by noting that

$$A_{i+1,k} + A_{ik} = 2\pi (r_{i+1} + r_i) \Delta z_k$$

$$C_{i,k+1} + C_{ik} = 2\pi (r_{i+1} + r_i) \Delta r_i$$

and

$$A_{i+1,k} - A_{ik} = 2\pi \Delta r_i \Delta z_k$$

for cylindrical geometry. Thus, if the definition

$$\alpha_{m\ell} + \alpha_{m, \ell-1} = w_{m\ell} \gamma_{m\ell} (A_{i+1,k} - A_{ik})$$

(22)

is used, the first six terms will have the same form.

To determine $\gamma_{m\ell}$, subtract $2\alpha_{m, \ell-1}$ from both sides of equation (22) and use (19) to obtain

$$\alpha_{m\ell} - \alpha_{m, \ell-1} = -w_{m\ell} \Omega_{r_{m\ell}} (A_{i+1,k} - A_{ik}) = w_{m\ell} \gamma_{m\ell} (A_{i+1,k} - A_{ik}) - 2\alpha_{m, \ell-1}$$

or

$$w_{m\ell} \gamma_{m\ell} (A_{i+1,k} - A_{ik}) = 2\alpha_{m, \ell-1} - w_{m\ell} \Omega_{r_{m\ell}} (A_{i+1,k} - A_{ik})$$

For $\ell = 1$,

$$\alpha_{m, 1} = 0$$

and

$$\gamma_{m, 1} = -\Omega_{r_{m, 1}}$$
For \( t = 2 \) and using (20) to substitute for \( \alpha_m, t-1 \)

\[
    w_m, 2\gamma_m, 2 = -w_m, 2\Omega_{rm, 2} - w_m, 1\Omega_{rm, 1} - w_m, 1\Omega_{rm, 1}
\]

and since \( \gamma_m, 1 = -\Omega_{rm, 1} \)

\[
    w_m, 2\gamma_m, 2 = -w_m, 1\gamma_m, 1 = -(w_m, 2\Omega_{rm, 2} + w_m, 1\Omega_{rm, 1})
\]

Continuing for \( t > 2 \), the relation

\[
    \begin{align*}
        \gamma_m, 1 &= -\Omega_{rm, 1} \\
        w_m, t\gamma_m, t - w_m, t-1\gamma_m, t-1 &= -(w_m, t\Omega_{rm, t} + w_m, t-1\Omega_{rm, t-1}) \quad t > 1
    \end{align*}
\]

(23)

can be obtained.

Solving for \( \psi_{g, i, k, m, t} \) gives the difference equation as

\[
    \psi_{g, i, k, m, t} = \left[ \Omega_{mf}(A_{i+1, k} + A_{ik})\psi_{g, i, k, m, t} + \gamma_{mf}(A_{i+1, k} - A_{ik})\psi_{g, i, k, m, t-1} + \Omega_{mf}(C_{i, k+1} + C_{ik})\psi_{g, i, k, m, t} + \psi_{g, i, k, m, t} \right]
\]

(24)

Equation (24) is the recursion relation for a sweep from \( i \) to \( i+1 \) and from \( k \) to \( k+1 \). Similar results for the other three possible sweep directions are obtained if \( \psi_{g, i, k, m, t} \) or \( \psi_{g, i, k, m, t} \) are replaced in equation (18) with \( \psi_{g, i+1, k, m, t} \) or \( \psi_{g, i, k+1, m, t} \), which then appear in an equation similar to (24). If the absolute values of the direction cosines are used, the same recursion relation may be used for all four possible sweep directions. Similar recursion relations may also be obtained for other geometries (see refs. 4 and 7).

Another equation is required to obtain the initialization at the quadrature boundaries. This equation is obtained from an \( f = 0 \) sweep for which \( \gamma_m, 0 = 0 \) and

\[
    \Omega_{rm, 0} = -\sqrt{1 - \Omega_{rm, 0}^2}.
\]

Equation (24) is used to obtain an average flux over mesh cell \( i, k \) for the discrete direction \( m, t \). Values to be used in the same manner for successive mesh cells and directions are then computed by using equation (21) with \( \psi_{g, i, k, m, t} = \psi_{g, i, k, m, t} \) for the quadrature initialization sweep.

The diamond model used to obtain equation (24) can result in negative values being computed for use in the next mesh cell or direction segment, and this possibility must be
accounted for in the difference equation solution procedure (see page 27 and ref. 10).

The scalar flux $\phi_{gik}$ can be obtained by integrating $\psi_{g,i,k,m,\ell}$ over the directions $m$ and $\ell$ or

$$\phi_{gik} = \sum_{m, \ell} w_{m\ell} \psi_{g,i,k,m,\ell}$$  \hspace{2cm} (25a)

while the currents are

$$J_{r_{gik}} = \sum_{m, \ell} w_{m\ell} r_{m\ell} \psi_{g,i,k,m,\ell}$$  \hspace{2cm} (25b)

and

$$J_{z_{gik}} = \sum_{m, \ell} w_{m\ell} z_{m\ell} \psi_{g,i,k,m,\ell}$$  \hspace{2cm} (25c)

**DIFFERENCE EQUATION SOLUTION PROCEDURE**

The difference equations are solved by carrying out iterations on the flux until convergence has been obtained. The iterations are performed by assuming a source for a sweep through the recursion relations involved in solving the $S_n$ equations. If the source is unchanged by the sweep, it is considered to be converged. By unchanged, one means that the new source and the old source are within a specified small number of being identical. The actual values to be compared can be the integrated source over all the mesh intervals, the integrated source over a given set of mesh intervals, or the new and old source can be compared at each mesh interval.

In the solution, the iterations performed are of two types. These are variously known as outer, major, or power iterations and inner, minor, or group iterations. The source is treated as inhomogeneous, and a portion of it is fixed for each power iteration. The fixed portion of the source is taken to be the fission source. The fixed source is normalized to a constant "power level" at the beginning of each power iteration.

The up- and down-scattering into a given group $g$ may also be considered part of the fixed source for this group. Then, for the group $g$, the fixed source becomes, from equation (13),
\[
\frac{\chi_g}{k_{\text{eff}}} F + T^t_g + T^l_g
\]

with similar results from equation (14) for transport approximation problems and

\[
\nu \Sigma^f_g F^* + T^t_g + T^l_g
\]

from equation (15) for adjoint problems.

To this group fixed source is added the portion of the source for which a new value is obtained for each group iteration. This source results from scattering within the group and becomes

\[
\Sigma_{g \rightarrow g}^{\text{st}} \Phi_g + 3 \Sigma_{g \rightarrow g}^{s1} \sum_q \Omega_q J_q g
\]

from equation (13), with similar results from equations (14) and (15).

The total source for the group iteration is then used for the sweep through the recursion relations of the \( S_n \) method to obtain an improved estimate of the scalar flux and, in turn, of the within-group-scattering portion of the source.

During the sweep through the recursion relations, negative values of the angular flux at mesh-cell and quadrature boundaries may be obtained because of the lack of a significant source, or because of a rapidly varying flux, or because of the lack of convergence of the source. Usually, these conditions may be remedied by rerunning the problem by the use of a finer spatial and, perhaps, angular mesh. However, when these negative values occur during the course of a computation, they can be handled in a number of ways. Two possibilities are allowing the negative values to remain in hopes that they will cease to occur as the source converges and recalculating the angular flux by the step model in which the possibility of a negative flux does not occur (unless the total group source is negative due to a large negative transport corrected \( \Sigma_{g \rightarrow g}^{\text{st}} \) or \( P_1 \) scattering term). The TDSN program replaces a negative angular flux with zero. If the negative values continue to be obtained, a pseudosource and an erroneous gain of neutrons to the system result. In obtaining the new flux estimates, a second or "reflection" sweep through the recursion relations may be required for reflective outer boundary problems to correct for nonzero boundary currents. The combination of the two sweeps is then known as the group iteration.

In addition to an estimate of the source, the boundary angular fluxes are required at boundaries where recursion sweeps begin before the sweep can be started. Zero inward
current conditions require only that the angular flux be set to zero for the inward directed angles. Reflective boundary conditions, however, imply an inward source of neutrons at the boundary that must be obtained from an estimate of the inward angular fluxes. The estimate of these angular fluxes must also be improved, along with that of the scalar flux, as the group iterations are continued.

If, after the sweep through the recursion relations has been completed, the neutron flow across the reflective boundary is not sufficiently close to zero, the reflective sweep may be performed as a correction. For this sweep, the last estimate of the boundary angular fluxes is used without any other source terms to compute a correction flux. The fluxes and flows from this sweep are then combined with those of the regular sweep so that the flow across the reflective boundary will be zero.

The group iterations can then be continued until the within-group-scattering portion of the source ceases to change significantly, that is, until it is converged. The number of group iterations required for convergence may be decreased by the application of different acceleration techniques discussed in the next section.

After the final estimate of the scalar flux has been obtained for a group iteration, certain convergence tests are applied. These include tests on the change in the within-group-scattering source and the change in the collision removal that resulted from the iteration. Other tests may be on point or region changes in the source or loss. These tests are described in more detail in the next section. Since the solution should always satisfy neutron conservation from the nature of the transport difference equations, neutron balance cannot be used as a convergence test.

When the tests are satisfied for a given group, the same procedure is followed for the remaining groups. The scattering source into these groups is always based on the latest estimate of the fluxes for the other groups, but the fission source is updated only at the end of the power iteration. Acceleration techniques may also be applied to the power iterations and are discussed in the next section.

After the group iterations have been performed in the last group and any power iteration acceleration has been applied, the fission source is renormalized by first computing the production rate

\[ F_{ik}^P = \sum_{g=1}^{G} \psi_{ik}^f \Sigma_{gik} \Phi_{ik} \]

as in equation (12). Then the growth factor for production is obtained by the ratio
\[ \lambda^P = \frac{\sum_{i,k} F^P_{ik} V_{ik}}{\sum_{i,k} F^{P-1}_{ik} V_{ik}} \]

The estimate of the multiplication factor \( k_{\text{eff}} \) resulting from the \( P^{\text{th}} \) iteration is then given by \( k_{\text{eff}}^P = \lambda^P k_{\text{eff}}^{P-1} \). Using this new value of \( k_{\text{eff}} \) yields a normalized total fission source equal to the fixed power level.

For homogeneous problems, a nonvanishing solution only exists for certain unique values of \( k_{\text{eff}} \) and for only one of these values, corresponding to the fundamental mode, will the fluxes be everywhere positive. The latest estimate of \( k_{\text{eff}} \) is used to obtain the fixed source estimate of the following power iteration. The power iterations are repeated until \( \lambda^P \) converges to 1 (indicating no change in the fission source for the iteration).

Several tests are applied to determine the convergence of \( \lambda^P \). Among these can be a test on the closeness of \( \lambda^P \) to 1, a test on the rate of change of \( \lambda \), and a test on the closeness of the group sum of the scattering source to the group sum of the scattering loss if the source does not contain extra neutrons from \( n \rightarrow 2n \) scattering reactions. A test on the change in the up-scattering source resulting from the power iteration may also be applied.

**ACCELERATION AND TESTS OF CONVERGENCE**

In this section, the application of scaling and overrelaxation techniques to accelerate the convergence of the inner and outer iterations is discussed along with the convergence tests applied by the TDSN program.

In the past, the only form of acceleration used has been the scaling factor applied to retain conservation of neutrons. Carlson discusses this use of scaling in reference 4 and in the remarks following reference 2. The scaling factor \( f_g \) is an integral quantity being determined from the summation of the changes that result in the within-group-scattering source because of a group iteration and is written as

\[
f_g = \sum_{i,k} \left( \frac{\chi_g}{k_{\text{eff}}} F_{ik} + T_{gik}^t \right) V_{ik}
\]

\[= \sum_{i,k} \left( \frac{\chi_g}{k_{\text{eff}}} F_{ik} + T_{gik}^t \right) V_{ik} - \sum_{i,k} \gamma_{g-g} \left( \Phi_{gik}^{\text{new}} - \Phi_{gik}^{\text{old}} \right) V_{ik} \]  

(26)
where \( V_{ik} \) is the volume of the mesh cell \( i,k \). The flux estimate \( \Phi_{gik}^{\text{new}} \), as well as the currents and angular boundary fluxes, resulting from a group iteration, are accelerated when multiplied by this scaling factor since it further changes the estimate in the same direction as the recursion sweep did on an integral basis. The effectiveness of this scaling factor may be negligible after a number of inner iterations. As convergence proceeds, the scaling factor quickly approaches 1 because increases in the within-group-scattering source in one region can be offset by decreases in another region with the overall flux level remaining almost constant. Thus no acceleration results.

Further acceleration may be achieved through point-wise techniques that will significantly speed up the convergence process. The method used is "overrelaxation" (ref. 11) for iterations in which a specified degree of convergence has been attained. Overrelaxation uses the fluxes before and after an iteration. The accelerated flux estimate is then used in determining the integrated scaling factor. After partial flux convergence has been attained a more optimum acceleration parameter may be calculated by the method of Wachspress described in reference 12.

When point-wise extrapolations of the fluxes are used, other point-wise quantities such as the boundary angular fluxes and currents will need to be corrected. The manner by which the correction factor for the overrelaxation is obtained is illustrated as follows. The extrapolated flux \( \Phi_{gik}^{e} \) for mesh interval \( i,k \) is given by

\[
\Phi_{gik}^{e} = \omega \Phi_{gik}^{p} - (\omega - 1) \Phi_{gik}^{p-1}
\]  

(27)

where \( p \) is the present inner iteration. After the extrapolation is performed, \( \Phi_{gik}^{p} \) will no longer be available to the program having been displaced by \( \Phi_{gik}^{e} \). In computing the scaling factor \( f_{g} \), \( \Phi_{gik}^{e} \) will become \( \Phi_{gik}^{\text{new}} \) and \( \Phi_{gik}^{p-1} \) will become \( \Phi_{gik}^{\text{old}} \). Angular fluxes and currents must also be extrapolated to correspond to the extrapolated scalar fluxes. For the angular boundary flux, the scalar fluxes at the midpoint of the corresponding mesh interval will be used in the computation of the extrapolation factor. This factor is \( E_{ik} = \Phi_{gik}^{e} / \Phi_{gik}^{p} \). Since \( \Phi_{gik}^{p} \) is no longer available, it must be obtained from the relation \( \Phi_{gik}^{p} = [\Phi_{gik}^{e} + (\omega - 1)\Phi_{gik}^{p-1}] / \omega \). With \( f_{g} \) being the scaling factor and \( \Phi_{gik}^{s} \) the scaled flux

\[
\Phi_{gik}^{s} = f_{g} \Phi_{gik}^{e} = f_{g} E_{ik} \Phi_{gik}^{p} = f_{g} E_{ik} \frac{\Phi_{gik}^{e} + (\omega - 1)\Phi_{gik}^{p-1}}{\omega}
\]

and

30
\[ E_{ik} = \frac{\omega \Phi_{gik}}{\Phi_{gik} + (\omega - 1)\Phi_{p-1}} \]

Using \( E_{ik} \) gives the scaled extrapolated currents as

\[ J^S_{r_{gik}} = \frac{f}{g} E_{ik} J^p_{r_{gik}} \]

and

\[ J^S_{z_{gik}} = \frac{f}{g} E_{ik} J^p_{z_{gik}} \]

and similarly the angular boundary fluxes.

The best value of the acceleration factor \( \omega \) is usually not known (ref. 13). Therefore, estimates of \( \omega \) must be made by some means (such as previous experience with similar problems) while being careful not to overestimate the amount of acceleration that can be used for a given problem. Otherwise instabilities may be introduced into the flux iterations. The overrelaxation factor should be chosen smaller for less converged problems than for more converged problems. This factor is also dependent on the reactor system under study. As the rate of convergence decreases and as the problem becomes more converged, the acceleration may be increased. A method for performing this operation is that of Wachspress as described in reference 12. In this method, new acceleration factors \( \omega^{p+1} \) are computed as convergence proceeds that depend on the original factors \( \omega_0 \) and the rate of flux convergence so that

\[ \omega^{p+1} = \frac{2}{1 + \sqrt{1 - \alpha^2}} \tag{28} \]

where

\[ \alpha = \frac{\sigma^2 + (\omega_0 - 1)}{\sigma \omega_0} \]

and
Two types of acceleration may be applied to the outer iterations. One of these is overrelaxation while the other is up-scattering scaling. Overrelaxation of the scalar flux is applied immediately after the inner iterations in a given lethargy group have been completed. The extrapolated flux is given by \( \Phi_{gik}^e = \omega \Phi_{gik}^P - (\omega - 1)\Phi_{gik}^{P-1} \). This group flux is used immediately for computing the down-scattering source but updating the fission source is deferred to the end of the outer iteration. Acceleration of the fission source after each outer iteration occurs through the use of the extrapolated fluxes. The currents and angular boundary fluxes must also be accelerated as in the inner iterations for a group. The acceleration parameter \( \omega \) used will usually be less than that used for the inner iterations as the outer iterations constitute a larger, less converged problem than do the inner iterations.

Since the scattering source into a group is based on the latest flux estimates of the group in which the scattering occurs, a down-scattering source (neutrons from energy degrading scattering) will also be updated in a given outer iteration before it is used. However, an up-scattering source in which the scattered neutrons have a greater energy than before the scattering collision will not be updated during the outer iteration. A slow rate of convergence results that increases the number of outer iterations required. Scaling those groups that up-scatter to obtain a better estimate of the up-scattering source for the next outer iteration improves the convergence rate. The up-scattering scale factor \( f_{up} \) is applied in the same manner as the within-group-scattering scaling of the inner iterations and is written as

\[
f_{up} = \frac{\sum_{g=1}^{G} \sum_{i,k} \sum \frac{\chi_g}{k_{eff}} F_{ik} V_{ik}}{\sum_{g=1}^{G} \sum_{i,k} \sum \frac{\chi_g}{k_{eff}} F_{ik} V_{ik} - \sum_{g=1}^{G} \sum_{i,k} \Sigma_{up} (\Phi_{gik}^{new} - \Phi_{gik}^{old}) V_{ik}}
\]

where \( \Sigma_{up}^g \) is the total up-scattering removal cross section for group \( g \), and \( \Phi_{gik}^{old} \) and \( \Phi_{gik}^{new} \) refer to the fluxes before and after the outer iteration.
The preceding discussion considers general methods for accelerating the convergence of the inner and outer iterations. However, the problem of determining when the convergence is satisfactory remains. Two aspects of the problem are to devise general criteria to be applied by the program and to list various quantities in the program output by which the user can make a judgment about the degree of convergence. The latter quantities are described in notes [10] and [11] of the section Program Operation in appendix A.

In the TDSN program, convergence of the group source or flux is indicated by the program if the following conditions are satisfied. First, the mesh interval for which the largest change in the within-group-scattering source occurred as a result of an inner iteration is determined. Then the ratio of this change to the total integrated group flux is found. This ratio $\delta_1$ is the first quantity that must satisfy a convergence test and is written as

$$\delta_1 = \frac{\left| \sum_{g-g}^{st} \left( \phi_{gik}^P - \phi_{gik}^{P-1} \right) V_{ik} \right|_{\max}}{\sum_{i,k} \phi_{gik}^P V_{ik}}$$

(30)

Second, two integrated quantities are determined. One of these quantities $\delta_2$ is the ratio of the integrated change in the within-group-scattering source occurring as a result of an inner iteration to the total integrated group flux:

$$\delta_2 = \frac{\sum_{i,k} \left| \Sigma_{g-g}^{st} \left( \phi_{gik}^P - \phi_{gik}^{P-1} \right) V_{ik} \right|}{\sum_{i,k} \phi_{gik}^P V_{ik}}$$

(31)

The other is the ratio of the integrated change in the total group removal to the integrated group flux:

$$\delta_3 = \frac{\sum_{i,k} \left| \Sigma_{g}^{t} - \Sigma_{g-g}^{st} \left( \phi_{gik}^P - \phi_{gik}^{P-1} \right) V_{ik} \right|}{\sum_{i,k} \phi_{gik}^P V_{ik}}$$

(32)
Both $\delta_2$ and $\delta_3$ must satisfy a convergence test. Third, if any of the outer boundaries are reflective, the leakage across the boundary must satisfy a test.

Convergence of the outer iterations is indicated by the program if the following conditions to be tested after each outer iteration are satisfied. First, all the group fluxes must be converged as determined by the preceding tests. This is an important criterion since the group iterations are stopped after a specified number of inner iterations in a given group. Thus, the group flux may not be well converged, and if the group has only a small effect on the eigenvalue, the tests on the eigenvalue may terminate the problem. Thus, unconverged fluxes could be included in the output for some groups if this test was not included. Second, the final growth factor $\lambda^P$ must not differ from 1 or its previous value $\lambda^{P-1}$ by more than some specified amount $\epsilon$:

$$\delta_4 = |\lambda^P - 1| < \epsilon \quad \text{and} \quad \delta_5 = |\lambda^P - \lambda^{P-1}| < \epsilon$$

(33)

Third, the change $\delta_6$ in the total integrated scattering loss from one iteration to the next, as given by the ratio

$$\delta_6 = \frac{\sum_{g=1}^{G} \sum_{i,k} \Sigma_{gik}^r \Phi_{gik}^{P-1} V_{ik}}{\sum_{g=1}^{G} \sum_{i,k} \Sigma_{gik}^r \Phi_{gik}^P V_{ik}} < \epsilon$$

(34)

must satisfy a convergence test. For some problems, for which there is no $\Sigma_{gik}^{n-2n}$, the scattering loss must match the scattering source. Also, if up-scattering scaling is used, the ratio $\delta_7$ of the total integrated change in the up-scattering source as summed over all groups to the total integrated flux must be small:

$$\delta_7 = \frac{\sum_{g=1}^{G} \sum_{i,k} \Sigma_{gik}^{\text{up}} \Phi_{gik}^P - \Phi_{gik}^{P-1} V_{ik}}{\sum_{g=1}^{G} \sum_{i,k} \Phi_{gik}^P V_{ik}} < \epsilon$$

(35)
### TABLE I. - NEUTRONS EMMITED PER COLLISION (c OF 1.1 CROSS SECTIONS) FOR CRITICAL SIZE CALCULATIONS

<table>
<thead>
<tr>
<th>Group</th>
<th>$\Sigma^a$</th>
<th>$\nu \Sigma^f$</th>
<th>$\Sigma_{tr}$</th>
<th>$\Sigma_{g+2-g}$</th>
<th>$\Sigma_{g+1-g}$</th>
<th>$\Sigma_g$</th>
<th>$\Sigma_{g-1-g}$</th>
<th>$\Sigma_{g-2-g}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.1</td>
<td>1.0</td>
<td>0.333333</td>
<td>0.333333</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>0.1</td>
<td>1.0</td>
<td>0.333333</td>
<td>0.333333</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>0.1</td>
<td>1.0</td>
<td>0.333333</td>
<td>0.333333</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### TABLE II. - CRITICAL SIZE IN MEAN-FREE PATHS AND PERCENT DEVIATION

<table>
<thead>
<tr>
<th>Number of neutrons emitted per collision, c</th>
<th>Quadrature type, $S_{16}$</th>
<th>Sphere</th>
<th>Plane</th>
<th>Cylinder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Solution</td>
<td>Deviation, $\delta$, percent</td>
<td>Solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TDSN</td>
<td>Exact (a)</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>One dimensional</td>
<td>4.8723</td>
<td>4.8727</td>
<td>-.01</td>
</tr>
<tr>
<td></td>
<td>Two dimensional</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>1.6</td>
<td>One dimensional</td>
<td>1.4753</td>
<td>1.4761</td>
<td>-.05</td>
</tr>
<tr>
<td></td>
<td>Two dimensional</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
</tr>
</tbody>
</table>

$^a$Ref. 2.
DISCUSSION OF SAMPLE PROBLEMS

Problems have been selected to illustrate various features of the TDSN program. Accuracy was tested by comparing results for homogeneous problems with those of reference 2. The effect of increasing the order of quadrature \( n \) is illustrated by the change in the flux shape with the change in \( S_n \) for a sample problem. The effect of the acceleration procedures in TDSN is illustrated by other sample problems.

Carlson and Bell (ref. 2) examine the accuracy of the \( S_n \) method by comparing \( S_n \) results for a range of test problems with the results obtained by other methods. For the purpose of testing the TDSN program, results for the critical dimension \( R \) for uniform spheres, plane slabs, and cylinders without reflectors were obtained for two values of \( c \), where

\[
c = \frac{\left( \nu \Sigma_f^g + \Sigma_b^g \right)}{\Sigma_i^g}
\]

Three groups were used with \( S_{16} \) moment modified quadrature and 32 intervals. The values of \( c \) chosen were 1.1 and 1.6. The cross sections for \( c = 1.1 \) are given in table I. Those for \( c = 1.6 \) used \( \nu \Sigma_f^g \) of 0.6. Table II gives the percent derivation \( \delta \) of the TDSN results from the exact values quoted by Carlson and Bell.

Also included in the table are results for plane slabs using the quadrature necessary for the cylindrical problems. This quadrature is the same one that would be used with two-dimensional problems, and, as the table shows, it has little effect on the plane-geometry problems. The percent deviations \( \delta \) for the three-group problems using the correct quadratures are comparable with those given by Carlson and Bell for one group problems in table 5 of reference 2.

Figure 1 shows the scalar flux as a function of radius for a one-group cylindrical-cell problem with five different material regions. A flat fixed source guess was used in the outer region, while the cross sections correspond to the thermal cross sections of a sample physical problem. The boundary condition at the outer radius was the isotropic return condition. The problem was computed by using moment modified quadratures of \( S_2, S_4, S_8, S_{16}, \) and \( S_{32} \). The flux shapes for \( S_2, S_4, \) and \( S_{32} \) are illustrated, while the \( S_8 \) and \( S_{16} \) results were nearly identical with the \( S_{32} \) result. The \( S_4 \) result follows \( S_{32} \) quite closely, while \( S_2 \) deviates from \( S_{32} \) in the smaller, scattering regions.

Three different examples will be used to illustrate the effect of accelerating techniques on the number of iterations required for convergence. The first illustrates the effect of overrelaxation of the inner and outer iteration fluxes on a small one-dimensional
reflected thermal-reactor problem. The second shows the effect of overrelaxation on a two-dimensional problem, and the third shows the effect of up-scattering scaling on a multithermal-group problem.

To show the effect of overrelaxation, a one-dimensional problem with three regions consisting of a cylindrical fueled core, thin aluminum wall, and water reflector was chosen with 25 mesh intervals, $S_4$ quadrature and two fast groups and one thermal group. This problem was chosen as representative of thermal problems that are usually more difficult to converge than fast-reactor problems. Overrelaxation was first applied only to the fluxes of the inner iterations. The results are given by curve A of figure 2. The curve is able to extend beyond $\omega = 2$ because TDSN only applies the inner iteration acceleration factor where a certain degree of convergence is maintained. Curves C and D are the result of also applying overrelaxation to the outer iteration fluxes with the value of $\omega$ for the inner iterations held fixed.

Curve B is the result of first applying the given overrelaxation factor to the inner iterations as soon as the degree of convergence required is attained and then computing a new group dependent overrelaxation factor by the method of Wachspress (ref. 12) and applying it as soon as a second degree of convergence is attained.

The number of iterations required for convergence of the fluxes, shown as curve A, had a minimum of approximately 56 percent of those required for the unaccelerated problem corresponding to point I. For those cases involving overrelaxation of both the inner and outer iteration fluxes the minimum was approximately 36 percent of the value shown as point I.

The effect of overrelaxation on a two-dimensional problem was obtained by running...
two similar problems made up of three radial regions consisting of a cylindrical fueled core, thin aluminum wall, and water reflector, and reflected axially about the midplane. Each problem contained 33 radial and nine axial mesh intervals with $S_4$ quadrature and four fast groups and one thermal group. The first problem was calculated by using $\omega$ as 1 for both the inner and outer iterations, that is, the unaccelerated case. The second problem was calculated by using the method of Wachspress. The number of iterations required for convergence of the fluxes for the accelerated case was approximately 67 percent of those required for the unaccelerated case.

The effect of up-scattering scaling on a multithermal group problem is illustrated by a 15-group one-dimensional cylindrical-cell problem with 10 groups that up-scatter. The inner structure of the cell contains rings of fissionable material and absorbing material separated by void spaces. One hundred mesh intervals were used to represent the cell, and an overrelaxation factor of 1.3 was applied to the inner iterations. Without up-scattering scaling, 3325 inner iterations and 49 outer iterations were required. Using the scaling required 915 inner iterations and 11 outer iterations. The problems that used up-scattering scaling converged in less than one-third of the time required without it. Full-core-radial problems involving flux-weighted cross sections from the cell problems converged in about one-half the time if up-scattering scaling was applied.

**CONCLUDING REMARKS**

The most important feature of the TDSN program is the convergence acceleration achieved from overrelaxation and scaling. Additional savings can accrue by testing group
convergence to prevent some groups from being converged to a much greater extent than others.

Sample problems have shown that accelerated problems can converge with about two-thirds the inner iterations of a nonaccelerated problem with only overrelaxation of the inner iterations and with less than half of the inner iterations of a nonconverged one-dimensional problem by using overrelaxation of both the inner and outer iterations. The effect of up-scattering scaling on multithermal group problems can also reduce the time required for convergence to less than one-half of that required without it.

Another important feature of the TDSN program permits a problem to be ended after a given number of inner iterations or a given elapsed time. Any output desired may then be obtained no matter what stage of convergence has been reached and a restart binary card dump is punched if requested. If it can be determined that the problem is sufficiently converged, all the output is available and, if not, the problem can be restarted with the restart dump. At restart, the convergence and acceleration parameters and output options desired may be changed.

Details of the TDSN program are given in the appendixes. Appendix A gives the input instructions, output description, and notes on program operations. Appendix B lists portions of the Set A and Set B quadrature tables of reference 2 in the format required for TDSN. Appendix C lists the names of program decks required by TDSN along with their programming language and a brief statement of their purpose. Appendix D is a listing of the program decks, and appendix E presents the input and output of a sample problem.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, May 2, 1966,
122-28-03-05-22.
APPENDIX A

TDSN OPERATING INSTRUCTIONS

Discussion of Input

The numbers in brackets in the remarks column designate notes in the section Program Operation that may be of value to the program user. Each Roman numerical refers to a new card or to a new subscripted variable that may require one or more cards. The symbol * means to use as much of the card and as many cards as necessary to read in the subscripted variable. See reference 14 for a description of FORTRAN IV names and formats, and reference 15 for a description of the operating system and processor.

I. Problem identification cards. These cards may contain any numeric or alphabetic information in card column 3 to 72. A number in card column one indicates the final identification card. (The format is I1, 1X, 14A5.)

II. BCREAD control card.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card Columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBCRED</td>
<td>I10</td>
<td>1-10 = 0</td>
<td>Read in cards III to XIX (new problem).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 1</td>
<td>Read in BCDUMP for restart instead of cards III to XIX. [1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 2</td>
<td>Read in 2 cards following the first 2 parts of the BCDUMP to change some of the controls on the problem. These two cards are listed as XX and XXI.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 3</td>
<td>Read in 3 cards following the first 2 parts of the BCDUMP. The first 2 are the same as those for KBCRED = 2, while the third card changes the output to be obtained and is the same as card VII.</td>
</tr>
</tbody>
</table>

If KBCRED equals 0, go to card III; if it is equal to 1, go to card XXIII; if it is equal to 2 or 3, go to card XX.

III. Control card.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card Columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>KREG</td>
<td>I10</td>
<td>1-10 = 0</td>
<td>A regular flux calculation is performed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 1</td>
<td>An adjoint calculation is performed.</td>
</tr>
<tr>
<td>Quantity</td>
<td>Format</td>
<td>Card columns</td>
<td>Remarks</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>KALC</td>
<td>I10</td>
<td>11-20 = 1</td>
<td>Calculate the multiplication constant $k_{eff}$ as the eigenvalue (only option available).</td>
</tr>
<tr>
<td>KGEO</td>
<td>I10</td>
<td>21-30 = 1</td>
<td>Slab $x,y$ geometry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2</td>
<td>Cylindrical $r,z$ geometry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3</td>
<td>Sphere $r$ geometry</td>
</tr>
<tr>
<td>KLBC</td>
<td>I10</td>
<td>31-40 = 0</td>
<td>No return current, left boundary condition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1</td>
<td>Plane mirror (''perfect'') reflection, left boundary condition</td>
</tr>
<tr>
<td>KRBC</td>
<td>I10</td>
<td>41-50 = 0, 1</td>
<td>Same as KLBC for right boundary condition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3</td>
<td>Isotropic (''white'') reflection, right boundary condition [2]</td>
</tr>
<tr>
<td>KBBC</td>
<td>I10</td>
<td>51-60 = 0, 1</td>
<td>Same as KLBC for bottom boundary condition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 = 4</td>
<td>$180^\circ$ rotational symmetry, bottom boundary condition [2]</td>
</tr>
<tr>
<td>KTBC</td>
<td>I10</td>
<td>61-70 = 0, 1, 3</td>
<td>Same as KRBC for top boundary condition</td>
</tr>
</tbody>
</table>

**IV. Control card.**

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSN</td>
<td>I10</td>
<td>1-10</td>
<td>This is the order $n$ of $S_n$ to be used. It must be an even integer.</td>
</tr>
<tr>
<td>NG</td>
<td>I10</td>
<td>11-20</td>
<td>The number of energy groups</td>
</tr>
<tr>
<td>NR</td>
<td>I10</td>
<td>21-30</td>
<td>The number of mesh intervals for the first direction [3]</td>
</tr>
<tr>
<td>NZ</td>
<td>I10</td>
<td>31-40 = 0</td>
<td>A one-dimensional calculation will be performed with one-dimensional quadrature (for slabs and spheres only).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1</td>
<td>A one-dimensional calculation will be performed with two-dimensional quadrature (for slabs and cylinders only).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; 1</td>
<td>The number of mesh intervals for the second direction of a two-dimensional calculation [3]</td>
</tr>
<tr>
<td>NMAT</td>
<td>I10</td>
<td>41-50</td>
<td>The number of different materials to be used [4]</td>
</tr>
<tr>
<td>NTYPS</td>
<td>I10</td>
<td>51-60</td>
<td>The number of cross section types per material per group [5]</td>
</tr>
<tr>
<td>Quantity</td>
<td>Format</td>
<td>Card columns</td>
<td>Remarks</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>N2N</td>
<td>I10</td>
<td>61-70 = 1</td>
<td>Read in total out-scattering cross sections along with the other cross sections for KCTYP = 1 or at XVI(4) for KCTYP ≠ 1. [6]</td>
</tr>
<tr>
<td><strong>V. Control card.</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCTYP</td>
<td>I10</td>
<td>1-10 = 0</td>
<td>The cross sections will be in TDSN format [5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1                                                                                                                                                    The cross sections to be supplied in the format of the RP1 diffusion-theory program [7]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 2                                                                                                                                                    The cross sections to be supplied in the format of the DTF and DDF programs [5]</td>
</tr>
<tr>
<td>KSISO</td>
<td>I10</td>
<td>11-20 = 0</td>
<td>All materials use only $P_0$ cross sections.                                                                                                    At least one material uses $P_1$ cross sections.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1</td>
</tr>
<tr>
<td>KGST</td>
<td>I10</td>
<td>21-30</td>
<td>The number of the group to begin the outer iteration. Almost always 1 (the highest energy group) for KREG = 0 and NG for KREG = 1.</td>
</tr>
<tr>
<td>KCTR</td>
<td>I10</td>
<td>31-40</td>
<td>The position of $\Sigma_g^t$ (or $\Sigma_{gtr}^t$) for $g = 1$ in the cross-section matrix. [5]</td>
</tr>
<tr>
<td>KCGG</td>
<td>I10</td>
<td>41-50</td>
<td>The position of $\Sigma_{g-g}^{\text{st}}$ in the cross section matrix. [5]</td>
</tr>
<tr>
<td>NBUCK</td>
<td>I10</td>
<td>51-60 = 0</td>
<td>No buckling of this type is to be used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= -1                                                                                                                                                    Buckling dimensions to be read in by card X and the KCTR field of $P_1$ scattering cross section sets should contain $\Sigma_{g}^{\text{tr}}$. [5] and [8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1                                                                                                                                                    Buckling-loss cross sections are to be read-in for NMAT materials along with the cross sections for KCTYP = 1 or at XVIII for KCTYP ≠ 1.</td>
</tr>
<tr>
<td>LBUCK</td>
<td>I10</td>
<td>61-70 = 0</td>
<td>No buckling of this type is to be used.</td>
</tr>
<tr>
<td>Quantity</td>
<td>Format</td>
<td>Card columns</td>
<td>Remarks</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>Quanuty</td>
<td>Format</td>
<td>Card columns</td>
<td>Remarks</td>
</tr>
<tr>
<td>-1</td>
<td>Print and punch in the output the buckling-loss cross sections in the first direction to be used later as the buckling-loss cross sections in the second direction of another problem run in a direction normal to the first problem.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Read-in the buckling-loss cross sections for the second direction as given by a previous problem at XVII, and print and punch in the output the buckling-loss cross sections in the first direction as for LBUCK = -1.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VI. Control card.

<table>
<thead>
<tr>
<th>ITMIMX</th>
<th>I10</th>
<th>1-10</th>
<th>The maximum number of inner iterations to be allowed before the problem is stopped; 4000 will be used if zero is supplied here [10] and [11]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITMIGM</td>
<td>I10</td>
<td>11-20</td>
<td>The maximum number of inner iterations to be allowed in a group per outer iteration; 20 will be used if zero is supplied here [9] and [11]</td>
</tr>
<tr>
<td>NZONER</td>
<td>I10</td>
<td>21-30</td>
<td>The number of input zones for the first direction [12]</td>
</tr>
<tr>
<td>NZONEZ</td>
<td>I10</td>
<td>31-40</td>
<td>The number of input zones for the second direction (zero if NZ = 0 or 1) [12]</td>
</tr>
<tr>
<td>KDREAD</td>
<td>I10</td>
<td>41-50 = 0</td>
<td>Use moment modified quadrature for direction cosines and weights as calculated by program.</td>
</tr>
<tr>
<td>KFLUXI</td>
<td>I10</td>
<td>51-60 = 0</td>
<td>Use an initial flux guess of constant value over all mesh intervals and groups as supplied by the program.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Supply a binary card dump of the fluxes of a previous problem that had the same number of mesh intervals and groups for the initial flux guess. This is done at XIX (1). [1]</td>
</tr>
<tr>
<td>Quantity</td>
<td>Format</td>
<td>Card columns</td>
<td>Remarks</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>KALC1</td>
<td>I10</td>
<td>61-70</td>
<td>Supply an initial flux guess at XIX(2) To be used for KALC &gt; 1 (not available at present).</td>
</tr>
<tr>
<td>GLAM</td>
<td>E10.6</td>
<td>1-10</td>
<td>The value of $\lambda$ to be used for problems other than KALC = 1. On restart, the final value of $\lambda$ of the previous run is given in the output.</td>
</tr>
<tr>
<td>EIGEN</td>
<td>E10.6</td>
<td>11-20</td>
<td>The eigenvalue to be used for problems other than KALC = 1. On restart, the final estimate of the eigenvalue of the previous run is given in the output.</td>
</tr>
<tr>
<td>EPS</td>
<td>E10.6</td>
<td>21-30</td>
<td>The convergence criteria for the outer iterations [10]</td>
</tr>
<tr>
<td>XNF</td>
<td>E10.6</td>
<td>31-40</td>
<td>The normalization factor; the number of fission neutrons at the beginning of an outer iteration is normalized to XNF.</td>
</tr>
<tr>
<td>RYF</td>
<td>E10.6</td>
<td>41-50</td>
<td>The ratio factor used in one of the convergence checks as listed in note [10]</td>
</tr>
<tr>
<td>TIMAX</td>
<td>E10.6</td>
<td>51-60</td>
<td>The maximum time in minutes for a problem to be allowed to run. The problem will be stopped after the outer iteration, in which TIMAX was exceeded, has been completed. Enough time should be allowed on a system time card to complete an outer iteration and give all the output requested. If it is zero, 1.5 minutes will be used. [10] and [11]</td>
</tr>
<tr>
<td>CALC1</td>
<td>E10.6</td>
<td>61-70</td>
<td>To be used in conjunction with KALC1.</td>
</tr>
<tr>
<td>OMEGA</td>
<td>E10.6</td>
<td>1-10</td>
<td>The relaxation parameter to be used for inner iterations. It is not used unless KACCEL $\neq$ 0. OMEGA $&gt; 1$ for overrelaxation. [11]</td>
</tr>
</tbody>
</table>

VII. Control card.

VIII. Control card.
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMEGAP</td>
<td>E10.6</td>
<td>11-20</td>
<td>The relaxation parameter to be used in accelerating the outer iterations. It is not used unless KACCEL = ±2. [11]</td>
</tr>
<tr>
<td>KACCEL</td>
<td>I10</td>
<td>51-60 = 0</td>
<td>No acceleration of the inner or outer iterations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= ±1</td>
<td>Relaxation of the inner iterations [11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= ±2</td>
<td>Relaxation of the inner and outer iterations; -1 or -2 to calculate $\omega^{p+1}$ for the inner iterations by the method of Wachspress (ref. 12) [11]</td>
</tr>
<tr>
<td>KUPS</td>
<td>I10</td>
<td>61-70 = 0</td>
<td>Up-scattering scaling will not be performed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; 0</td>
<td>Scaling of the outer iteration fluxes for those groups that up-scatter will be performed.</td>
</tr>
</tbody>
</table>

**IX. Output control card.**

<p>| ITMPRT   | I10    | 1-10 = 0    | No intermediate output is to be printed. |
|          |        | = 1         | A two-line edit of each outer iteration is to be printed. |
|          |        | = 2         | In addition to the preceding, a line edit that gives the inner iteration relaxation parameter to be applied during the next inner iteration in a given group along with convergence data is to be printed. This is given if overrelaxation was applied during the iteration. |
|          |        | = 3         | In addition to the preceding, all the negative angular fluxes calculated during the last inner iteration in each group are printed out along with the corrections applied. The group scalar fluxes must be within approximately a factor of 10 of meeting the convergence criteria for the print out to be given. |
| KBCDUP   | I10    | 11-20 = 0   | No binary card dump will be punched. |
|          |        | = -1        | A binary card dump of the scalar fluxes will be punched for the last outer iteration performed. It is to be used in another problem with KFLUXI = -1. [1] |</p>
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 1</td>
<td></td>
<td>= 2</td>
<td>A binary card dump for restart will be punched if the problem stops before the outer iterations are converged (see II. p. 40). [1] and [10]</td>
</tr>
<tr>
<td>= 2</td>
<td></td>
<td>= 2</td>
<td>Same as KBCDUP = 1, except that the card dump will be given even if the problem is converged. [1] and [10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= -2, 2</td>
<td>No activity or averaging tables is to be given in the output. Activity tables will be given in the output along with disadvantage factors. Averaging tables will be given in the output and punched. These tables give flux-weighted average cross sections by the zones requested plus a set averaged over all zones (only given if KREG = 0). [13] Both sets of tables will be given</td>
</tr>
<tr>
<td>KACAV</td>
<td>I10</td>
<td>21-30 = 0</td>
<td>If KACAV &gt; 0 is used, the zones will be by material, and if KACAV = 2 or 3, only the set averaged over all zones will be punched. If KACAV &lt; 0, a separate zone map for each type of table requested must be read in at XXIV.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= -3, 3</td>
<td></td>
</tr>
<tr>
<td>KGAVE</td>
<td>I10</td>
<td>31-40 = 0</td>
<td>No group collapsed cross sections will be given. The number of groups of collapsed cross sections. The identification numbers must be read in at XXIII. Punched output is given. If averaging tables are requested, the same zones will be used. If averaging tables are not requested, the collapsing will be done by materials. The collapsed cross section ( \Sigma^{st}_{g-g} ) will not be correct if ( n - 2n ) scattering is present (only given if KREG = 0). [13]</td>
</tr>
<tr>
<td>KNIP</td>
<td>I10</td>
<td>41-50 = 0</td>
<td>No flux integrals are punched or effective axial bucklings printed in the output. The fluxes as integrated for each material are printed and punched by group for each material.</td>
</tr>
</tbody>
</table>

46
X. Buckling dimensions. This card is a part of the input only if NBUCK = -1.

HZ    E10.6  1-10  The first buckling dimension which is the second dimension of a one-dimensional problem and the third dimension of a two-dimensional problem.

HY    E10.6  11-20  The second buckling dimension should be zero except for one-dimensional slabs.

BF    E10.6  21-30  If buckling factor is zero, \( \pi/\sqrt{3} \) will be used. [8]

XI. Specifications for first dimension. One card of this type is included for each of the NZONER input zones.

NMI    I5     1-5     The number of mesh intervals in the zone

RMI    E10.6  6-15     The distance from the left boundary of the problem to the outside boundary of the zone

MID    I5     16-20    The material identification of the zone (not used if NZONEZ is greater than 1 [4]
XII. Specifications for second dimension. (No cards are included here unless NZ is greater than 1.) One card of this type is included for each of the NZONEZ input zones.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMI</td>
<td>I5</td>
<td>1-5</td>
<td>The number of mesh intervals in the zone NZOZER values are included per card(s).</td>
</tr>
<tr>
<td>RMI</td>
<td>E10.6</td>
<td>6-15</td>
<td>The distance from the bottom boundary of the problem to the upper boundary of the zone NZOZER values are included per card(s).</td>
</tr>
</tbody>
</table>

XIII. Material specifications for problems for which NZONEZ is greater than 1. (No cards are included here unless NZONEZ is greater than 1.) One card(s) of this type is included for each of the NZONEZ input zones.

| MID | 7110 | 1-10,* | The material identification as described in note [4]; NZONER values are included per card(s). |

XIV. Fission spectrum.

| XKI | 7E10.6 | 1-10,* | NG values are included to give the fission spectrum \( \chi \) by energy group. The sum should be 1. |

XV. Direction cosines and weights. (No cards are included here if KDREAD = 0.) ND values are included for each function. [14]

1. DR 7E10.6 1-10,* The direction cosines to the first direction
2. DZ 7E10.6 1-10,* The direction cosines to the second direction (no cards are included if NZ = 0 or 1).
3. W 7E10.6 1-10,* The direction weights (which must add to 1).

XVI. Cross sections for each material included here.

1. If KCTYP = 0, use this format. NG sets of cards for each material \((P_1 \text{ materials counting as 2 materials})\) are included here with a set covering only one energy group.

| C 7E10.6 | 1-10,* See note [5] for description. |

2. If KCTYP = 1, use this format. Three different sets of cards may be included for each material. [7]

   a. The first set of cards is given as follows and contains NG cards:
The diffusion coefficient
The absorption cross section that will be put in the $C^a$ position.
If $(N_{TYP} - K_{CGG}) = 1$, this is the scattering removal cross section and will be put into the first (and only) $C_{down}$ position. If $N_{2N} = 1$, this is the total out-scattering cross section $C_{2N}$. [6]

The fission neutron cross section that will be put in the $C^f$ position
If $N_{BUCK}$ is greater than zero, the effective buckling cross section is included here. When multiplied by the flux it will yield the transverse leakage.

(b) If $N_{DOWN} = N_{TYP} - K_{CGG}$ is greater than 1, or if $N_{UP} > 0$, the second set of cards consists of the down-scattering cross sections for the material and are included here.

$$\Sigma_{down} \ 7E10.6 \ 1-10,* \ \Sigma_{g-g+1}, \Sigma_{g-g+2}, \Sigma_{g-g+3}, \text{etc.}, \text{until either} \ \text{NDOWN values have been given or} \ \Sigma_{g-NG} \ \text{has been reached.}$$

(c) If $N_{UP} = K_{CGG} - (K_{CTR} + 1)$ is greater than zero, the third set of cards consists of the up-scattering cross sections for the material. The cross sections start for the scattering out of the $(NG - N_{UP} + 1)$-th group to the $NS = (NG - N_{UP})$-th group.

$$\Sigma_{up} \ 7E10.6 \ 1-10,* \ \Sigma_{g-g-1}, \Sigma_{g-g-2}, \text{etc.}, \text{until} \ \Sigma_{g-NS} \ \text{has been reached.}$$

The cross sections for the next material are then given. In each case the cross sections are put in positions as specified by note [5] by the program to work the problem. Also $\Sigma_{g-g}$ is computed for each material and put in the $C_{g-g}$ position. No activity cross sections can be included. [7]

(3) If $K_{CTYP} = 2$, use this format. The order of the cross sections on the cards is the same as that for XVI(1), but the cross sections are divided into sets only by material and not by material and group with $C^{act}$ for the next group ($C^a$ if $K_{CTR} = 3$) beginning
in the next field following the final down-scattering cross section of a given group. The format is 6F12.7. This is the same format as used by DDF and DTF (ref. 16). [5]

(4) If N2N = 1 and if KCTYP ≠ 1, the total out scattering cross sections are supplied here before the next material is included under (1) or (3).

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN2N</td>
<td>7E10.6</td>
<td>1-10, *</td>
<td>NG values [6]</td>
</tr>
</tbody>
</table>

XVII. Effective buckling cross sections if LBUCK = 1 included here.

BUCLK    7E10.6  1-10, *  NG values of BUCLK are given. Each value when multiplied by the flux will yield the transverse leakage by group.

XVIII. Effective buckling cross sections if NBUCK = 1 included here for KCTYP ≠ 1. NMAT sets of cards are included. XVII and XVIII should not both be used for the same problem.

BUCKG    7E10.6  1-10, *  NG values of BUCKG are given in each set. The second set for P₁ materials should contain zeros. Each value when multiplied by the flux will yield the transverse leakage by group and material.

XIX. Initial flux guess (no cards are included if KFLUXI = 0).

(1) The binary card dump of the fluxes of a previous problem with the same values of NG, NR, and NZ is included here if KFLUXI = -1.

(2) If KFLUXI = 1, the flux guess is included here.

N, X      5(I5, E10.6)  (1-5, 6-15), *  N is the number of successive intervals at which to use X for the flux guess. The second value of N (appearing in card columns 16 to 20) gives the number of successive intervals at which to use the next value of X (appearing in card columns 21 to 30), and so forth. If N becomes zero before NR times NZ values have been given, the remaining values for the flux guess are set equal to zero for the group. Repeat for other groups as desired.
If KBCRED = 0, go to XXIII since the following cards are used only for a restart.

XX. Restart control card for KBCRED = 2 or 3.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITMIMX</td>
<td>I10</td>
<td>1-10</td>
<td>Same as for card VI [11]</td>
</tr>
<tr>
<td>ITMIGM</td>
<td>I10</td>
<td>11-20</td>
<td>Same as for card VI [11]</td>
</tr>
</tbody>
</table>

XXI. Restart control card for KBCRED = 2 or 3.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPS</td>
<td>E10.6</td>
<td>1-10</td>
<td>Same as for card VII</td>
</tr>
<tr>
<td>TIMAX</td>
<td>E10.6</td>
<td>11-20</td>
<td>Same as for card VII [11]</td>
</tr>
<tr>
<td>OMEGA</td>
<td>E10.6</td>
<td>21-30</td>
<td>Same as for card VIII [11]</td>
</tr>
<tr>
<td>OMEGAP</td>
<td>E10.6</td>
<td>31-40</td>
<td>Same as for card VIII [11]</td>
</tr>
<tr>
<td>KACCEL</td>
<td>I10</td>
<td>51-60</td>
<td>Same as for card VIII [11]</td>
</tr>
<tr>
<td>KUPS</td>
<td>I10</td>
<td>61-70</td>
<td>Same as for card VIII</td>
</tr>
</tbody>
</table>

XXII. Restart output control card for KBCRED = 3. This card is included only if KBCRED = 3 and is the same as card IX.

The following cards are for special edits and are utilized only after the problem has been completed.

XXIII. Collapsed group identification numbers. Cards are included here if KGAVE > 0.

<table>
<thead>
<tr>
<th>IDGP</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T110</td>
<td>1-10, *</td>
<td>Include NG numbers identifying the KGAVE collapsed group to which each group belongs.</td>
</tr>
</tbody>
</table>

XXIV. Activity table and flux-weighted averaging table identification numbers (cards are needed if KACAV is negative).

(1) The map for the activity table. Cards are needed if KACAV = -1 or -3.

(a) The number of output zones

<table>
<thead>
<tr>
<th>NZONRA</th>
<th>I10</th>
<th>1-10</th>
<th>The number of output zones in the first direction (maximum of 50). [12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>NZONZA</td>
<td>I10</td>
<td>11-20</td>
<td>The number of output zones in the second direction (zero for a one-dimensional problem; maximum of 30). [12]</td>
</tr>
</tbody>
</table>

(b) Mesh interval map.
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Format</th>
<th>Card columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMRA</td>
<td>7I10</td>
<td>1-10,*</td>
<td>The number of mesh intervals for the first direction to be included in each output zone. Include NZONRA values.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(c) Include this card if NZONZA &gt; 0.</td>
</tr>
<tr>
<td>NMZA</td>
<td>7I10</td>
<td>1-10,*</td>
<td>The number of mesh intervals for the second direction to be included in each output zone. Include NZONZA values.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(d) The identification numbers for the zones. Include NZONZA sets of cards (one set if NZONZA = 0).</td>
</tr>
<tr>
<td>IDM</td>
<td>14I5</td>
<td>1-5,*</td>
<td>Include NZONRA values per set of cards identifying which zones (and thereby mesh intervals) are to be combined in obtaining the activity table. Zones with the same value of IDM will be combined.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2) The map for the averaging table. No cards are included here unless KACAV = -2 or -3. If KACAV = -3 both XXIV(1) and XXIV(2) must be included. The explanation of the input is the same as under XXIV(1).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XXV. Input for EDIT subroutine if KEDIT &gt; 0.</td>
</tr>
<tr>
<td>NISET</td>
<td>I10</td>
<td>1-10</td>
<td>The number of activity edits (integration sets) to be performed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1) The number of integrals to include in a given edit set (INT has a maximum of 200).</td>
</tr>
<tr>
<td>INT</td>
<td>I10</td>
<td>1-10</td>
<td>This card will be read in NISET times.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cards (2) and (3) are included in order INT times following each card (1).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2) The integration limits for a given integral.</td>
</tr>
<tr>
<td>IG1</td>
<td>I10</td>
<td>1-10</td>
<td>The upper limit of the groups to be included in a given integral</td>
</tr>
<tr>
<td>IG2</td>
<td>I10</td>
<td>11-20</td>
<td>The lower limit of the groups</td>
</tr>
<tr>
<td>IR1</td>
<td>I10</td>
<td>21-30</td>
<td>The left limit of the first direction mesh intervals to be included in a given integral</td>
</tr>
<tr>
<td>Quantity</td>
<td>Format</td>
<td>Card columns</td>
<td>Remarks</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>IR2</td>
<td>I10</td>
<td>31-40</td>
<td>The right limit</td>
</tr>
<tr>
<td>IZ1</td>
<td>I10</td>
<td>41-50</td>
<td>The lower limit of the second direction mesh intervals to be included in a given integral (one for a one-dimensional problem).</td>
</tr>
<tr>
<td>IZ2</td>
<td>I10</td>
<td>51-60</td>
<td>The upper limit (one for a one-dimensional problem)</td>
</tr>
</tbody>
</table>

(3) Include this card only if KEDIT = 1.

SIGMA F10.6 1-10 The group cross section to be used in the integration. Include one value (on separate cards) for each group within the limits IG1 and IG2.

Description of Output

Each Roman numeral refers to a new section of output. The numbers in brackets again refer to notes in the section Program Operation.

I. Input title and control cards. LAST is the storage required for the subscripted variables of the 17 500 available, and LAST1, LAST2, LAST3, and LAST4 are the storages required by each of the first four overlays for these variables.

II. Dimension specifications by zone as included in input and by mesh interval as calculated by program. Only the latter is given on restart from a binary card dump.

III. Material specifications MA by mesh interval beginning with lower left mesh interval. [3]

IV. Fission spectrum input $\chi$, XXI.

V. All cross section input in TDSN format. If KCTYP = 1, this is preceeded by the cross sections in their input format. Only the TDSN format is given on restart.

VI. If LBUCK = 1, effective buckling cross sections BUCLK are listed; if NBUCK = 1, effective buckling cross sections BUCKG are listed.

VII. Total out-scattering cross sections are listed by group.
VIII. Volumes of mesh intervals as calculated by program are listed beginning at lower left corner. [3]

IX. Direction functions as included in input or as calculated by program are listed.

X. If ITMPRT > 0, two lines of intermediate output for each outer iteration are listed. The output listed is as follows for the first line:

<table>
<thead>
<tr>
<th>ITMA</th>
<th>Number of outer iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITMI</td>
<td>Total number of inner iterations performed [11]</td>
</tr>
<tr>
<td>ITMIGX</td>
<td>Maximum number of inner iterations performed in a group during outer iteration</td>
</tr>
<tr>
<td>ITD</td>
<td>Number of inner iterations per outer iteration</td>
</tr>
<tr>
<td>TIME REQ</td>
<td>Total time required to perform iterations in minutes; time starts over on restart [11]</td>
</tr>
<tr>
<td>EPG</td>
<td>Convergence criterion for inner iterations [9]</td>
</tr>
<tr>
<td>EPGM</td>
<td>Maximum change in within-group-scattering source (or in the collision removal) that has occurred for last inner iteration in a group [9]</td>
</tr>
<tr>
<td>KASOR</td>
<td>If KSISO = 1, KASOR &gt; 0 indicates an adjustment was made to $P_1$ source for final inner iteration in a group; value of KASOR is number of groups in which adjustments were made; KASOR must be zero for convergence</td>
</tr>
<tr>
<td>KONV</td>
<td>Equal to 0, problem unconverged and continuing to run; equal to 1, problem unconverged but stopped by iteration count (ITMIMX); equal to -1, problem unconverged but stopped by TIMAX; equal to 2, problem converged [10]</td>
</tr>
<tr>
<td>KFINSH</td>
<td>Number of groups that did not converge during the inner iterations (see XIV. [9] and [11])</td>
</tr>
<tr>
<td>NEUTRON BAL</td>
<td>Neutron balance as given by note [6]; if large, after convergence, an error such as in cross sections or in setting a large number of negative angular fluxes to zero is indicated</td>
</tr>
<tr>
<td>GLAM</td>
<td>Eigenvalue $\lambda$ that must be within EPS of 1 for convergence, and within EPS of previous value of $\lambda$</td>
</tr>
<tr>
<td>EIGEN</td>
<td>$k_{eff}$</td>
</tr>
</tbody>
</table>
The second line of the two-line output is as follows:

**UPS2**  Up-scattering error; sum of absolute change in up-scattering source over all mesh intervals and groups from before and after an outer iteration; not used if KUPS = 0 [9]

**XNG(NGP1)**  Total flux over all mesh intervals and groups

**SING(NGP1)**  Total in-scattering source over all mesh intervals and groups [9]

**SDG(NGP1)**  Total out-scattering loss [9]

**AG(NGP1)**  Total absorption [9]

**HNLG(NGP1)**  Total leakage along first direction [9]

**VNLG(NGP1)**  Total leakage along second direction [9]

**XNLG(NGP1)**  Total leakage including HNLG, VNLG, and any buckling losses [9]

**XI.** If ITMPRT > 1, the following line is listed during each inner iteration in which overrelaxation has been applied:

**GROUP**  Group to which acceleration was applied

**ITMIG**  Inner iteration for which acceleration was applied

**FLUX ERROR**  If KACCEL > 0, this is the larger of $\delta_2$ and $\delta_3$ (EPGX) (see note [9]) of previous iteration of group; if KACCEL < 0, this is the integrated absolute change in scalar flux that occurred during present iteration

$$\sum_{i,k} |\phi_{gik}^p - \phi_{gik}^{p-1}| V_{ik}$$

**ERROR RATIO**  If KACCEL < 0, this is $\sigma^2$ of equation (28) which is ratio of present FLUX ERROR to previous iteration FLUX ERROR

**FORMER RATIO**  If KACCEL < 0, this is ERROR RATIO of previous iteration

**FACTOR**  If KACCEL < 0 and two ratios are within 0.005 of being identical, this is the parameter $\alpha$ of equation (28) that is used to compute new overrelaxation factor $\omega^{p+1}$.

**OMEGA**  Value of overrelaxation factor $\omega^{p+1}$ to be used for next inner iteration in group

55
XII. If ITMPRT = 3, information will be printed out if negative values are obtained in applying the recursion relation to obtain angular fluxes at mesh cell and quadrature boundaries. This information is only printed out if the group scalar fluxes are within about a factor of 10 of convergence and for the last inner iteration in a group. The information given is a number indicating the type of angular flux that was negative (1310 for mid-mesh cell, mid-quadrature fluxes, 1340 for fluxes on the second-direction boundaries of mesh cells, 1355 for those on quadrature boundaries, 1375 for those on first-direction mesh-cell boundaries on inward passes, and 1390 for outward passes), the value that the flux was adjusted to, the value it had before being adjusted, the quadrature direction involved, the mesh interval involved (starting with 1 in the lower left corner and ending with NR times NZ for the upper right corner), the lethargy group involved, and the number of the inner iteration being performed.

XIII. The first line of the output described under X is listed for the final outer iteration performed.

XIV. If KFINISH > 0, NG values of KFING are listed indicating the status of each group. A zero means the group converged, -1 means EPGX for the group was less than one-twentieth of the previous outer iteration EPGM, 1 means that the number of group iterations had become equal to the maximum allowed. [9]

XV. If KASOR > 0, NG values of KASORG are listed indicating the number of mesh intervals in each group at which an adjustment was made in the $P_1$ source.

XVI. The mesh interval in which the greatest change in the within-group-scattering source ESLJM occurred for the last inner iteration is listed under MXVARI. [10]

XVII. This mesh interval listing is followed by the values of ESLJM which is the ratio of the actual change in the within-group-scattering source to the flux integral for the group.

XVIII. A number of labeled quantities are listed next by group with the last value being the sum over all the groups. Each quantity represents a sum over all the mesh intervals. Many of the quantities lose their meanings for an adjoint calculation.

XIX. The next set of quantities (F, XN, XJR, and XJZ) are listed by mesh intervals and have been normalized so that the maximum value of the flux will be 1. The reciprocal of the maximum value of the flux is listed first as the normalization factor.
(1) F  Production rate per unit volume
(2) XN  Scalar flux  \( \phi \) by group
(3) XJR  If KSISO = 1, the current in the first direction is given by group.
(4) XJZ  If KSISO = 1 and the problem is two dimensional, the current in the second direction is given.

XX. If KNIP = 1, flux integrals over the range of each material will be listed and punched on cards (in 7F10.6 format). (For the \( P_1 \) portion of \( P_1 \) materials, the net two-dimensional current integral is listed and punched.) If KNIP = 2, the effective buckling cross section for the transverse direction is also listed by group. This is the leakage in the transverse direction divided by the flux as integrated over all mesh intervals.

XXI. If LBUCK = -1 or 1, an effective buckling cross section for the first direction is listed and punched (in 7F10.6 format) by group. This is the leakage in the first direction divided by the flux as integrated over all mesh intervals.

XXII. If KACAV = -3, -1, 1, or 3, activity table output will be listed.

(1) If KACAV is negative, the activity table input map is listed by mesh interval.

(2) An activity table for each region specified by the activity table map (either as at XIX(1) or as at III) with negative identifications yielding successive \( P_0 \) and \( P_1 \) activity tables. Then a table for all the mesh intervals specified by the map is presented (one for \( P_0 \) and one for \( P_1 \) if KSISO = 1 and negative identifications have been included in the map.) These tables are integrals of \( \Sigma \phi V \) \( (\Sigma JV, \text{if } P_1) \) for each region listed and for each cross section \( \Sigma \) included in the input and are listed in the same format as that used for the TDSN cross sections. The final value for each group is the flux integral for the group. The final line (or set of lines) is the sum over all the groups. The flux \( \phi \) used for the scattering cross sections is that required to obtain the source.

(3) Disadvantage factors are listed by group for each region. These are the ratio of the flux integral for the region to the volume of the region divided by the ratio of the total flux integral to the total volume or
\[ f_{g,m} = \left( \frac{\sum_{i,k} \Phi_{gik} V_{ik}}{\sum_{i,k} V_{ik}} \right) \left( \frac{\sum_{i,k} \Phi_{gik} V_{ik}}{\sum_{i,k} V_{ik}} \right) \]

where \( \sum_{i,k} \) means to sum over those mesh intervals \( i,k \) contained in region \( m \).

XXIII. If \( KACAV = -3, -2, 2, \) or \( 3 \), averaging table output will be listed and punched and/or, if \( KGAVE \) is greater than zero, collapsed group output will be listed and punched (only if \( KREG = 0 \)).

1. If \( KACAV \) is negative, the averaging table input map is listed by mesh interval.

2. Averaging tables similar to the activity table described by XXII(2) are listed. These are flux-weighted tables. If \( P_0 \) and \( P_1 \) material regions are combined, the average \( \Sigma^t \) obtained will be a combination of the \( \Sigma^t \) of the \( P_1 \) region and the \( \Sigma^{tr} \) of the \( P_0 \) region. This will also occur in the final table covering all the mesh intervals in the map. If \( KACAV \) is negative, punched cards are also given for each region and for the final combined region. If \( KACAV \) is positive, punched cards are given only for the final combined region since the cross sections for the other regions will be just like the input. [13]

3. If \( KGAVE \) is greater than zero, the collapsed group identification numbers are listed.

4. If \( KGAVE \) is greater than zero, collapsed group cross sections are listed and punched for each region in the map. [13]

XXIX. If \( KEDIT \) is greater than zero, the edit output is listed.

1. Edit input is listed for each integration set.

2. For each integration set, the activity, flux integral, and flux-weighted
cross section (activity/flux integral) are listed for each specified integration limit along with the final values for the entire set.

Program Operation

[1] The binary card dump for restart consists of three dumps. The first of these is the program COMMON (ref. 14) and contains 10 cards. The second dump contains the subscripted variables required for restart except for the fluxes and currents. The number of cards depends on the problem. The third dump contains the fluxes and currents. If KSISO = 0, only fluxes will be given and the number of sets of cards will be NG. These cards may also be used for the flux guess with KFLUXI = -1. If KSISO = 1, the sets consist of the fluxes (XN), first-direction currents (XJR) and, if NZ > 1, the second-direction currents (XJZ) for each group in order. Thus, if the fluxes are desired for a flux guess they must be separated from the currents.

[2] Plane mirror reflection represents a symmetry boundary condition that does not apply to curved outer boundaries. Isotropic reflection is more realistic for many cell problems (ref. 5), and its use is recommended for curved boundaries.

For 180° rotational symmetry, the mesh below the diagonal must be such that it can undergo a rotation through 180° about the midpoint of the configuration and duplicate the mesh above the diagonal. Taking advantage of the symmetry by solving only that portion of the configuration above the midpoint requires that the mesh spacings in the first direction must be symmetric about the midpoint of the new bottom boundary.

[3] The number of mesh intervals required can be determined by a rule given in reference 2. This rule states that the ratio of the average flux for adjacent mesh intervals to the absolute difference of these fluxes should be approximately equal to the order of quadrature $n$ for important regions and groups. For instance, for $S_4$ the ratio should be approximately 4. If the ratio is much smaller than $n$, too few mesh intervals have been used, and if it is much larger than $n$, more mesh intervals than required have been used.

Another criteria has been that $\Sigma^t \Delta r$ or $\Sigma^t \Delta z$ should be less than or equal to 1. However, this criterion seems to be unnecessarily stringent and has not been required for problems calculated by the author.

The mesh layout for a two-dimensional problem is given in figure 3 that shows the mesh numbering scheme used.

[4] The materials are identified by the order in which they occur in the input with the $P_1$ matrix of $P_1$ materials counting as a separate material. Thus, the total number of materials $NMAT$ equals the number of $P_0$ materials plus twice the number of $P_1$ materials.
To identify the material to be used in a given input zone (MID), the position in which it occurs in the input is used. $P_1$ materials are identified by using the negative of the input position for MID. For example, if the fifth set of cross sections in the input is a $P_1$ material, it would be identified by -5 for MID. This would mean that the sixth set of cross sections would be the $P_1$ scattering matrix for material 5. The next allowed value of MID would be 7. If material 5 is the last material to be included in the input, NMAT becomes 6 to correspond to the number of sets of cross sections in the input.

[5] Cross sections $C$ for $S_n$ programs have usually been arranged in the following order for any one group. The first cross sections (if any are used) for the group are known as activity cross sections $\Sigma_{g}^{\text{act}}$ and are only used in the program output in the tables in flux integrations such as

$$A_{g,v} = \int \Sigma_{g,v}^{\text{act}} \Phi_{g,v} dv$$

Following the activity cross sections (or the first in order if there are no activity cross sections) comes the absorption cross section $\Sigma_{g}^{a}$. This cross section is used to compute neutron balance but is not used in the actual flux iterations.

In the next field occurs $\nu$ times the fission cross section $\nu \Sigma_{g}^{f}$ followed by the total cross section $\Sigma_{g}^{t}$ (or $\Sigma_{g}^{\text{tr}}$ if the transport approximation has been used). The field containing $\Sigma_{g}^{t}$ is designated by KCTR.

The $\Sigma_{g}^{a}$ and $\nu \Sigma_{g}^{f}$ fields are not used for the $P_1$ scattering cross sections of a $P_1$ material and neither is the $\Sigma_{g}^{t}$ field unless a buckling cross section based on the reactor dimensions ($\text{NBUCk} = -1$) is to be obtained. Then this field should contain $\Sigma_{g}^{\text{tr}}$ to be used in obtaining the buckling cross section.

The scattering cross sections in the usual $S_n$ order represent the scattering into the group rather than the scattering out of it. Positions for the same number of cross sections are allowed in each group so that there are fields provided for some nonexistent cross sections and these should contain zeros. The field containing the within-group-scattering cross section $\Sigma_{g-g}^{g}$ is designated by KCGG, while the total number of fields required for all the cross section input for any given group is given by NTYPS.

If there is any up-scattering, the number of fields allowed for each group is $\text{NUP} = \text{KCGG} - (\text{KCTR} + 1)$ with the $\Sigma_{g+1-g}$ cross section occurring in the field just before $\Sigma_{g-g}$, the $\Sigma_{g+\text{NUP}-g}$ cross section occurring just after $\Sigma_{g}^{t}$, and the others in
order between these two. For any particular group, the cross sections for the left most fields (or for all the fields) may not exist.

The down-scattering cross sections allowed for any group is NTYPS - KCGG with the \( \Sigma_{g-1} \) cross section occurring in the field immediately following \( \Sigma_g \). No down-scattering cross sections will exist for the first group and only one for the second group, etc., until all the fields are filled for group NTYPS - KCGG + 1. The entire cross section array over all groups and materials is stored in the program array \( C \).

For \( P_1 \) material, the first set of cross sections contains \( \Sigma_{g}^{act}, \Sigma_{g}^{a}, \nu\Sigma_{g}^{f}, \Sigma_{g}^{t}, \) and the total scattering array for all groups, while the second set of cross sections contains \( \Sigma_{g}^{act} \) (to be integrated over the current) and the \( P_1 \) scattering array and \( \Sigma_{g}^{tr} \) if NBUCK = -1. For KCTYP = 0, the \( 2l + 1 \) multiplier must be included in the \( P_1 \) scattering matrix. For KCTYP = 2, the \( P_1 \) scattering matrix will be multiplied by \( 2l + 1 \) by the program before being stored in the \( C \) array, and the \( 2l + 1 \) multiplier must not be supplied in the cross-section matrix.

[6] If a problem contains materials that involve \( n - 2n \) scattering, the total scattering removal cross section \( \Sigma_{g}^{r} \) must be included in the problem input as \( CN2N \) for use in computing the neutron balance. The reason is that the total \( P_0 \) scattering matrix contains the extra neutrons that result from the \( n - 2n \) scattering and would yield too large a value for neutron loss if it were obtained by summing the scattering matrix as if it represented the scattering out of a group.

The total scattering removal \( \Sigma_{g}^{r} \) is defined by

\[
\Sigma_{g}^{r} = \sum_{g'}^{G} \left( \Sigma_{g-g'}^{s0} + \Sigma_{g-g'}^{in} + \Sigma_{g-g'}^{n-2n} \right)
\]

\( g' \neq g \)

The neutron imbalance per group is computed by the program as the difference between the fixed source to the group before the inner iterations for the group and the losses from the group by absorption, scattering removal, and leakage from the system as computed after the inner iterations. The output quantity XNB(NGP1) is the ratio of the sum over all groups of the group imbalance to the "power level" XNF.

[7] If the cross sections are supplied to the program in the form of diffusion cross sections (KCTYP = 1), \( \Sigma_{g}^{tr} \) and \( \Sigma_{g-g} \) must be computed and the whole array rearranged before being stored in the \( C \) array. Both these operations are performed by TDSN: \( \Sigma_{g}^{tr} \) is calculated as \( 1/(3D_{g}) \), while \( \Sigma_{g-g} \) is computed to give balance as \( \Sigma_{g-g} = \Sigma_{g}^{tr} - \Sigma_{g}^{a} - \Sigma_{g}^{r} \), where \( \Sigma_{g}^{r} \) is the scattering removal from the group. If there is no \( n - 2n \) scattering (N2N = 0), \( \Sigma_{g}^{r} \) is obtained by summing the down- and up-scattering
out of group $g$. If there is $n - 2n$ scattering, the scattering removal cross section must be included in the input as $CN2N$ (see note [6]).

Since the cross sections are diffusion program cross sections and are rearranged to be transport approximation $S_n$ cross sections, $P_1$ scattering cannot be allowed, and $KSISO$ must be zero.

[8] If buckling dimensions are supplied, buckling-loss cross sections are computed by the program as

$$\frac{B^2}{(\Sigma^{tr}H + 2\gamma)^2} \Sigma^{tr}$$

where $B$ is the buckling factor, $H$ the buckling dimensions, and $\gamma = 0.71045608$. The buckling factor is $\pi/\sqrt{3}$ for plane boundaries and $2(2.405/\sqrt{3})$ for cylindrical boundaries. For cylindrical boundaries, $H$ is the diameter.

[9] The inner iteration convergence in the TDSN program may be seen through various quantities listed after each outer iteration and others listed after the last outer iteration.

The first of these is EPGM; the largest (considering all groups) of the quantities $\delta_2$ and $\delta_3$ given by equations (31) and (32). If this value of EPGM is larger than EPG, the problem is not converged. The amount of change in this quantity from one outer iteration to the next is an indication of the rate of convergence.

If either the top or right boundaries are reflected ($KTBC$ or $KRBC$ equal 1 or 3), the leakage across the boundary must be less than $EPG/8$ for convergence to be indicated. If either of these boundaries is not reflected, the values of the total leakage across these boundaries ($VNLG$ and $HNLG$) and of the net leakage ($XNLG$) summed over all groups is of interest in determining the amount of convergence of the boundary fluxes. Changes in these leakages as the outer iterations proceed indicate that the fluxes at the boundaries have not converged.

If up-scattering scaling is used ($KUPS > 0$), $\delta_7$ of equation (35) must be less than $EPS$ for the outer iterations to be considered converged. This quantity should decrease as convergence proceeds and is $UPS2/XNG(NGP1)$.

Other quantities that may be seen in the output to indicate convergence are $SING(NGP1)$, $SDG(NGP1)$ and $AG(NGP1)$. All three should cease to change as convergence proceeds and, if $KREG$ and $N2N$ equal zero, $SING(NGP1)$ and $SDG(NGP1)$ should be equal, within $EPS$, for convergence.

If inner iteration convergence for any group is not indicated for the last outer iteration, the inner iterations may have been terminated for either of two reasons. First, because $EPGX$ (the maximum of $\delta_2$ and $\delta_3$ of eqs. (31) and (32)) has become less than one-twentieth of the previous outer iteration $EPGM$ or second, because convergence was
not indicated within the allowed number of inner iterations ITMIGM. The quantity that indicates this lack of convergence after each outer iteration is KFINISH; the number of groups not satisfying all the convergence tests. Group convergence is required (KFINISH = 0) before the problem is considered to be converged.

[10] The outer iteration convergence in the TDSN program may be seen through the quantities listed after each outer iteration, as in note [9], and in quantities listed only after the final outer iteration.

First, the mesh interval MXVARI for which the largest change in the within-group-scattering source occurred as a result of the last inner iteration in a group is determined and listed. Then \( \delta_1 \) given by equation (30), the ratio ESIJM of this change to the total integrated group flux XNG, is found and listed. This quantity must be less than EPG/100 for group convergence to be indicated.

Another quantity listed that is concerned with the group convergence is KFING which gives by group, if KFINISH was greater than zero, the reason why the groups were not converged. Here -1 indicates that EPGX for the group was one-twentieth of the previous outer iteration EPGM, and 1 indicates that all the allowed inner iterations were performed for the group. KFING = 0 indicates convergence.

The outer iteration convergence tests are given by equations (33), (34), and (35) with \( \delta_4, \delta_5, \) and \( \delta_7 \) compared with EPS. However, a form of \( \delta_6 \) given by the ratio of the total out-scattering loss SDG to the total fission source FG for the previous outer iteration divided by the same quantity for the present outer iteration must be within EPS/RYF of 1.

If these tests are not met, the problem will continue unless stopped by either of two checks. The first of these is that the total number of inner iterations ITMI exceeds the maximum number allowed ITMIMX. The second check is that the elapsed time TIMEX required to perform the problem exceeds the maximum allowed time TIMAX. If ITMI exceeds ITMIMX, KONV is set equal to 1 and the problem terminates, or if TIMEX exceeds TIMAX, KONV is set equal to -1 and the problem terminates. In either case, all the requested output is then given along with a restart dump if requested by KBCDUP.

[11] Certain parameters of the problem can be changed on restart as listed for cards XX, XXI, and XXII of the input. They include parameters affecting the problem convergence such as the number of inner iterations to allow per group (ITMIGM), the amount of overrelaxation to use, and the use of up-scattering scaling.

It is recommended that ITMIGM be small (about 10 to 20) initially. This enables the fixed portion of the group source to be updated more often and avoids wasting time converging group fluxes while using a grossly inaccurate source guess. As the source becomes converged, the number of inner iterations allowed per group can be increased.

The relaxation parameters to be used are problem dependent. Therefore, the parameters that will yield the fastest convergence must be determined for any given class of
similar problems. In general, however, the larger the number of mesh intervals and
the less converged the fluxes, the closer the inner iteration parameter OMEGA should be
to 1. The size of the problem to which the outer iteration parameter OMEGAP is applied
is also related to the number of energy groups. Usually OMEGAP should be closer to 1
than OMEGA.

The program does have test procedures so that some convergence must be attained
before overrelaxation will be applied. Also, some degree of convergence must be main-
tained to continue the application of the relaxation parameter. However, too large a
value of the parameter will, for those iterations in which it is applied, override the con-
vergence achieved during the iterations in which it was not applied.

For negative KACCEL, a new inner iteration overrelaxation factor is computed by
the method of Wachspress (ref. 12) by the program after further convergence of the group
fluxes is attained, and this parameter is then used for the rest of the iterations in the
group. The process is repeated during each outer iteration.

The total number of inner iterations ITMI and time required TIMEX is roughly pro-
portional to the average number of neutron collisions per neutron lifetime (ref. 2). Con-
vergence will be slower in regions with longer lifetimes, and problems containing such
regions, such as problems with large voids or reflectors, may be expected to require
more inner iterations to obtain converged fluxes. These regions may have relatively
small effect on the eigenvalue, however. If this is the case, the problem will not need to
be restarted if only the eigenvalue is desired.

[12] An input zone is a region with constant mesh width and only one material. If
either the mesh width or the material changes, another input zone is required. An output
zone is a region with only one value of the output identification number. The output zone
is not dependent on mesh width, and different materials can have the same identification
number if they are to be combined in the output.

[13] For \( P_0 \) cross sections, the flux-weighted average cross sections \( \bar{\Sigma}_g \) are ob-
tained so that

\[
\bar{\Sigma}_g = \frac{\sum_{i,k} \Sigma_{gik} \Phi_{gik} V_{ik}}{\sum_{i,k} \Phi_{gik} V_{ik}}
\]

for those mesh intervals \( i,k \) over which the average is to be obtained. For the scattering
array, the average cross section \( \bar{\Sigma}_{g'g} \) is obtained by replacing \( \Sigma_{gik} \) by
\( \Sigma_{g'g, i, k} \) and \( \Phi_{gik} \) by \( \Phi_{g', i, k} \). The average scattering removal cross section is ob-
tained in the same way as $\Sigma_g$, and the within-group-scattering cross section is obtained by neutron balance so that

$$\overline{\Sigma}_{g-g} = \overline{\Sigma}_g - \overline{\Sigma}_g^{a} - \overline{\Sigma}_g^{r}$$

The cross section $\overline{\Sigma}_{g-g}$ is obtained so that $\overline{\Sigma}_g^s = \overline{\Sigma}_g^r + \overline{\Sigma}_{g-g}$ and will be incorrect for computing the scattering source if $n - 2n$ scattering is present.

Collapsed group cross sections, except for the scattering array, are obtained in the same manner as flux-weighted cross sections with the summations over the groups to be collapsed rather than the mesh intervals. The scattering array is obtained so that the in-scattering source to the collapsed group is the sum of the sources to the groups to be collapsed from the other groups. The collapsed-group-scattering removal is computed by summing the appropriate terms from the collapsed-group-scattering array and will be incorrect if $n - 2n$ scattering is present. The within-collapsed-group scattering is computed similar to that for flux-weighted cross sections by neutron balance. If $n - 2n$ scattering is present the within-collapsed-group scattering will be incorrect both because the scattering removal is incorrect and because of the balance technique.

[14] The number of directions ND for which direction cosines and weights are supplied in the input depends on the geometry of the problem and the order n (NSN) of the quadrature. For two-dimensional problems ($NZ > 0$) or for one-dimensional cylindrical geometry, $ND = NSN(NSN + 4)/4$. For one-dimensional slabs and spheres, $ND = NSN + 1$. Note here that one-dimensional slabs can be treated as two dimensional as far as quadrature is concerned by setting $NZ = 1$.

For two-dimensional problems, the ND values of the weights (W) should add to 1. The actual number of directions used by the program is twice that of ND and includes the downward directions as well as the upward directions included in the input. The weights are halved by the program to account for the extra directions.

Tables of values for two possible methods of quadrature are listed in appendix B.
APPENDIX B

DIRECTION QUADRATURE TABLES

The projection invariant sets are given in the following tables, where \( DR \) is the cosine in the \( R \)-direction, \( W \) is the weighting factor, and \( DZ \) is the cosine in the \( Z \)-direction.

**PROJECTION INVARIANT SET A (REF. 2)**

<table>
<thead>
<tr>
<th>Quadrature</th>
<th>One-dimensional set</th>
<th>Two-dimensional set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DR</td>
<td>W</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>-1.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>-.57735027</td>
<td>.5</td>
</tr>
<tr>
<td></td>
<td>.57735027</td>
<td>.5</td>
</tr>
<tr>
<td>( S_4 )</td>
<td>-1.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>-.88191710</td>
<td>.1666667</td>
</tr>
<tr>
<td></td>
<td>-.33333333</td>
<td>.33333333</td>
</tr>
<tr>
<td></td>
<td>.33333333</td>
<td>.33333333</td>
</tr>
<tr>
<td></td>
<td>.88191710</td>
<td>.1666667</td>
</tr>
<tr>
<td></td>
<td>-.33333333</td>
<td>.33333333</td>
</tr>
<tr>
<td>( S_6 )</td>
<td>-1.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>-.93094934</td>
<td>.0916667</td>
</tr>
<tr>
<td></td>
<td>-.68313005</td>
<td>.150</td>
</tr>
<tr>
<td></td>
<td>-.25819889</td>
<td>.25833333</td>
</tr>
<tr>
<td></td>
<td>.25819889</td>
<td>.25833333</td>
</tr>
<tr>
<td></td>
<td>.68313005</td>
<td>.150</td>
</tr>
<tr>
<td></td>
<td>.93094934</td>
<td>.0916667</td>
</tr>
<tr>
<td></td>
<td>-.96609178</td>
<td>.25819889</td>
</tr>
<tr>
<td></td>
<td>-.93094934</td>
<td>.25819889</td>
</tr>
<tr>
<td></td>
<td>-.68313005</td>
<td>.25819889</td>
</tr>
<tr>
<td></td>
<td>-.25819889</td>
<td>.25819889</td>
</tr>
<tr>
<td></td>
<td>.25819889</td>
<td>.25819889</td>
</tr>
<tr>
<td></td>
<td>.68313005</td>
<td>.25819889</td>
</tr>
<tr>
<td></td>
<td>.93094934</td>
<td>.25819889</td>
</tr>
<tr>
<td>Quadrature</td>
<td>One-dimensional set</td>
<td>Two-dimensional set</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td></td>
<td>DR</td>
<td>W</td>
</tr>
<tr>
<td>$S_8$</td>
<td>-1.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>-.95118972</td>
<td>.06349070</td>
</tr>
<tr>
<td></td>
<td>-.78679579</td>
<td>.091338352</td>
</tr>
<tr>
<td></td>
<td>-.57735027</td>
<td>.12676087</td>
</tr>
<tr>
<td></td>
<td>-.21821790</td>
<td>.21836491</td>
</tr>
<tr>
<td></td>
<td>.57735027</td>
<td>.12676087</td>
</tr>
<tr>
<td></td>
<td>.78679579</td>
<td>.091338352</td>
</tr>
<tr>
<td></td>
<td>.95118972</td>
<td>.06349070</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PROJECTION INARIANT SET B (REF. 2)

<table>
<thead>
<tr>
<th>Quadrature</th>
<th>One-dimensional set</th>
<th>Two-dimensional set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DR</td>
<td>W</td>
</tr>
<tr>
<td>$S_2$</td>
<td>-1.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>-.57735027</td>
<td>.5</td>
</tr>
<tr>
<td></td>
<td>.57735027</td>
<td>.5</td>
</tr>
<tr>
<td>$S_4$</td>
<td>-1.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>-.70412415</td>
<td>.33333333</td>
</tr>
<tr>
<td></td>
<td>-.09175171</td>
<td>.16666667</td>
</tr>
<tr>
<td></td>
<td>.09175171</td>
<td>.16666667</td>
</tr>
<tr>
<td></td>
<td>.70412415</td>
<td>.33333333</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-.70412415</td>
<td>.09175171</td>
</tr>
<tr>
<td></td>
<td>.70412415</td>
<td>.09175171</td>
</tr>
</tbody>
</table>
APPENDIX C

OUTLINE OF PROGRAM DECKS

The program decks required by the TDSN program as used on the Lewis Monitor System, are listed. The output of the sample problem in appendix E also gives a listing of the subroutines used by TDSN as well as subroutines provided by the Lewis Monitor System.

<table>
<thead>
<tr>
<th>Deck name</th>
<th>Language</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDSN</td>
<td>FORTRAN IV</td>
<td>To call the overlays</td>
</tr>
<tr>
<td>TIME1</td>
<td>MAP</td>
<td>To read the storage cell clock. Time returned is in clock pulses, which are $1/3600$ of a minute each. This subroutine is system oriented. Other users will need to change the program to call their own subroutine or write a dummy subroutine to return a zero when called.</td>
</tr>
</tbody>
</table>

The first overlay is as follows:

<table>
<thead>
<tr>
<th>INPUT</th>
<th>FORTRAN IV</th>
<th>To read in the nonsubscripted input and calculate the storage locations for the subscripted input</th>
</tr>
</thead>
<tbody>
<tr>
<td>READSV</td>
<td>FORTRAN IV</td>
<td>To read in the subscripted input variables</td>
</tr>
<tr>
<td>XSTDSN</td>
<td>FORTRAN IV</td>
<td>To obtain $S_n$ format cross sections from diffusion format input</td>
</tr>
<tr>
<td>SETUP</td>
<td>FORTRAN IV</td>
<td>To compute various quantities required for every iteration</td>
</tr>
<tr>
<td>BCREAD</td>
<td>MAP, Version 12</td>
<td>To read absolute binary cards (maximum of 22 words per card) from a previous problem dump</td>
</tr>
<tr>
<td>.READ5</td>
<td>MAP, Version 12</td>
<td>Used by BCREAD</td>
</tr>
</tbody>
</table>

The second overlay is as follows:

<p>| OUTER      | FORTRAN IV | Contains the outer (power) iteration loop and convergence tests.                                   |</p>
<table>
<thead>
<tr>
<th>Deck name</th>
<th>Language</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORSCAL</td>
<td>FORTRAN IV</td>
<td>To compute the correction factor to be applied to the currents and angular boundary fluxes if overrelaxation has been applied</td>
</tr>
<tr>
<td>FISION</td>
<td>FORTRAN IV</td>
<td>To compute the production neutron density and the new $\lambda$ and to renormalize to correct the power level after each outer iteration</td>
</tr>
<tr>
<td>FIXED</td>
<td>FORTRAN IV</td>
<td>To compute that portion of the source that remains fixed within the inner iterations</td>
</tr>
<tr>
<td>INNER</td>
<td>FORTRAN IV</td>
<td>Contains the inner iteration loop and convergence tests</td>
</tr>
<tr>
<td>SORAXS</td>
<td>FORTRAN IV</td>
<td>To look up the within-group-scattering cross section and the total collision-loss cross section</td>
</tr>
<tr>
<td>FLUX</td>
<td>FORTRAN IV</td>
<td>To compute the angular fluxes and the new scalar flux and current</td>
</tr>
</tbody>
</table>

The third overlay is as follows:

<table>
<thead>
<tr>
<th>Deck name</th>
<th>Language</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMPBC</td>
<td>FORTRAN IV</td>
<td>To give restart and flux and current binary card dumps</td>
</tr>
<tr>
<td>BCDUMP</td>
<td>MAP, Version 12</td>
<td>To punch data in absolute binary (maximum of 22 words per card)</td>
</tr>
<tr>
<td>.PCH.</td>
<td>MAP, Version 12</td>
<td>Used by BCDUMP</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>FORTRAN IV</td>
<td>To give the output after the final outer iteration consisting of integrated quantities and the production rate, fluxes, and currents</td>
</tr>
</tbody>
</table>

The fourth overlay is as follows:

<table>
<thead>
<tr>
<th>Deck name</th>
<th>Language</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLES</td>
<td>FORTRAN IV</td>
<td>To set up to give the activity and averaging tables</td>
</tr>
<tr>
<td>IDACAV</td>
<td>FORTRAN IV</td>
<td>To read in output identification maps to be used by TABLES</td>
</tr>
<tr>
<td>TABLE</td>
<td>FORTRAN IV</td>
<td>To call subroutines required and to list the results for the tables requested</td>
</tr>
<tr>
<td>SUM</td>
<td>FORTRAN IV</td>
<td>To compute sums of activities and fluxes over the output regions requested</td>
</tr>
<tr>
<td>Deck name</td>
<td>Language</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>AVERAG</td>
<td>FORTRAN IV</td>
<td>To compute flux-weighted average cross sections</td>
</tr>
<tr>
<td>COLAPS</td>
<td>FORTRAN IV</td>
<td>To compute flux-weighted collapsed-group cross sections</td>
</tr>
</tbody>
</table>

The fifth overlay is as follows:

<table>
<thead>
<tr>
<th>EDIT</th>
<th>FORTRAN IV</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>To compute activities, flux integrals, and average activities for a specified cross section and over specified groups and regions</td>
</tr>
</tbody>
</table>

Also, it should be noted that the Lewis Monitor System has the following off-line printer carriage control characters to be included as the first character in a format statement:

- **J**: Single space before printing rest of line.
- **K**: Double space before printing rest of line.
- **L**: Triple space before printing rest of line.
- **+**: Space suppress.
- **$**: Punch the line instead of print.
- *****: Print and punch this line.

For the first four control characters, the character must be counted as one of those being printed, but for the last two, the format is given as if the control characters were not a part of the format statement. The last control character is actually * followed by a blank to make up the complete control symbol. If these characters are not available in the operating system to be used, then other FORTRAN IV statements must be included in the program, particularly the PUNCH statement must be included if punched output is desired.
APPENDIX D

LISTING OF PROGRAM DECKS

SIBFTC TDSN DECK

PROGRAM TDSN IS A TWO DIMENSIONAL S(N) TRANSPORT CODE.

THE COMMON STATEMENTS

COMMON X
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED
COMMON /CALL3/ KBCRED
COMMON /CALL4/ KBCRED

1 KREG, KALC, KGED, KBCRED, KBB, KTBC, KLBC,
2 KBB, N5N, NG, NZ,
3 NR, NTYP, NMAT, LBUC, KCTYP,
4 KSIO, KGST, KCTR, KCGG, KDF, KFLUX,
5 KTBC, ITMIX, ITMIG, NZONER, NZONEZ,
6 KACC, KALC1, NFN, N2N, ITM,
7 KBCDUP, KACA, KGA, KNIP, KEDIT,
8 KBBC, ITMIG, GLAM, EIGEN, EPS, XNF,
9 CACAV, TIMAX, CALC1, OMEGA, OMEGA,

COMMON /CALL3/ KCHAIN, KEND, KEFN

1 HZ, HY, BF,
2 KCA, KCF,

COMMON /CALL4/ EPG, EPGX, EPXM,
2 EPG, EPGX, EPXM,
3 EPG, EPGX, EPXM,

COMMON /OUTER/ KDR, KDZ, KMR, KWHITE, KDR,
1 KNI, KNO, K180, KWHITE, KNDR,
2 KNDRUP, KNDZ, KNR, KJRR, KJZR,
3 KNZ

COMMON /CINPT1/ LKI, LKE, LFG, LKF,
1 LKAS, LAG, LFG, LSG, LSG, LBNL,
2 LSG, LSG, LSG, LSG, LSG,
3 BNS, BNNG, BNNG, BNNG, L&S,
4 LAR, LAZ, LR, LRM, LRA, LAV,
5 LST, LS, LS, LS, LST,

COMMON /CINPT2/ LMA, LV, LF, LNN,
1 LNA, LJR, LJM, LJZ, LJZ,
2 LJZ, L180, LNDR, LNR, LNDR,
3 LNZ, LNSUP, LNS, LNS, LNS,
4 LJR, LJR, LC, LNBUC,

COMMON /CINPT3/ LNDRA, LNM2A, LDM, LDMP,
1 LIDGP

COMMON /CINPT3/ LMA, LV, LF, LNN,
1 LNA, LJR, LJM, LJZ, LJZ,
2 LJZ, L180, LNDR, LNR, LNDR,
3 LNZ, LNSUP, LNS, LNS, LNS,
4 LJR, LJR, LC, LNBUC,

COMMON /CINPT3/ LAST1, LAST2,
1 LAST3, LAST4, LAST

C C C
THE DIMENSION STATEMENTS
C C C
DIMENSION X(17500)
C C C
1000 KCHAIN=1
C 1100 GO TO (1105,1205,1305,1405,1500), KCHAIN
C 1105 CALL INPUT
GO TO 1100
C 1205 CALL OUTER (X(LKI),X(LKE), X(LMA),X(LC),X(LDR),
 1 X(LFG),X(LF), X(LNOS),X(LNOSUP),X(LNZS),X(LN),X(LJR),
 2 X(LYZ), X(LV),X(LAR),X(LAZ),X(LDELZ), X(LKF),X(LKAS),
 3 X(LAG),X(LFNG),X(LSG),X(LASG),X(LSCG),X(LSDG),X(LSING),X(LXNG),
 4 X(LXNB),X(LHNLG),X(LVNLG),X(LXNLG), X(LCT),X(LCGG),X(LCGGA),
 5 X(LTVL),X(LSISO),X(LST),X(LARST),X(LAZST),X(LAZST),
 6 X(LDC), X(LBNLG),X(LBUCPS),X(LN2N), X(LNM),
 7 X(LNA),
 8 X(L180), X(LBUC),X(LNBUC),X(LMXVAR),
 9 X(LESIJM) )
C C BCDUMP OF THE PROBLEM.
C 1250 CALL DUMPBC
GO TO 1100
C 1305 CALL OUTPUT (X(LMA),
 1 X(LFG),X(LF), X(LNZS),X(LNOS),X(LNOSUP),X(LNZS),X(LN),X(LJR),
 2 X(LYZ), X(LV),X(LAR),X(LAZ),X(LDELZ), X(LKF),X(LKAS),
 3 X(LAG),X(LFNG),X(LSG),X(LASG),X(LSCG),X(LSDG),X(LSING),X(LXNG),
 4 X(LXNB),X(LHNLG),X(LVNLG),X(LXNLG), X(LCT),X(LCGG),X(LCGGA),
 5 X(LTVL),X(LSISO),X(LST),X(LARST),X(LAZST),X(LAZST),
 6 X(LDC), X(LBNLG),X(LBUCPS),X(LN2N), X(LNM),
 7 X(LNA),
 8 X(L180), X(LBUC),X(LNBUC),X(LMXVAR),
 9 X(LESIJM) )
GO TO 1100
C 1405 CALL TABLES (X(LMA),X(LC), X(LNZS), X(LN),X(LJR),X(LJZ),
 1 X(LV), X(LNMRA),X(LNMZA), X(LIDM),X(LIDMAP), X(LIDGP) )
C 1500 IF (KEDIT) 1000,1000,1505
1505 IF (KEDIT=10) 1510,1000,1000
1510 CALL EDIT (X(LMA),X(LC),X(LNZS),X(LN),X(LV) )
GO TO 1000
C C C
END
SUBROUTINE INPUT IS THE OVERLAY (OR CHAIN) SUBROUTINE OF TDSN THAT READS IN THE INPUT PARAMETERS, COMPUTES OTHER PARAMETERS, AND CALLS THE READSV, SETUP, AND FISON SUBROUTINES.

THE COMMON STATEMENTS

COMMON X
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED, KGEO, KBC, KTBC, KLBC,
1 KREG, KALC, KGO, KSN, NS, N,
2 KRBC, NTPS, NMA, LBUCK, KCTYP,
3 NR, NTRSP, LREG, LGE, LZERO,
4 KSISO, KGST, KCTR, KSTE, KREAD,
5 KACC, KGSM, ILIM, ILIM, NZERO,
6 KCDUP, KACAV, KNAS, KNIP, KEDIT,
7 KBCDU, KACAV, KNAS, KNIP, KEDIT,
8 KLBC, GLAM, EIGEN, EPS, XNF,
9 RYF, TIMAX, CALC, OMEGA, OMEGAP

COMMON /CALL3/ HZ, HY, BF, KCA, KCF,
1 KONV, KFINSH, KASOR, KUPS, NZONER, NZONEZ,
2 NBF, ND, NGP1, NZP1, NRPI,
3 NIP, NTRSP, LREG, LGE, LZERO,
4 NNXS, NAR, NC, NIJG, NBCK,
5 ITMA, ITMA, ITMIN, ITMIN, ITD,
6 ITHIMX, ITHIMX, ITNIGH, NZONER, NZONEZ,
7 KACCEL, KALC, FNL, N2N, ITNPRT,
8 KBCDUP, KACAV, KNAS, KNIP, KEDIT,
9 TIMEX, TIMEX, SDGF, SCALUP

COMMON /CALL4/ EPG, EPGX, EPGN, EPGNP, OMEGAS, OMEGPS,
1 EXTR, EXTRA, EXTRA, EXTRA, EXTRA, EXTRA,
2 KXTRA, KXTRA, KXTRA, KXTRA, KXTRA

COMMON /COUTER/ KN, KNU, K180, KWHITE, KNR,
1 KNRI, KNR, KJRR, KJZI,
2 KNOR, KNR, KJRR, KJZI,
3 KZ COMMON /CINPII/ LKI, LKE, LF, KLF,
1 LKAS, LAG, LFN, LSG, LASG, LBNL,
2 LSCG, LSOL, LSOLG, LSN, LSN, LSN,
3 LKVO, LHVOL, LHLV, LHVL, LHC, LHC,
4 LTHAT, LTHAT, LR, LHR, LRAV,
5 LTH, LTH, LSIG, LS, DST, LST,
6 LALT, LAKST, LAZST, LLUPS, LEST, LUC,
7 LON, LTOL, LEDLZ, LZ, LDR, LDZ,
8 LMR, LMR, LNI, LNO, LNZ

COMMON /CINPTZ/ LMA, L, LF, LNN,
1 LNA, LJRN, LJZJ, LJRA, LJZA, LJRR,
2 LJZJ, L180, LNOR, LNK, LNOR, LNZ,
THE DIMENSION STATEMENTS
DIMENSION X(17500)
DIMENSION TITLE(14)

THE FORMAT STATEMENTS
100 FORMAT (1H1)
101 FORMAT (1H1)
102 FORMAT (1H1)
103 FORMAT (1H1)
104 FORMAT (1H1,21HHPROGRAM STOP AT KEFN=,18)
105 FORMAT (11,1X,14A5)
106 FORMAT (2X,14A5)
107 FORMAT (1M,6x,TMBCRED=,.12)
108 FORMAT (1M,29HHPROGRAM SIZE EXCEEDED. LAST=,(8,32H WITH 17500 ST
   ORAGES AVAILABLE.)
110 FORMAT (7110)
111 FORMAT (7116)
112 FORMAT (7E10.6)
113 FORMAT (7E10.6)
114 FORMAT (2E16.8,48X,2116)
115 FORMAT (5E10.6,2110)
120 FORMAT (1H1,10X,4HKLREG,12X,4HKCALC,12X,4HKGED,12X,4HKLBC,12X,
   4HKRB,C,12X,4HKBCD,12X,4HKTB,C)
121 FORMAT (1H1,11X,3HNSN,41X,2HNG,41X,2HNR,41X,2HNZ,12X,4HNMAT,11X,
   15HNTYP$,13X,3HN2N)
122 FORMAT (1H1,9X,5HKLREG,11X,5HNSN,12X,4HKCALC,12X,4HKGED,12X,4HKLBC,12X,
   4HKRB,C,12X,4HKBCD,12X,4HKTB,C)
123 FORMAT (1H1,8X,6HTMIGM,10X,6HTMIGM,10X,6HNZONER,10X,6HNZONEZ,
   1 10X,6HFLUXI,11X,5HKALCI)
124 FORMAT (1H1,10X,6HITMPRT,10X,6HKKCALC,11X,5HKACAV,11X,5HKGAVE,
   12X,6HKNIP,11X,5HKEDIT)
125 FORMAT (1H1,10X,4HGLAM,11X,5HEIGEN,13X,3HEPS,13X,3HXN,13X,3HYF,
   1 11X,5HTIMAX,11X,5HHCALC)
126 FORMAT (1H1,10X,4HKLREG,11X,5HNSN,12X,4HKCALC,12X,4HKGED,12X,4HKLBC,12X,
   4HKRB,C,12X,4HKBCD,12X,4HKTB,C)
127 FORMAT (1H1,9X,5HKLREG,11X,5HNSN,12X,4HKCALC,12X,4HKGED,12X,4HKLBC,12X,
   4HKRB,C,12X,4HKBCD,12X,4HKTB,C)
130 FORMAT (1H1,10X,4HITMPRT,10X,6HITMIGM,10X,6HTMIGM,10X,6HNZONER,10X,6HNZONEZ,
   1 10X,6HFLUXI,11X,5HKALCI)
131 FORMAT (1H1,10X,4HGLAM,11X,5HEIGEN,13X,3HEPS,13X,3HXN,13X,3HYF,
   1 11X,5HTIMAX,11X,5HHCALC)
132 FORMAT (1H1,10X,4HKLREG,11X,5HNSN,12X,4HKCALC,12X,4HKGED,12X,4HKLBC,12X,
   4HKRB,C,12X,4HKBCD,12X,4HKTB,C)
133 FORMAT (1H1,9X,5HKLREG,11X,5HNSN,12X,4HKCALC,12X,4HKGED,12X,4HKLBC,12X,
   4HKRB,C,12X,4HKBCD,12X,4HKTB,C)
134 FORMAT (1H1,10X,4HITMPRT,10X,6HITMIGM,10X,6HTMIGM,10X,6HNZONER,10X,6HNZONEZ,
   1 10X,6HFLUXI,11X,5HKALCI)
135 FORMAT (1H1,10X,4HGLAM,11X,5HEIGEN,13X,3HEPS,13X,3HXN,13X,3HYF,
   1 11X,5HTIMAX,11X,5HHCALC)
136 FORMAT (1H1,10X,4HKLREG,11X,5HNSN,12X,4HKCALC,12X,4HKGED,12X,4HKLBC,12X,
   4HKRB,C,12X,4HKBCD,12X,4HKTB,C)
1005 READ (5,105) NTITLE, (TITLE(I), I=1,14)
WRITE (6,106) (TITLE(I), I=1,14)
IF (NTITLE) 1005,1005,1010
1010 READ (5,110) KBCRED
WRITE (6,107) KBCRED
IF (KBCRED) 1020,1020,1015
1015 ITEMP=KBCRED
CALL BCREAD (KREG,LAST)
KBCRED=ITEMP
CALL BCREAD (X11,X1(LHCDUP))
IF (KBCRED-2) 1100,1016,1016
1016 READ (5,110) ITIMIX,ITIMGM
READ (5,115) EPS,TIMAX,OMEGA,OMEGAP,TEMP,KACCEL,KUPS
IF (KBCRED-3) 1160,1019,1060
1019 READ (5,110) ITMPRT,KBCDUP,KACAV,KGAVE,KNIP,KEDIT
GO TO 1060
1020 READ (5,110) KREG,KALC,KGEO,KLBC,KRBC,KBBC,KTBC,
  1 NSN,NG,NR,NZ,NMAT,NTYPS,N2N,
  2 KCTYP,KSI0,KGST,KCTR,KCGG,NBUCK,LBLUCK,
  3 ITIMIX,ITIMGM,NZoner,NZONEZ,KOREAO,KFLUXI,KALCI

READ (5,112) GLAM,EIGEN,EPS,XNF,RYF,TIMAX,CALCI
READ (5,115) OMEGA,OMEGAP,TEMP,TEMP,KACCEL,KUPS
READ (5,110) ITMPRT,KBCDUP,KACAV,KGAVE,KNIP,KEDIT
NBF=1
IF (NBUCK) 1025,1060,1060
1025 READ (5,112) HZ,HY,BF
IF (BF-O.1E-07) 1035,1035,1040
1035 BF=1.8137994
1040 NBF=NR
IF (NZ) 1060,1060,1045
1045 GO TO (1050,1060,1060), KGE0
1050 NBF=NBF*NZ
C
1060 IF (ITIMIX) 1065,1065,1070
1065 ITIMIX=4000
1070 IF (ITIMGM) 1075,1075,1080
1075 ITIMGM=20
1080 IF (TIMAX-O.1E-07) 1085,1085,1100
1085 TIMAX=1.5
C
C WRITE INPUT PARAMETERS.
C
1100 WRITE (6,120)
WRITE (6,111) KREG,KALC,KGEO,KLBC,KRBC,KBBC,KTBC
WRITE (6,121)
WRITE (6,111) NSN,NG,NK,NZ,NMAT,NTYPS,N2N
WRITE (6,122)
WRITE (6,111) KLTYP, KSISO, KGST, KCTR, KGCS, NBUCK, LBUCK
WRITE (6,123)
WRITE (6,111) ITMIMX, ITMIGM, NZONEK, NZONEZ, KREAD, KFLUXI, KALC1
WRITE (6,125)
WRITE (6,113) GLAM, EIGEN, EPS, XNF, RYF, TIMAX, CALC1
WRITE (6,127)
WRITE (6,114) OMEGA, OMEGAP, KACCEL, KUPS
WRITE (6,124)
WRITE (6,111) ITMPRT, KBCDUP, KACAV, KGAVE, KNIP, KEDIT
IF (NBUR=1) 1125, 1125, 1105
1105 WRITE (6,126)
WRITE (6,113) HZ, HY, BF
NBUR=0
C
TO COMPUTE OTHER PARAMETERS.
C
1125 IF (IABS(KACCEL)-1) 1130, 1135, 1140
1130 OMEGA=1.0
1135 OMEGAP=1.0
1140 OMEGAS=OMEGA-1.0
OMEGAP=OMEGAP-1.0
C
1150 IF (KBCRED) 1155, 1155, 1240
1155 GO TO (1160, 1165, 1161), KGE0
1160 IF (NZ=1) 1161, 1165, 1165
1161 ND=NSN+1
GO TO 1170
1165 ND=(NSN*(NSN+4))/4
1170 NGP1=NG+1
NKPI=NP+1
NDNR=ND*NR
NDNRNG=NDNR*NG
LZSAV=0
IF (NZ=1) 1180, 1180, 1175
1175 AVF=2.0
IF (KTBC) 1181, 1181, 1177
1177 NNZS=NDKNG
LZSAV=1
GO TO 1185
1180 AVF=1.0
NZ=1
1181 NZNZS=1
1185 NIJ=NR*NZ
NZP1=NP+1
NK=NP1*NZ
NTYPS=NTYPS*NG
NDZ=ND*NZ
NDNJ=ND*NIJ
NDNZ=NDNZ*NG
NC=NTYNG*NMAT
NIJG=NIJ*NG
IF (NBUR) 1189, 1189, 1186
1186 NBUR=NMAT*NG
C
1189 KEND=0
KEFN=1189
ITMIST=0
ITMA=0
KCA=KCTR-2
KCF=KCTR-1
EPGM=0.0

C
C LOCATION OF VARIABLES IN INCLUSIVE X ARRAY.
C

1190 LKI=1
   LKE=LKI+NGP1
   LTG=LKE+NGP1
   LKF=LFG+NGP1
   LKAS=LKF+NG
   IF (KSI=0) 1191,1191,1192
1191 LASG=LKAS+1
   LJRG=LASG+1
   LJZG=LJRG+1
   LAG=LJZG+1
   GO TO 1193
1192 LASG=LKAS+NG
   LJRG=LASG+NG
   LJZG=LJRG+NG
   LAG=LJZG+NG
1193 LFG=LAG+NG
   LG=LAG+NG
   LSFG=LG+NGP
   LSG=LSFG+NG
   LSCG=LSG+NGP1
   LSDG=LSCG+NGP1
   LSN=LSDG+NGP1
   LSN=LSIGNG+NGP1
   LSN=LXNG+NGP1
   LSN=LHNLG+NGP1
   IF (NZ-1) 11186,11186,11187
11186 LVNLG=LHNLG+1
   GO TO 11188
11187 LVNLG=LHNLG+NGP1
11188 LXNLG=LVNLG+NGP1
   LBNLH=LXNLG+NGP1
   IF (NBUGK) 11189,11189,11191
11189 IF (NBK-1) 11190,11190,11191
11190 LMVAR=LBNLH+1
   GO TO 11192
11191 LMVAR=LBNLH+NGP1
11192 LBSIJM=LMVAR+NG
   LBUC=LBSIJM+NG
C
   IF (LBUC) 11193,11194,11194
11193 LDR=LBUC+1
   GO TO 11195
11194 LDR=LBUC+NG
11195 KDR=1
   LDZ=LDR+ND
   KDZ=KDR+ND
   LMR=LDZ+ND
   KMR=KDZ+ND
   LW=LMR+ND
   KW=KMR+ND
LNI = Lw + ND
KNI = Kw + ND
LNO = LNI + ND
KNO = KNI + ND

C
LNOSUP = LNO * NU
IF (NZ - 1) 1194, 1194, 1195
1194 LNZS = LNOSUP + 1
GO TO 1196
1195 LNZS = LNOSUP + NU * NING
1196 LNDS = LNZS + NXNIS
LN = LNDS + ND * NING
LJR = LR + NJG
IF (KSISO) 1197, 1197, 1199
1197 LJZ = LJR + 1
1198 LN2N = LJZ + 1
GO TO 1201
1199 LJZ = LJR + NJG
IF (NZ - 1) 1198, 1198, 1200
1200 LN2N = LJZ + NJG
LDPUS = LN2N + NMAT * NG
IF ((KCTR + 1) - KCGG) 1198, 1198, 1199
1197 KUPS = 0
LNBU = LCUPS + 1
GO TO 1199
1198 LNBU = LCUPS + NMAT * NG
C
11199 IF (NBUCK) 11200, 11200, 11201
11200 LAR = LNBU + 1
GO TO 11202
11201 LAR = LNBU + NBUCK
11202 LAZ = LAR + NRP1
LRA = LAZ + NRP1
LRAV = LRA + NRP1
LDLZ = LRAV + NR
LZ = LDLZ + NZ
C
LMA = LZ + NZP1
LV = LMA + NJ
LF = LV + NJ
LG = LF + NJ
LBCDUP = LC + NC - 1
C
LCT = LC + NC
LCGG = LCT + NR
LSISO = LCGG + NR
LST = LSISO + NJ
LDC = LST + NRP1
LM = LDC + NR
LTVL = LM + NR
LTVK = LTVL + NBF
IF (KSISO) 1202, 1202, 1204
1202    LARSO=LCGGA+1
        LARST=LARSO+1
        LAZSU=LARST+1
1203    LAZST=LAZSU+1
        LJRA=LAZST+1
        GO TO 1206
1204    LARSO=LCGGA+NR
        LARST=LARSO+NIJ
        LAZSU=LARST+NR
        IF (NZ-1) 1203,1203,1205
1205    LAZST=LAZSU+NIJ
        LJRA=LAZST+NR
C
1206    IF (KSI50) 1207,1207,1209
1207    LJKN=LJRA+1
        LJZN=LJKN+1
1208    LJZA=LJZN+1
        LNN=LJZA+1
        GO TO 1211
1209    LJRN=LJRA+NIJ
        LJZN=LJRN+NIJ
        IF (NZ-1) 1208,1208,1210
1210    LJZA=LJZN+NIJ
        LNN=LJZA+NIJ
1211    LNA=LNN+NIJ
C
    L180=LNA+NIJ
    K180=1
    IF (KKBC-4) 11211,11212,11211
11211   LWHITE=L180+1
        KWHITE=K180+1
        GO TO 11213
11212   TEMP=NSN
        TEMP=TEMP/2.0
        TEMP=TEMP+0.1
        ITEMP=TEMP
        ITEMP=NR*ITEMP
        LWHITE=1/160+ITEMP
        KWHITE=K180+ITEMP
12113   IF (KRBC-3) 1212,1213,1212
1212    LNURUP=LWHITE+1
        KNRURUP=KWHITE+1
        GO TO 1214
1213    LNURUP=LWHITE+NZ
        KNRURUP=KWHITE+NZ
1214   IF (KRBC) 1215,1215,1220
1215   IF (KTBC) 1216,1216,1220
1216    LNZR=LNURUP+1
        KNZR=KNRURUP+1
        LNDR=LNZR+1
        KNR=KNZR+1
        LNR=LNDR+1
        KNR=KNGR+1
        LJRR=LNRR+1
        KJRR=KNR+1
1217    LJZR=LJRR+1
KJZR=KJZR+1
1218 LNZ=LJZR+1
KNZ=KNZR+1
IF (NZ-1) 1218,1219,1220
1219 LAST2=LNZ+1
GO TO 1220
1220 IF (NZ-1) 1221,1222,1223
1221 LNZR=LNZR+1
KNZR=KNZR+1
1222 LNZ=LNZR+1
KNZR=KNZR+1
GO TO 1223
1223 LNZR=LNZR+NDNZ
KNZR=KNZR+NDNZ
IF (NZ-1) 1223,1224,1225
1224 LNZR=LNZR+NDNR
KNZR=KNZR+NDNR
1225 LNZ=LNZR+NDNZ
KNZR=KNZR+NDNZ
IF (NZ-1) 1225,1226,1227
1226 LNZR=LNZR+NIJ
KNZR=KNZR+NIJ
IF (NZ-1) 1227,1228,1229
1227 LAST2=LNZ+NDNR
1228 NMXZ=NZONER
IF (NZNEZ-NMXZ) 1230,1231,1232
1230 NMXZ=NZONEZ
1231 NMXZ=NMZONER
1232 LMID=LCT
LNMI=LMID+NMXZ
LNMR=LNMI+NMXZ
IF (KFLUX) 1233,1234,1235
1233 IF (NMXZON-5) 1233,1234,1235
1234 NMXZON=5
1235 LMRI=LNMR+NMXZ
LDZ1=LMRI+NMXZ
LCI=LDZ1+(NSN+2)+1
LAST1=LCI+NG+NTYPES
1237 LSTG=LASTG
LNORM=LSTG+NGP1
LAST3=LNORM+NIJ
1240 LNMRA=LCT
IF (KACAV) 1240,1241,1242
1241 IF (KGAVE) 1241,1242,1243
1242 LNMZA=LNMRA+1
LIDM=LNMZA+1
LIDMAP=LIDM+1
LIDGP=LIDMAP+1
1243 LAST4=LIDGP+1
GO TO 1250
1244 LNMZA=LNMRA+50
LIUM=LNMZA+30
LIDMAP=LIDUM+50
LIDGP=LIDUM+NIJ
IF (KGAVE) 1243, 1243, 1245
1245 LAST4=LIDGP+NG

1250 LAST=LAST1
IF (LAST-LAST2) 1251, 1252, 1252
1251 LAST=LAST2
1252 IF (LAST-LAST3) 1253, 1254, 1254
1253 LAST=LAST3
1254 IF (LAST-LAST4) 1255, 1260, 1260
1255 LAST=LAST4
1260 WRITE (6,131)
WRITE (6,111) LAST, LAST, LAST, LAST, LAST, LAST
IF (LAST-17500) 1275,1275,1261
1261 WRITE (6,108) LAST
KENO=1
KEFN=1261
GO TO 1450

1275 CALL TIME1(TIMEX)
KALCP1=KALC+1

1300 CALL READSV (I)
1 X(LKI),X(LMA),X(LC),X(LDR),X(LDZ),X(LW),X(LNOS),X(LNOSUP),
2 X(LNZS),X(LNL),X(LJR),X(LJZ),X(LBUC),X(LNBUC),X(LCUPS),X(LNZN),
3 X(LMD),X(LMMI),X(LNMR),X(LRMI),X(LR),X(LZ),X(LCI)
IF (KENO) 1350, 1350, 1305
1305 WRITE (6,104) KEFN
KEFN=1350
GO TO 1450

1350 CALL SETUP (X(LKI), X(LKE), X(LMA),
1 X(LV),X(LRAR),X(LAZ),X(ILDELZ),X(LDR),X(LDZ),X(LMR),X(LW),
2 X(LR),X(LZ),X(LRM),X(LRA),X(LRAV),X(LDZ1)
IF (KENO) 1360, 1360, 1355
1355 WRITE (6,104) KEFN
KEFN=1355
GO TO 1450
1360 IF (NZ-1) 1400, 1400, 1365
1365 LZP=LZ+NZ
IF (HZ=0.1E-07) 1366, 1366, 1400
1366 IF (KBBC) 1370, 1370, 1375
1370 HZ=X(LZP)
GO TO 140C
1375 HZ=2.0*X(LZP)

1400 KCHAIN=2
RETURN

1450 WRITE (6,104) KEFN
KCHAIN=1
RETURN

END
SUBROUTINE READSV I XKI, MA, C, DR, DZ, W, XNOS, XNOSUP, XNZS, XN, XJK, XJZ, BUCK, BUCKG, CUPS, CN2N.
2 MID, NMR, KMI, R, Z, C1.

SUBROUTINE READSV OF SUBROUTINE INPUT OF TDSN IS TO READ THE INPUT SUBSCRIPTED VARIABLES.

THE COMMON STATMENTS
COMMON /CALL1/ KCHAIN, KEND, KEFN.
1 COMMON /CALL2/ KBCRED, KBCRED, KBC, KLCBC.
2 COMMON /CALL3/ KBCRED.
3 COMMON /CALL4/ KBCRED.
4 COMMON /CALL5/ KBCRED.
5 COMMON /CALL6/ KBCRED.
6 COMMON /CALL7/ KBCRED.
7 COMMON /CALL8/ KBCRED.
8 COMMON /CALL9/ KBCRED.
9 COMMON /CALL10/ KBCRED.

THE DIMENSION STATEMENTS
DIMENSION XKI(I), MA(I), C(I), DR(I), DZ(I), W(I), XIZS(I), XNZ(I), XNOS(I), XNOSUP(I), XNZ(I), XJZ(I), BUCK(I), BUCKG(I), CUPS(I), CN2N(I), MID(I), NMR(I), KMI(I), R(I), Z(I).

THE FORMAT STATEMENTS
105 FORMAT (1P, )
110 FORMAT (711U)
112 FORMAT (7E10.6)
113 FORMAT (7E16.8)
114 FORMAT (15,E10.6,15)
115 FORMAT (116,E16.8,116)
116 FORMAT (40I3)
118 FORMAT (15,E10.6,15,E10.6,15,E10.6,15,E10.6,15,E10.6)
119 FORMAT (1HL,35HTOO MANY VALUES FOR FLUX XN READ IN )
120 FORMAT (1HL,2X,45H NM(II), RM(II), MID(II))
121 FORMAT (1HL,56HTHE FIRST DIRECTION MESH CELL BOUNDARY POSITIONS (R
1(IIK(NK))
122 FORMAT (1HL,57HTHE SECOND DIRECTION MESH CELL BOUNDARY POSITIONS (R
1(II))
123 FORMAT (1HL,28HTHE MATERIAL MAP (MAI(NZ,NR)))
124 FORMAT (1HL,30HTHE FISSION SPECTRUM (XXI(NG)))
125 FORMAT (1HL,37HTHE CROSS SECTIONS (C(INMAT,NG,NTPS)))
130 FORMAT (6F12.7)
131 FORMAT (1HL,70HTHE MATERIAL INDEPENDENT EFFECTIVE BUCKLING CROSS S
1ECTIONS (BUCK(NG)))
132 FORMAT (1HL,73HTHE MATERIAI DEPENDENT EFFECTIVE BUCKLING CROSS SEC
1TIONS (BUCKG(INMAT,NG)))
134 FORMAT (1HL,57HTHE OUT SCATTERING REMOVAL CROSS SECTIONS (CNZ(NM
1T,NG)))
135 FORMAT (6X,2HN=,15)

THE INTEGER FUNCTIONS
INDEXLENGTH, INDEXO, INDEXL=LENGTH*(INDEXO-1)+INDEXL

1000 IF (KBCRED) 1001,1001,1040
1001 WRITE (6,120)
   DO 1005 I=1,NZURER
      READ (5,114) NMR(I), RMI(I), MID(I)
      WRITE (6,115) NMR(I), RMI(I), MID(I)

1010 I=1
   J=1
   R(I)=0.0
   DELR=RMI(I)
   GO TO 1020
1015 DELR=RMI(I)-RMI(I-1)
1020 VMI=NMR(I)
   DELR=DELR/VMI
   I2=0
1025 I2=I2+1
1030 IF (I2=NMR(I)) 1030,1030,1035
1030 I1M1=I1
11=I1+1
   R(I1)=R(I1)+DELR
   MAX(I1)=MID(II)
   GO TO 1025
1035 I=I+1
   IF (I-NZURER) 1015,1015,1040
1040 WRITE (6,121)
      WRITE (6,113) (KI(I), I=1,NRPI)

1050 IF (NZ-1) 1180,1180,1055

83
1055 IF (KBCRED) 1060, 1060, 1140
1060 IF (NZONEZ-1) 1065, 1065, 1100
1065 I=NR
   DO 1070 J=2, NZ
   DO 1070 I=1, NR
   I=I+1
1070 MA(I)=MA(I)
C
1100 WRITE (6, 120)
   DO 1105 J=1, NZONEZ
   READ (5, 114) NMI(J), RMI(J)
1105 WRITE (6, 115) NMI(J), RMI(J)
C
1110 J=1
   I=1
   Z(I)=0.0
   DELR=RMI(J)
   GO TO 1120
1115 DELR=RMI(J)-RMI(J-1)
1120 VMI=NMI(J)
   GELR=0.0
   I2=0
1125 I2=I2+1
   IF (I2-NMI(J)) 1130, 1130, 1135
1130 I1M1=I1
   I1=I1+1
   Z(I)=Z(I1M1)+DELR
   GO TO 1125
1135 J=J+1
   IF (J-NZONEZ) 1115, 1115, 1140
1140 WRITE (6, 122)
   WRITE (6, 113) (Z(J), J=1, NZP1)
C
1150 IF (KBCRED) 1155, 1155, 1180
1155 IF (NZONEZ-1) 1180, 1180, 1160
1160 K=0
   J=0
   DO 1175 JJ=1, NZONEZ
      READ (5, 11C) (MID(I), I=1, NZONER)
   K=K+NMI(JJ)
1170 J=J+1
   I=0
   DO 1170 J=1, NZONER
         L=NMR(I)
      DO 1170 KK=1, L
         I=I+1
   1J=INDEX(NR, J, I)
1170 MA(J)=MID(I)
   IF (J-K) 1165, 1175, 1175
1175 CONTINUE
C
1180 WRITE (6, 123)
   DO 1185 J=1, NZ
   1J=INDEX(NR, J, I)
K=INDEX(NR,J,NR)
1185 WRITE (6,116) (MA(I), I=IJ,K)
C
C
1200 IF (KBCRED) 1205,1205,1210
1205 READ (5,112) (XKI(IG), IG=1,NG)
1210 WRITE (6,124)
WRITE (6,113) (XKI(IG), IG=1,NG)
C
1250 IF (KBCRED) 1255,1255,1375
1255 IF (KDREAD) 1300,1300,1260
1260 READ (5,112) (DR(M), M=1,ND)
IF (NZ-1) 1270,1270,1265
1265 READ (5,112) (DZ(M), M=1,ND)
1270 READ (5,112) (W(M), M=1,ND)
C
C
1300 IF (KCTYP-1) 1375,1305,1355
C
1305 WRITE (6,125)
NTYPS1=NTYPS-1
KUP=KCGG-(KCTR+1)
DO 1350 N=1,NMAT
WRITE (6,135) N
L=0
DD 1310 IG=1,NG
DD 1310 K=1,NTYPS
L=L+1
1310 C1(L)=0.0
C
DD 1325 IG=1,NG
K1=INDEX(NTYPS,IG,1)
K2=INDEX(NTYPS,IG,4)
K3=INDEX(NTYPS,IG,NTYPS)
READ (5,112) (CI(K), K=K1,K2,K3,TEMP,C1(K3)
WRITE (6,113) (CI(K), K=K1,K2),TEMP,C1(K3)

IF (NBUCK) 1316,1316,1315
1315 K1=INDEX(NG,N,IG)
BUCK(K1)=TEMP
1316 K1=INDEX(NTYPS,IG,3)
K2=INDEX(NTYPS,IG,KCGG)
TEMP=C1(K1+1)
IF (KCGG+1)-NTYPS) 1319,1317,1317
1317 IF (KUP) 1318,1318,1319
1318 C1(K2)=C1(K1)
GO TO 1322
1319 IF (N2N) 1321,1321,1320
1320 K3=INDEX(NG,N,IG)
CN2N(K3)=C1(K1)
1321 C1(K2)=0.0
1322 C1(K1)=TEMP
IF (KUP) 1325,1325,1323
1323 C1(K1+1)=0.0
CONTINUE
C IF (KCGG-1-NTPS) 1330, 1340, 1340
1330 WRITE (*, 105)
   IGEND=NG
   NGM1=NG-1
   DU 1335  IG=1,NGM1
   IGEND=IGEND-1
   IF (IGEND=(NTyps-KCGG)) 1331,1331,1332
1331 K4=KCGG+IGEND-1
   GO TO 1333
1332 K4=NTYP$1
1333 K1=INDEX(NTyps,IG,KCGG)
   K2=INDEX(NTyps,IG,K4)
   READ (5,112) (C1(KI), K=K1,K2)
1335 WRITE (6,113) (C1(KI), K=K1,K2)
C
1340 IF (KUP) 1345, 1345, 1341
1341 WRITE (*, 105)
   IG江北=NG+1-KUP
   K3=KCTR+1
   DU 1342  IG=1G1,NG
   K4=KCGG+IG-NG+1)
   K1=INDEX(NTyps,IG,K3)
   K2=INDEX(NTyps,IG,K4)
   READ (5,112) (C1(KI), K=K1,K2)
   WRITE (6,113) (C1(KI), K=K1,K2)
1342 CONTINUE
C 1345 CALL XSTDSN ( N, KUP, MA, C1, CN2N, C )
1350 WRITE (*, 105)
   GO TO 1375
C
1355 DU 1365 N=1,NMAT
   K1=INDEX(NTyNg, N, 1)
   K2=INDEX(NTyNg, N, NTyNg)
   READ (5,130) (C(KI), K=K1,K2)
   IF (KSISU) 1360, 1360, 1356
1356 DU 1357 IJ=1,NIJ
   IF ((MA(IJ)-I)*N) 1357, 1358, 1357
1357 CONTINUE
   GO TO 1360
1358 DU 1359 IG=I+NG
   K3=INDEX(NTyps,IG,KCTR)
   K3=K3+1
   K3=INDEX(NTyNg, N, K3)
   K4=INDEX(NTyps, IG, NTyps)
   K4=INDEX(NTyNg, N, K4)
   DU 1359 K=K3,K4
1359 CI(K)=3.0*C(K)
1360 IF (N2-N) 1365, 1365, 1361
1361 K1=INDEX(NG, N, 1)
   K2=INDEX(NG, N, NG)
   RIAD (5,112) (CN2N(K), K=K1,K2)
CONTINUE

WRITE (6, 125)
DO 1390 N = 1, NMAX
WRITE (6, 135) N
DO 1380 IG = 1, NG
K3 = INDEX(NTRYPS, IG, 1)
K4 = INDEX(NTRYPS, IG + 1, NTYPES)
K1 = INDEX(NTRYPS, N, K3)
K2 = INDEX(NTRYNG, N, K4)
IF (KBCRED) 1376, 1376, 1380
DO 1377 IC(K1, K2) = IC(K1, K2)
WRITE (6, 113) IC(K1, K2)
CONTINUE

READ (5, 112) (C(K), K = K1, K2)
WRITE (6, 116) IC(5), IC(K1, K2)
CONTINUE

READ (5, 112) (CN2N(K), K = K1, K2)
WRITE (6, 105) IC(5), IC(K1, K2)
CONTINUE

CALCULATING THE OUT SCATTERING (CN2N)

IF (KBCRED) 1400, 1400, 1440
DO 1401 IC(K1) = IC(K1) + IC(K2)
CONTINUE

READ (5, 112) (CN2N(K), K = K1, K2)
WRITE (6, 105) IC(5), IC(K1, K2)
CONTINUE

CALCULATING SIG UP, OUT SCATTERING (CN2N)

IF (KBCRED) 1426, 1426, 1450
CONTINUE

1425 IF (IC(K1) = IC(K2)) 1426, 1426, 1450
CONTINUE
1428 IF (IGG-NG) 1425,1429,1435
1429 K1=INDEX(NTYP5,IGG,N)
GO TO 1431
1430 K1=INDEX(NTYP5,IG,N)
1431 K2=INDEX(NTYNG,M,K1)
LUPS(K)=CUPS(K)+C(K2)
1435 CONTINUE
C
C
1440 IF (KUPS) 1450,1450,1441
1441 DO 1445 IG=1,NG
DD 1445 M=1,NMAT
K=INDEX(NG,M,IG)
IF (ABS(CUPS(K) - 0.1E7) 1443,1443,1442
1442 KUPS=IG
GO TO 1450
1443 CONTINUE
1445 CONTINUE
C
C
1450 IF (LBUCK) 1455,1455,1451
1451 IF (KBCKED) 1452,1452,1453
1452 READ (5,112) (BUCLK(IG), IG=1,NG)
1453 WRITE (6,131)
WRITE (6,131) (BUCLK(IG), IG=1,NG)
C
C
1455 IF (NBUCK) 1465,1465,1456
1456 IF (KBCKED) 1457,1457,1460
1457 IF (KCTYP=1) 1458,1460,1458
1458 DO 1459 N=1,NMAT
K1=INDEX(NG,N,1)
K2=INDEX(NG,N,NG)
1459 READ (5,112) (BUCKG(K), K=K1,K2)
1460 WRITE (6,132)
DO 1461 N=1,NMAT
K1=INDEX(NG,N,1)
WRITE (6,135) N
K2=INDEX(NG,N,NG)
WRITE (6,135) (BUCKG(K), K=K1,K2)
1461 WRITE (6,105)
C
C
1465 WRITE (6,134)
DO 1466 N=1,NMAT
WRITE (6,135) N
K1=INDEX(NG,N,1)
K2=INDEX(NG,N,NG)
1466 WRITE (6,113) (CN2N(K), K=K1,K2)
C
C
C
1500 LOS=0
LZS=0
IJGS=0
DO 1590 IG=1,NG
1590 IJGST=IJGS+1
IJGSP=IJGS+NIJ
IF (KBCRED) 1505,1505,1525
1505 IF (KFLUX1) 1525,1506,1530
1506 DO 1507 IG=1IJGST,1JGSP
1507 XN(IG)=1.0
IF (KSI50) 1560,1560,1510

C
1510 IF (KBCRED) 1511,1511,1513
1511 DO 1512 IG=1IJGST,1JGSP
1512 XJR(IG)=1.0
GO TO 1515
1513 CALL BCREAD (XJR(IGST),XJR(IGSP))
1515 IF (NZ-1) 1560,1560,1516
1516 IF (KBCRED) 1517,1517,1519
1517 DO 1518 IG=1IJGST,1JGSP
1518 XJZ(IG)=1.0
GO TO 1560
1519 CALL BCREAD (XJZ(IGST),XJZ(IGSP))
GO TO 1560

C
1525 CALL BCREAD (XN(IGST),XN(IGSP))
IF (KSI50) 1560,1560,1510

C
1530 L=0
1535 READ (5,118) (NMRI(I),RMII(I)), I=1,5
1540 IF (NMRI(I)) 1550,1550,1541
1541 LI=L+1
L=L+NMRI(I)
IF (L-NI) 1543,1543,1542
1542 WRITE (6,119)
KEND=1
GO TO 1600
1543 TEMPI=L+IJGS
TEMP2=L+IJGS
DO 1544 1JG=TEMPI,TEMP2
1544 XN(1JG)=RMII(I)
IF (I-5) 1545,1545,1535
1545 I=I+1
GO TO 1540
1550 IF (L-NI) 1551,1551,1555
1551 TEMPI=TEMP2+1
DO 1552 1JG=TEMPI,1JGSP
1552 XN(1JG)=0.0
1555 IF (KSI50) 1560,1560,1510

C
1560 IF (KBCRED) 1561,1561,1590
1561 IF (KRBG-1) 1562,1562,1566
1562 DO 1563 I=1,NDNZNG
1563 XNOSOI=0.0
IF (NZ-1) 1575,1575,1564
1564 DO 1565 I=1,NDNZNG
1565 XNOSUP(I)=0.0
GU TO 1575
1566 DO 1570 J=1,NZ
    IJG=INDEX(NZ,IG,J)
1567    IJG=IJG*NR
    DO 1570 M=1,ND
    LOS+LOS+1
    XNOS(LOS)=XN(IJG)
1568    IF (NZ-I) 1570,1570,1567
1569    XNOSUP(LOS)=XN(IJG)
1570 CONTINUE
C
1575 IF (KTBC-I) 1576,1578,1578
1576 DO 1577 I=1,NXNZS
1577  XNZS(I)=0.0
    GO TO 1540
1578 IF (NZ-I) 1590,1590,1579
1579    IJG=INDEX(NR,IG,0)
    DO 1580 I=1,NR
    IJG=IJG+1
    DO 1580 M=1,ND
    LZS=LZS+1
1580    XNZS(LZS)=XN(IJG)
C
1590 IJGS=IJGSP
C
C
C
1600 RETURN
C
END
SUBROUTINE XSTOSN ( N, KUP, MA, CI, CN2N, C )

SUBROUTINE TDSNXS IS TO OBTAIN TDSN CROSS SECTIONS FROM RP-2 FORMAT CROSS SECTIONS.

THE COMMON STATEMENTS
COMMON /CALL1/ KCHAIN, KEND+, KEFN
COMMON /CALL2/ KBCRED, KBBC, KTBC, KLBC,
1 KREG+, KALC, KGBD, NSN, NG, NZ,
2 KRBC, NTYPS, NMAT, LBUCK, KCIYP,
3 NR, KGST, KCTR, KCGG, KREAD, KFLUXI,
4 KSISO, ITMIXX, ITMIGX, NZONE, NZONEX
5 KACCEL, KACAV, KAVE, KNI, KEV, KEDIT,
6 KBCDUP, KNS, NFX, XNS,
7 COMMON /CALL3/ HZ, HY, BF,
1 COMMON /CALL3/ HZ, HY, BF,
2 KCA, KCF,
3 KCONV, KFINISH, KASOR, KUPS,
4 NBF, NO, NPI, NIPL, NRP1,
5 NIJ, NITYN, NDN, NDNI, NDNIJ, NDNZNG, NDNZ
6 NaNzs, NAR, NC, NJG, NBUCK,
7 ITMA, ITMI, ITMIST, ITMIGX, ITD,
8 TIMEX, TIMEZ, NEGSOR, LSAV,
9 COMMON /CALL4/ EPG, EPGX, EPX, SCALUP,
1 DIMENSION MA(I), CN2N(I), C(I)
2 DIMENSION MA(I), CN2N(I), C(I)

INDEX[L,INDEXO,INDEXL]=LENGTH*INDEXD-1+INDEXL

1000 KCTRPI=KCTR+1
KUPI=KUP+1
NTYPSI=NTYPS-1
KSI=0
IF (KSIISC) 1010,1110,1001
1001 DO 1005 JJ=1,NIJ
IF (MA(IJ)) 1102,1003,1002
1002 IF ((MA(IJ)-1)+N) 1005,1004,1005
1003 KSI=-1
     GO TO 101C
1004 KSI=1
     GO TO 101G
1005 CONTINUE

C

1010 DO 1100 IG=1,NG
     L=INDEX(NTYPS,IG,1)
     K1=INDEX(INTYNG,N1,L)
     K1=K1-1
     IF (KSI) 1011,1011,1013
1011 K3=K1+1
     K4=K1+NTYPS
     DO 1012 K=K3,K4
1012 C(K)=0.0
     GO TO 1020
1013 K3=K1+1
     K4=K1+KCTR
     DO 1014 K=K3,K4
1014 C(K)=0.0
     K3=K4+2
     K4=K1+NTYPS
     DO 1015 K=K3,K4
1015 C(K)=0.0

C

1020 K=K1+KCTR
     IF (KSI) 1022,1021,1025
1021 C(K)=1.0/(3.0*C1(L))
     GO TO 1023
1022 K3=1+NTYPS1
     C(K)=C1(K3)
     N1=N+1
     K3=INDEX(INTYNG,N1,L)
     K3=K3*KCTR-1
     C(K3)=1.0/(3.0*C1(L))
1023 C(K-1)=C1(L+2)
     C(K-2)=C1(L+1)
     GO TO 1030
1025 K2=K1+KCG6
     C(K2)=C1(L)

C

1030 K3=KCTR-3
     IF (K3) 1040,1040,1031
1031 K2=K-3
1035 C(K2)=0.0
     K2=K2-1
     K3=K3-1
     IF (K3) 1040,1040,1035

C

1040 K2=K1+KCG6
     IF (KSI) 1041,1041,1050
1041 C(K2)=C(K)-C1(L+1)
     IF (N2N) 1043,1043,1042
1042 K4=INDEX(NG,N+IG)
     C(K2)=C(K2)-CN2N(K4)
GO TO 1050
10+3 DD 1045 K3=KCTR,1,NTYPS1
K4=L+K3-1
1045 C(K2)=C(K2)-C(K4)
C
1050 IF (IG-NG) 1055,1075,1075
1055 IF (IG+KUP-NG) 16/9,1060,1060
1060 K=K2
K3=KCTR
IG1=IG
L=0
KUP1=KUP1-1
L1=0
1065 LI=LI+1
/ IF (LI-KUP1) 1070,1070,1075
1070 K=K-1
K3=K3+1
IG1=IG1+1
L=L+1
K4=INDEX(NTYPS,IG1,K3)
C(K)=C(K4)
IF (L-KUP1) 1065,1075,1075
C
1075 IF (IG-1) 1100,1100,1080
1080 L=1
K3=KCGG-1
IG1=IG
1085 K2=K2+1
K3=K3+1
IG1=IG1-1
L=L+1
K4=INDEX(NTYPS,IG1,K3)
C(K2)=C(K4)
IF (IG-L) 1100,1100,1090
1090 IF ((K2-K1)-NTYPS) 1085,1100,1100
C
C
1100 CONTINUE
C
C
RETURN
C
END
**SUBROUTINE SETUP**

SUBROUTINE SETUP (XXI, XKE, MA, V, AR, AZ, DELZ, DR, DZ)

THE COMMON STATEMENTS

COMMON /CALL1/ KCHAIN, KEND, KEFN

COMMON /CALL2/ KBCRED, KHBC, KTBC, KLBC,

1 KREG, KALC, KBCRED, KHBC, KTBC, KLBC,

2 KREG, KREG, NSN, NG, N2,

3 NK, NK, NTYPES, NMAT, LBUCK, KCTYP,

4 KSISO, KGST, KCTK, KCGG, KOREAD, KFLUXI,

5 KACC, KALC1, NFN, NZN, KEDIT,

6 KBCRUP, KALC, KGAVE, KNUM, KEDIT,

7 KBCRUP, KALC, KGAVE, KNUM, KEDIT,

8 RYF, TIMAX, CALCL, OMEGA, OMEGAP

COMMON /CALL3/

1 MZ, HY, BF,

2 KAK, KLA, KCF,

3 KONV, KFINSH, KASOR, KUPS,

4 NBF, NO, NBP, NZP, NBP,

5 NIJ, NTYPES, NDNZ, NUNI,

6 NXNZS, NK, NC, NIJ, NBUCK,

7 ITM, ITMI, ITMIST, ITMIGX, ITD,

8 TIMEX, TIMEZ, OMEGAS, OMEGAP,

9 TIMEX, TIMEZ, GSOFIG, SCALUP

COMMON /CALL4/ EPG, EPGX, EPGM, EPGL,

1 EPG, EPG, EPGL, OMEGAS, OMEGAP,

2 EXTR, EXTR, EXTR, EXTR, EXTR,

3 KXTR, KXTRA, KXTRA, KXTRA, KXTRA

COMMON /CGSETUP/ KALCP1, KALCP1

1 EXTRA

THE DIMENSION STATEMENTS

DIMENSION XXI(11), XKE(I), MA(I)

DIMENSION V(I), AR(I), AZ(I),

1 DELZ(I), DR(I), OZ(I),

2 MR(I), WI(I)

DIMENSION MA(I), Z(I), RM(I), RA(I)

1 OZ(I), RAM(I), OZ(I)

THE FORMAT STATEMENTS

111 FORMAT (1116)

113 FORMAT (7E16.8)

122 FORMAT (1F8.0,22HDIIRECTIONAL FUNCTIONS )

130 FORMAT (1F8.0,13)
141 FORMAT (1HL, 32H THE MESH CELL VOLUMES (VINZ, NR)))
142 FORMAT (1HL, 16, 37H NEGATIVE VOLUMES HAVE BEEN COMPUTED.)
C
C THE INTEGER FUNCTIONS
INDEXI(LENGTH, INDEXO, INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C COMPUTE INITIAL VALUES.
C
1000 PI=3.14159265
GO TO (1005, 1010, 1015), KGE0
1005 AF=1.0
VF=1.0
GO TO 1020
1010 AF=2.0*PI
VF=0.5
GO TO 1020
1015 AF=4.0*PI
VF=1.0/3.0
1020 SDFGL=0.0
C
C IF (ITMIST) 1025, 1025, 1200
1025 GLAMP=0.0
EXTRP=0.0
C
C COMPUTE 5A CONSTANTS.
C
1050 IF (KBCRED) 1051, 1051, 1200
1051 IF (KDDREAD) 1055, 1055, 1052
1052 DO 1053 M=1, NO
1053 W(M)=W(M)/AVF
GO TO 1150
1055 IF (NO-(NSN+1)) 1125, 1125, 1060
C
C CODED FOR CYLINDERS AND TWO DIMENSIONAL SLABS.
C
1060 NSND2=NSN/2
DD 1065 M=1, NSND2
VM=M
1065 AREA=AREA+VM
M=0
TEMP=0.0
DZ1(M+1)=1.0
1070 M=M+1
IF (M-NSND2) 1075, 1080, 1080
1075 VM=M
TEMP=TEMP+VM
DZ1(M+1)=1.0-(TEMP/AREA)
GO TO 1070
1080 DZ1(M+1)=0.0
C
SN=NSN-(NSN+2)
DZF = SQRT(SN/(SN - 2.0))
DZF = DZF / 2.0
M = 0
M1 = 0
1100 M = M + 1
   IF (M - NSND2) 1105, 1105, 1150
1105 L = 0
   MT2 = 2 * M
   VMT4 = 4 * M
   M1 = M + 1
   DZ2 = DZF * (DZ1(M + 1) * DZ1(M))
   D1 = SQRT(1.0 - (DZ2 * DZ2))
   DZ(M1) = DZ2
   DR(M1) = -D1
   W(M1) = 0.0
   IF (M - NSND2) 1115, 1115, 1100
1110 L = L + 1
   IF (L - MT2) 1115, 1115, 1100
1115 M1 = M1 + 1
   DZ(M1) = DZ2
   VLT2 = (2 * L) - 1
   PHI = PI * (1.0 - (VLT2 / VMT4))
   COSPHI = COS(PHI)
   DR(M1) = D1 * COSPHI
   W(M1) = (1.0 / 2.0) / AVF
   GO TO 1110

C CODED FOR SPHERES AND ONE DIMENSIONAL SLABS.
C
1125 NSND2 = NSN / 2
   SN = NSN
   AREA = 2.0 / SN
   M = 0
   DZ1(M + 1) = 1.0
1130 M = M + 1
   IF (M - NSND2) 1135, 1140, 1140
1135 DZ1(M + 1) = DZ1(M) - AREA
   GO TO 1130
1140 DZ1(M + 1) = G.C

C
   DZF = SN / SQRT((SN**2) - 1.0)
   DZF = DZF / 2.0
   M = 1
   DR(M) = -DZ1(M)
   W(M) = 0.0
1141 IF (M - NSND2) 1142, 1142, 1143
1142 M = M + 1
   DR(M) = -DZF * (UZ1(M) + DZ1(M - 1))
   W(M) = (1.0 / SN) / AVF
   GO TO 1141
1143 L = 0
1144 L = L + 1
   IF (L - NSND2) 1145, 1145, 1150
1145 M = M + 1
   M1 = (NSND2 + 2) - L
   DR(M) = -DR(M)
   W(M) = W(M1)
GO TO 1144

C

1150 MI=ND
1155 WD=AVF+W(MI)*DR(MI)
   IF (ABS(WD)-0.1E-07) 1160,1160,1156
1156 IF (WD) 1170,1160,1165
1160 MR(MI)=MR(MI+1)
1165 MI=MI-1
   IF (MI) 120C,1200,1155
1170 M=M+1
1175 M=M+1
   IF (1.0E-04-ABS(DR(M)+DR(MI))) 1175,1180,1180
1180 MR(M)=M
   MR(MI)=M
   GO TO 1165
C

C

1200 IF (ITMA) 1305,1305,1365
1305 IF (KAC1-1) 1325,1310,1325
1310 DO 1315 IG=1,NG
1315 XKXI(IG)=XKXI(IG)/CALC1
1325 XKXI(NGP1)=-0.0
   DO 1330 IG=1,NG
   XKXI(INGP1)=XKXI(INGP1)+XKXI(IG)
1330 XKXI(INGP1)=XKXI(INGP1)
   XKE(NGP1)=XKXI(INGP1)
   XKE(INGP1)=XKXI(INGP1)
C

C

1350 DO 1360 I=1,NRP1
1360 RA[I]=R[I]
C

C

1365 GO TO (1400,140C,1400,1400,1370,1385), KALCPI
1370 DO 1375 I=1,MR
   K=MA[I]
   RA[I+1]=RA[I]+(R[I+1]-R[I])*(1.0+EIGEN*RMIK))
   IF (1.0+EIGEN*RMIK) 1380,1375,1375
1375 CONTINUE
   GO TO 1400
1380 KEFN=1370
   KEND=1
   GO TO 1600
1385 DO 1390 I=1,NRP1
1390 RA[I]=EIGEN*R[I]/R[NRP1]
C

C

1400 IF (ITMA) 1410,1410,1405
1405 GO TO (1500,150C,1500,1500,1410,1410), KALCPI
1410 GO TO (1411,1415,1450), KGE0
1411 IF (NZ-1) 1450,1450,1415
1415 DO 1420 I=1,NR
   IP1=I+1
   RAV(I)=0.5*(RA(IP1)+RA(I))
   GO TO (1416,1417,1420), KGE0
AR(I)=AF
AZ(I)=RA(IP1)-RA(I)
GO TO 1420
AR(I)=AF*RA(I)
AZ(I)=AF*RA(I)^2*(RA(IP1)-RA(I))
CONTINUE
GO TO (1421,1422,1450), KGEO
AR(NRP1)=AF
GO TO 1423
AK(NRP1)=AF*RA(NRP1)
J=1
DELZ(J)=1.0
IF (NZ-J) 1430,1430,1425
JP1=J+1
DELZ(J)=Z(JP1)-Z(J)
ITEMP=0
DO 1440 I=1,NR
J1=INDEX(NR,J1)
V(IJ)=AZ(I)*DELZ(J)
IF (V(IJ)) 1435,1440,1440
ITEMP=ITEMP+1
CONTINUE
IF (NZ-J) 1475,1475,1445
J=JP1
GO TO 1425
C
DO 1465 I=1,NRP1
GO TO (1455,1465,1460), KGEO
AR(I)=AF
GO TO 1465
AR(I)=AF*(RA(I))^2
CONTINUE
ITEMP=0
DO 1470 I=1,NR
RAV(I)=0.5*(RA(I)+RA(I+1))
V(I)=VF*(AR(I)+RA(I+1))-AR(I)*RA(I)
IF (V(I)) 1469,1470,1470
ITEMP=ITEMP+1
CONTINUE
DELZ(I)=1.0
C
IF (ITEMP) 1500,1500,1480
WRITE (6,142) ITEMP
KEFN=ITEMP
KEND=1
C
PRINT SOME SETUP QUANTITIES.
C
1500 IF (ITMA) 1505,1505,1501
1501 IF (KBCRED) 1575,1575,1505
1505 WRITE (6,141)
DO 1510 J=1,NZ
WRITE (6,130) J
K1=INDEX(NR,J1)
K2=INDEX(NR,J2,NR)
WRITE (6,113) (V(IJ), IJ=K1,K2)
WRITE (6,122)
WRITE (6,113) (DR(M), M=1,ND)
IF (NZ-1) 1520,1520,1515
1515 WRITE (6,113) (DZ(M), M=1,ND)
1520 WRITE (6,113) (H(M), M=1,ND)
WRITE (6,111) (MR(M), M=1,ND)
C
C
1550 IF (KBCRED) 1575,1575,1555
1555 IF (NFN-1) 1600,1560,1600
1560 DO 1565 IG=1,NGP1
1565 XKE(IG)=XXII(IG)
GO TO 1600
C
C
1575 NFN=1
C
C
1600 RETURN
C
END

$EMAP BCREAD 60,M94,DECK

7094 RELMOD ASSEMBLY.

$IBLDR BCREAD

$TEXT BCREAD
BCREAD SUBROUTINE FOR IBSYS

ENTRY BCREAD

BINARY CARD ID. BCREAD00
00000 00000 0 00000 10001 BCREAD SAVE 1,2,4
00001 0774 00 2 00000 10000
00002 0774 00 1 00000 10000
00003 0774 00 4 00000 10000
00004 0620 00 4 00001 10000
00005 0634 00 4 06000 10011
00006 0634 00 4 00110 10001
00007 0634 00 4 00003 10001
00010 0634 00 1 00002 10001
00011 0634 00 2 00001 10000
00012 0500 00 4 00003 10000
00013 0560 00 4 00004 10000
00014 0040 00 C 01002 10011
00015 0131 00 6 00000 10000
00016 4600 00 0 0107 10001
00017 0620 00 0 0107 10001
00020 0734 00 1 00000 10000
00021 06001 1 01001 10011
00022 0534 00 2 00107 10001

BCREAD SUBROUTINE FOR IBSYS

ENTRY BCREAD

BINARY CARD ID. BCREAD01
00023 0634 00 1 00037 10001 SXA IX1,1
00024 0634 00 2 00040 10001 SXA IX2,2
00025 05CC 60 0 00105 10001 CLA* IN5
00026 0621 00 0 01002 10011 STA **2
00027 GC74 0C 4 07000 10011 TXS .CLOSE,4
00030 5 00000 0 00000 10000 MON **
00031 0560 60 0 00106 10001 CLA* READ5
00032 0621 00 0 00036 10001 STA MON
00033 0621 00 0 00044 10001 STA READ2
00034 0621 00 0 00065 10001 STA SHUT
00035 0674 00 4 10000 10011 TXS .OPEN,4
00036 5 00000 0 00000 10000 MON MON **
00037 0774 00 1 00000 10000 IX1 AXT **,1
00040 0774 00 2 00000 10000 IX2 AXT **,2
00041 0634 00 2 00047 10001 SXA SXA 10,2
00042 7 00226 1 00053 10001 TXL LASTS,1,22
00043 0074 00 4 11000 10011 READ TXS .READ,4
00044 0 00067 C 00000 10100 READ2 PZE **,EOB
00045 0 00074 0 00100 10101 PZE EDF,**ERR

BINARY CARD ID. BCREAD02
00046 4 00002 2 00000 10000 IOCPSN **,2
00047 0 0026 0 00000 10000 ID IOCCP **,22
00050 1 77752 1 01001 10111 TXI **+1,1,-22
00051 1 00026 2 01001 10111 TXI **+1,2,22
00052 0620 00 0 00041 10001 TXA TXA
00053 0634 00 0 00057 10001 LASTA CLA DONE
00054 0601 00 4 01002 10111 STO **,2
00055 4634 00 1 00047 10001 SXD ID,1
00056 0620 00 0 00043 10001 TRA READ
00057 0620 00 0 01001 10111 DONE TRA **,1

100
00060 0774 00 4 00041 10001 AXT SXA,4
00061 0634 00 4 00052 10001 SXA LASTC-1,4
00062 0774 00 4 00026 10000 AXT 22,4
00063 4634 00 4 00047 10001 SXD IO,4
00064 0074 00 4 07000 10011 TSX CLOSE,4
00065 5 00000 0 00000 10000 SHUT MON **
00066 RETURN BCREAD
00067 000000000000 00010 EOB CALL *FXEM*(EOB2)
00067 0074 00 4 04000 10011

BINARY CARD ID. BCREAD03
00070 1 00001 0 01003 10011
00071 0 00110 0 00074 10100
00072 0 00000 0 00104 10001
00073 0020 0 0 00100 10001
00074 0074 0 0 04000 10011
00075 1 00010 0 01003 10011
00076 0 00110 0 0076 10100
00077 0 00000 0 00103 10000
00100 000000000000 00100 EOF CALL EXIT
00101 0074 0 0 05000 10011
00102 1 00000 0 01002 10011
00103 0 00110 0 0077 10100
00104 0 00000 0 0044 10000 ERR2 PZE 35
00105 0 00000 0 12000 10011 INS PZE *UN05*
00106 0 00000 0 13000 10011 READ5 PZE *READ5*
00107 0 00000 0 00000 10000 TEMP PZE
00110 000000000000 10000 *DIR

BINARY CARD ID. BCREAD04
00111 222351252124 10000
00000 01111 END

CONTROL DICTIONARY

$CDICT BCREAD

BINARY CARD ID. BCREAD06
00111200000000
000000000000
222351252124
000112000000
222351252124
0CC000000000
222351252124
000000000000
332667254433
2000CC100000
256731636060
2000CC100000
670262434623
200000000000
332343466225
200000000000
334647254560
2000C0000000
335125212460
2C0000000000
3364450C0533
2C0000000000

BINARY CARD ID. BCREAD07
335125212405
2C0000000000

$DKEND BCREAD
## SYMBOL REFERENCE DATA

### REFERENCES TO DEFINED SYMBOLS.

<table>
<thead>
<tr>
<th>CLASS</th>
<th>SYMBOL</th>
<th>VALUE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCREAD</td>
<td>00000</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>DONE</td>
<td>00057</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>EDB2</td>
<td>01104</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>EDI</td>
<td>00067</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>EOF</td>
<td>00100</td>
<td>45,73</td>
<td></td>
</tr>
<tr>
<td>ERK2</td>
<td>00103</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>ERR</td>
<td>00074</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>INS</td>
<td>00105</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>00047</td>
<td>41,55,63</td>
<td></td>
</tr>
<tr>
<td>IX1</td>
<td>00037</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>IX2</td>
<td>00040</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>.DOUC1</td>
<td>00003</td>
<td>7,10,11</td>
<td></td>
</tr>
<tr>
<td>.DOUC2</td>
<td>00004</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>.DOUC3</td>
<td>00005</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LASIC</td>
<td>00053</td>
<td>42,61</td>
<td></td>
</tr>
<tr>
<td>MON</td>
<td>00036</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>READ2</td>
<td>00044</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>READ5</td>
<td>00106</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>READ</td>
<td>00043</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>LCIR</td>
<td>00005</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>SXM</td>
<td>00041</td>
<td>52,60</td>
<td></td>
</tr>
<tr>
<td>TEMP</td>
<td>00107</td>
<td>16,17,22</td>
<td></td>
</tr>
</tbody>
</table>

### REFERENCES TO VIRTUAL SYMBOLS.

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>VALUE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXIT</td>
<td>5</td>
<td>100</td>
</tr>
<tr>
<td>CLOSE</td>
<td>7</td>
<td>27,64</td>
</tr>
<tr>
<td>FXEN</td>
<td>4</td>
<td>67,74</td>
</tr>
<tr>
<td>OPEN</td>
<td>8</td>
<td>35</td>
</tr>
<tr>
<td>READS</td>
<td>11</td>
<td>106</td>
</tr>
<tr>
<td>READ</td>
<td>4</td>
<td>43</td>
</tr>
<tr>
<td>UNOS</td>
<td>10</td>
<td>105</td>
</tr>
<tr>
<td>SYSLUC</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

$IBM .READS 5,DECK

7094 RELMC ASSEMBLY.

```
$IBLUE .READS 02/25/66 .READ0000
$FILE .READS .READS .IN1,READY,INPUT,DLK=28,MULTIEEE,MA01,.ULIST .READ001
```
FILE DICTIONARY.
$S DICT $READS

BINARY CARD 10. $READCC
664C20000034 $READS FILE $READS
060040000000
51252124566C
66C6C6666660
$TEXT $READS

ENTRY $READS

BINARY CARD 10. $READCC
0060 C00000 0 0401 10010 $READS PZE
0060 00111

CONTROL DICTIONARY
$S DICT $READS

BINARY CARD 10. $READCC
00CC01000000 $PREFACE $READS
00CC04000000
335125212465 $READS DECK
00CC01000000
335125212465 $READS REAL
00CC01000000

$S KEND $READS

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS SYMBOL VALUE REFERENCES

FILE $READS 00CC0
LCTR 1 0
LCIR
QUAL
LCTW

103
SUBROUTINE OUTER IS THE OVERLAY (OR CHAIN) SUBROUTINE OF TDSN THAT PREFORMS THE OUTER (OR MAJOR) ITERATIONS.

THE COMMON STATEMENTS

COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBRED,
  1 KREG, KALC, KGEO, KBBC, KTBC, KLBC,
  2 KBGC, NSN, NG, NZ,
  3 NTR, NTYPES, NMAT, LBUCK, KCTYP,
  4 KSISU, KGIS, KCTR, KCOS, KSELECT, KTIMGM, NZONER, NZONEZ,
  5 KACCEL, KALC1, NFN, NZN,
  6 KBCDUP, KACAV, KGAVE, KNIP, KEDIT,
  7 KLBCUP, KACAV, KGAVE, KNIP, KEDIT,
  8 KCF, TIMAX, CALC1, OMEGA, EPS, XNF,
COMMON /CALL3/ HZ, HY, KCA, KCF,
  1 KCONV, KFINISH, KASOR, KUP,
  2 KREG, NBF, ND, NGP1, NZP1, NR1,
  3 KNI, NTYPES, NDNZ, NDN, NDNJ, NDNZ,
  4 NXNZS, NAR, NC, NIJG, NBUCK,
  5 ITMA, ITMI, ITMIST, ITMIGX, ITD,
  6 ITMAX, ITMEZ, LPSAV,
COMMON /CALL4/ EPG, EPGX, EPGM, EPGMP,
  1 NBF, OMEGAS, OMEGAP,
  2 KREG, EXTRA, EXTRA2, EXTRA3, EXTRA4, EXTRA5,
  3 KXTRA1, KXTRA2, KXTRA3, KXTRA4, KXTRA5,
COMMON /OUTER/ KDR, KDR, KDR, KDR,
  1 KNI, KNG, KI0, KWIT, KNDR,
  2 KREG, KNR, KNR, KJRR, KJRR,
  3 KNRUP, KNR,
COMMON /INNER/ ITMIG, IG, IZ, JUP,
  1 KNZ, XTIGGA, LG, NEGPR, JZ, KX

THE DIMENSION STATEMENTS

DIMENSION XI1, X2
DIMENSION XKI, XKE
DIMENSION MA(1)
DIMENSION FG(1), F(1)
DIMENSION XNOS(I), XNOSUP(I), XN2S(I)
DIMENSION XN(I), XJR(I), XJZ(I)
DIMENSION VI(I), AR(I), AZ(I)
1  DELZ(I)
DIMENSION KFING(I), KASORG(I)
DIMENSION AG(I), FNG(I)
1  SG(I), ASG(I), SCG(I)
2  SGD(I), SING(I), XNG(I)
3  XNB(I), HNLG(I), VNLG(I)
4  XNLG(I), XNLG(I)

DIMENSION CT(I), CGG(I), CGGA(I)
1  TVKL(I), SISO(I), ST(I)
2  ARSISO(I), AZSISO(I), ARST(I)
3  AZST(I), DC(I)
DIMENSION XNM(I)
DIMENSION XNN(I), XJRN(I), XJZN(I)
DIMENSION XNA(I), XJRA(I), XJZA(I)
DIMENSION BUCKG(I), MXVAR(I), ESJN(I)
DIMENSION CUPS(I), CNZN(I)

C THE FORMAT STATEMENTS
100 FORMAT (1H1)
104 FORMAT (1PL2)PROGRAM STOP AT KEFN=,I8)
120 FORMAT (12HITMA ITMI ITMIX ITD TIME REQ EPG
1EPGM KASOR KFINSH NEUTRIN BALGLAM E
2GEN )
121 FORMAT (14H16,14,17,19,10.6,15,16,18.8,2F16.8)
122 FORMAT (20H4HUPSZ.8X,6HNGP/1.5X,16HSINGP/I.5X,5X,10HNLG(I),6X,5X,
7 10HNLG(I))
123 FORMAT (15X,8F16.6)
125 FORMAT (5H10HFLUX ERROR,7X,11ERROR RATIO,7X,12F10.6,5X,5HFLUX ERROR,7X,11ERROR RATIO,7X,12F10.6)
C THE INTEGER FUNCTIONS
INDEXLENGTH,INDEXO,INDEXL) LENGTH*(INDEXO-1)+INDEXL
C C
1000 ITMI=ITMIST
1001 IF (KBCRED) 1005,1005,1020
1005 KONV=0
1010 CALL FISION (MA, V, C, FG, XK, XN, XJR, XJZ, F, XNOS, XNOSUP,
1  XN2S )
1015 KEND=1
1020 IF (INESOR) 1025,1025,1015
1025 IF (ITMPRT) 1050,1050,1030
1020 KBCRED=0

THE BEGINNING OF THE MAJOR ITERATION LOOP. BEGIN THE LOOP IN GROUP KGST.

ITMA = ITMA + 1
ITD = 0

IG = KGST
ITMIGX = 0
KFINGX = 0
KASOR = 0
EPGMP = EPGM
EPGM = 0.0
USP1 = 0.0
USP2 = 0.0
IF (ITMRT-2) 1061, 1070, 1061
1061 IF (ITMA-IG) 1065, 1065, 1070
1065 TEMP = ITMA
GO TO 1075
1070 TEMP = 10.0
1075 EPG = EPG/TEMP

ITERATIONS IN A NEW GROUP IG BEGIN HERE. BEGIN AT THE TOP OF THE REACTOR STARTING A Z DOWN AND UP PASS.

JUP = 0
IZ = NZ
ITMIG = 0
KFING(IG) = 2
SG(IG) = 0.0
IJJG = INDEX(NIJ, IG, 0)
DO 1105 IJ = I, NIJ
   IJJG = IJJG + 1
   XNN(IJ) = XN(IJJG)
   IF (KSISU) 1150, 1150, 1110
1110 IJJG = INDEX(NIJ, IG, 0)
   DO 1115 IJ = I, NIJ
   IJJG = IJJG + 1
1115 XJR(IJJG) = XJR(IJJG)
   IF (NZ-1) 1150, 1150, 1120
1120 IJJG = INDEX(NIJ, IG, 0)
   DO 1125 IJ = I, NIJ
   IJJG = IJJG + 1
1125 XJZ(IJJG) = XZ(IJJG)
TO COMPUTE THE ISOTROPIC FIXED SOURCE SISO WHICH INCLUDES THE FISSION SOURCE AND THE ANISOTROPIC FIXED SOURCES ARSISO AND AZSISO. ALL THE FIXED SOURCES INCLUDE UP AND DOWN SCATTERING BUT NO WITHIN GROUP SCATTERING.

1150 CALL FIXED (XKE, MA, C, F, V, XN, XJR, XJZ,
1 SG, SISO, ARSISO, AZSISO )

ITERATIONS FOR A NEW Z DOWN AND UP PASS IN A GIVEN GROUP IG BEGIN HERE.

1200 CALL INNEK ( XKE(1),
1 MA(1),C(1), X1(KD1),X1(KD2),X1(KM1),X1(KW), F(1),
2 XNOS(1),XNOSUP(1),XNZS(1),
3 AZ(1),DELZ(1),KFING(1),KASOR(1), SG(1),ASG(1),
4 XNG(1),HNLG(1),VNLG(1),XNLG(1),
5 CT(1),CGG(1),CGGAI(1),TVLK(1),SISO(1),ST(1),ARSISO(1),AZSISO(1),
6 ARST(1),AZST(1),BC(1), X1(KNI),X1(KNO),X2(KNOR),X3(KNORUP),
7 XNMI(1),X2(KN2),X2(KN3),
8 XJR(1),XJZ(1),XJRA(1),XJRA(1),X2(KJR),X2(KJZ),
9 XZ(180),X2(KWHITE), BUCK(1),BUCK(1), MXVAR(1),ESIJ(1) )

1300 KFINSH=KFINSH+ABS(KFING(1))
1305 IF (KSIU(1) 1315,1315,1305
1310 KASOR=KASOR+1
1315 ITM1=ITM1*ITM1
1320 ITMIGX=ITMIG
1325 IF (EPG1-EPGM) 1400,1400,1330
1330 EPG1=EPG1
TO COMPUTE THE NEUTRON BALANCE PER GROUP.

1400 IF (NBF-1) 1415,1415,1405
1405 BNLG(IG)+=0.0
DO 1410 IJ=1,NIJ
1410 BNLG(IG)=BNLG(IG)+TVLK(IJ)*XNN(IJ)
BNLG(IG)=XNLG(IG)+BNLG(IG)
1415 IF (LBUCK) 1425,1425,1420
1420 TEMP=BUCK(IG)*XNG(IG)
XNLG(IG)=XNLG(IG)+TEMP
1425 IF (NBUCK) 1500,1500,1430
1430 BNLG(IG)=0.0
DO 1435 IJ=1,NIJ
K=IABS(MA(IJ))
K1=INDEXNG(K,IG)
1435 BNLG(IG)=BNLG(IG)+BUCK(K1)*V(IJ)*XNN(IJ)
XNLG(IG)=XNLG(IG)+BNLG(IG)
1500  SING(IG) = SG(IG) - FG(IG)
        AG(IG) = 0.0
        FNG(IG) = 0.0
        SCG(IG) = 0.0
        SDG(IG) = 0.0
        K1G = INDEX(INTYPES, IG, KCGG)
        K1A = INDEX(INTYPES, IG, KCA)
        DO 1535 IJ = 1, NIJ
        K = ABS(KA(IJ))
        XNVIJ = V(IJ) * XNN(IJ)
        K2G = INDEX(NIYNG, K, K1G)
        K2A = INDEX(NIYNG, K, K1A)
        AG(IG) = AG(IG) + C(K2A) * XNVIJ
        FNG(IG) = FNG(IG) + C(K2A+1) * XNVIJ
        SCG(IG) = SCG(IG) + C(K2G) * XNVIJ
        K1 = INDEX(NIYNG, K, IG)
1535  SDG(IG) = SDG(IG) + CN2N(K1) * XNVIJ
C
1550  XNB(IG) = SING(IG) - XNLG(IG) - AG(IG) - SDG(IG)) / XNF
C
C
C
DOVER-RELAXATION OF THE OUTER ITERATIONS.
C
C
2000 IF (ABS(KACCEL) - 2) > 2150, 2005, 2150
2005 IF (ITMA-1) > 2150, 2150, 2010
2015 IF (ABS(SCALUP-1.0) <= 50.0 * EPS) > 2020, 2020, 2150
2020 IF (EPGMP-10.0 * EPS) > 2025, 2025, 2150
2025 IF (IJG) = INDEX(NIJ, IG, 0)
        DU = 2030 IJ = I + NIJ
        IJG = IJG + 1
        XNA(IJ) = XN(IJG)
2030 IF (IJG) = INDEX(NIJ, IG, 0)
        XNN(IJ) = OMEGAP * XNN(IJ) - OMEGPS * XNA(IJ)
C
DO 2050 J = 1, NIJ
        IJ = INDEX(NR, J, NR)
        TEMP = ORSCAL(IJ, OMEGAP, UMEGPS, XNA, XNN)
        DO 2050 M = 1, ND
        K = INDEX(ND, J, M)
        L = INDEX(NDNZ, IG, K)
        XNOS(L) = XNOS(L) * TEMP
        IF (NZ-I) > 2050, 2050, 2046
        XNOSUP(L) = XNUSUP(L) * TEMP
2046 CONTINUE
        IF (LZSAV) > 2080, 2080, 2055
2055 DU = 2075 I = 1, NR
        IJ = INDEX(NR, NZ, I)
        TEMP = ORSCAL(IJ, OMEGAP, UMEGPS, XNA, XNN)
        DO 2075 M = 1, ND
        K = INDEX(ND, J, M)
        L = INDEX(NDNZ, IG, K)
2075 XNZS(L) = XNZS(L) * TEMP
2080 IF (KSISO) > 2150, 2150, 2085
2085 DU = 2090 IJ = I + NIJ

108
TEMP = USCAL(IJ, OMEGAP, OMEGPS, XNA, XNN)
XJRNI(IIJ) = XJRN(IIJ) * TEMP
IF (NZ-1) 2090, 2090, 2089
2089 XJZN(IIJ) = XJZN(IIJ) * TEMP
2090 CONTINUE
C THE SUMMING TO OBTAIN THE UP SCATTERING ERROR.
C
2150 IF (KUPS) 1800, 1800, 2155
2155 IF (KREG) 2157, 2157, 2156
2156 IF (IG-NG) 2157, 1800, 1800
2157 IF (IG-KUPS) 1800, 2160, 2160
2160 IJG = INDEX(NIJ, IG, 0)
DO 2165 IJ = I, NIJ
   IJG = IJG + 1
   M = IABS(MA(IIJ))
   K = INDEX(NG, M, IG)
   TEMP = VS(IJ) * UPSI(K) * IXNI(IJG) - XNN(IIJ)
   UPS1 = UPS1 + TEMP
2165 UPS2 = UPS2 + ABS(TEMP)
C FINISHED WITH GROUP.
C
1800 IJG = INDEX(NIJ, IG, 0)
DO 1805 IJ = I, NIJ
   IJG = IJG + 1
1805 XN(IJG) = XNN(IIJ)
   IF (KSISO) 1900, 1900, 1810
1810 IJG = INDEX(NIJ, IG, 0)
DO 1815 IJ = I, NIJ
   IJG = IJG + 1
1815 XJRI(IJG) = XJRN(IIJ)
   IF (NZ-1) 1900, 1900, 1820
1820 IJG = INDEX(NIJ, IG, 0)
DO 1825 IJ = I, NIJ
   IJG = IJG + 1
1825 XJZ(IJG) = XJZN(IIJ)
C GROUP INDEXING.
C
1900 IF (KREG) 1901, 1901, 1925
1901 IG = IG + 1
   IF (KGST-1) 1905, 1905, 1910
1905 IF (IG-NG) 1100, 1100, 2200
1910 IF (IG-KGST) 1100, 2200, 1915
1915 IF (IG-NG) 1100, 1100, 1920
1920 IG = 1
GO TO 1100
1925 IG = IG + 1
   IF (KGST-NG) 1935, 1930, 1930
1930 IF (IG) 2200, 2200, 1100
1935 IF (IG-KGST) 1940, 2200, 1100
1940 IF (IG) 1945, 1945, 1100
1945 IG = NG
GO TO 1100

SUMS OF GROUP QUANTITIES ARE COMPUTED AND STORED IN THE NG+1 POSITION.

2200 SING(NGPI)=0.0
SDG(NGPI)=0.0
SCG(NGPI)=0.0
XNLG(NGPI)=0.0
HNLG(NGPI)=0.0
IF (NZ=1) 2202,2202,2201
2201 VNLG(NGPI)=0.0
2202 IF (INBF-1) 2203,2203,2204
2203 IF (INBUCK) 2205,2205,2204
2204 BNLG(NGPI)=0.0
2205 AG(NGPI)=0.0
XNB(NGPI)=0.0
XNG(NGPI)=0.0
DO 2215 IG=1,NG
XNG(NGPI)=XNG(NGPI)+XNG(IG)
SING(NGPI)=SING(NGPI)+SING(IG)
SDG(NGPI)=SDG(NGPI)+SDG(IG)
SCG(NGPI)=SCG(NGPI)+SCG(IG)
XNLG(NGPI)=XNLG(NGPI)+XNLG(IG)
HNLG(NGPI)=HNLG(NGPI)+HNLG(IG)
IF (NZ=1) 2207,2207,2206
2206 VNLG(NGPI)=VNLG(NGPI)+VNLG(IG)
2207 IF (INBF-1) 2208,2208,2209
2208 IF (INBUCK) 2210,2210,2209
2209 BNLG(NGPI)=BNLG(NGPI)+BNLG(IG)
2210 AG(NGPI)=AG(NGPI)+AG(IG)
2215 XNB(NGPI)=XNB(NGPI)+XNB(IG)

CALCULATE UP SCATTERING SCALING.

2225 IF (KUPS) 2300,2300,2226
2226 SCALUP=FG(NGPI)/(FG(NGPI)+UPSL)
IF (SCALUP) 2227,2227,2230
2227 SCALUP=1.0
GO TO 2300
2230 IF (KREG) 2231,2231,2232
2231 ITEMP=NG
GO TO 2235
2232 ITEMP=NG+1
2235 DO 2245 IG=KUPS,ITEMP
IJG=INDEX(INIJ,IG,0)
DO 2240 IJ=1,NIJ
IJG=IJG+1
2240 XN(IJG)=SCALUP*XN(IJG)
DO 2245 J=1,NZ
DO 2245 M=1,ND
K=INDEX(KD,J,M)
L=INDEX(INNZ,IG,K)
XNOS(L)=SCALUP*XNOS(L)
IF (NZ-1) 2245, 2245, 2243
2243 XNOSUP(L)=SCALUP*XNOSUP(L)
2245 CONTINUE
   IF (LZSAV) 2260, 2260, 2250
2250 DO 2255 I=1, NR
   DO 2255 M=1, ND
      K=INDEX(ND, I, M)
      L=INDEX(NDNR, I, K)
2255 XNZS(L)=SCALUP*XNZS(L)
2260 IF (KSISU) 2285, 2285, 2265
2265 IJG=INDEX(NIJ, I, G, 0)
   DO 2270 IJ=1, NIJ
      IJG=IJG+1
   CONTINUE
2270 XJR(IJG)=SCALUP*XJR(IJG)
   IF (NZ-1) 2285, 2285, 2271
2271 IJG=INDEX(NIJ, IJG, 0)
   DO 2275 IJ=1, NIJ
      IJG=IJG+1
2275 XJZ(IJG)=SCALUP*XJZ(IJG)
2285 CONTINUE
C NEW LAMBDA QUANTITIES COMPUTED.
C
2300 GLAM=GLAM
   IF (ABS(SCG(NGPII))-0. LE-6) 2305, 2305, 2310
2305 SDGF1=1.0
   SCAT=1.0
   GO TO 2350
2310 TEMP=SDG(NGPG)/FG(NGPG)
   SCAT=SDGF1/TEMP
   SDGF1=TEMP
C TO COMPUTE THE NEW FISSION SOURCE.
C
2350 EPG1=EPG
   CALL FISON ( MA, V, C, FG, XKE, XN, XJR, XJZ, F, XNOS, XNOSUP,
      1 XNZS )
   IF (NEGSOR) 2375, 2375, 2355
2355 KEND=1
   WRITE (6,104) KEFN
   KEFN=2355
   KDNV=3
   GO TO 2500
C COMPUTE CONVERGENCE NUMBERS.
C
2375 DIFONE=1.0-GLAM
   TEST1=ABS(DIFONE)
   TEST2=RYP*ABS(1.0-SCAT)
   TEST3=ABS(GLAM1-GLAM)
   IF (KREG) 2377, 2377, 2376
2376 TEST4=0.0
GO TO 2375
2377 IF (N2N) 2378,2378,2376
2378 TEST4=ABS((SING(NGP1)-SDG(NGP1))/XNF)
2379 CALL TIME1(TIMEY)
    TIMEZ=ABS((TIMEY-TIMEX))/3600.0
    IF (KALC=1) 2385,2380,2385
2380 EIGEN=KKI(NGP1)/XKE(NGP1)
    GO TO 2400
2385 IF (GLAM1) 2390,2395,2390
2390 EXTRP1=EXTRP
    EXTRP=(EIGEN-EIGEN)/(GLAMP-GLAM)
    TESTS=ABS((EXTRP1/EXTRP)-1.0)
2395 GO TO 2500
C C C
TEST CONVERGENCE NUMBERS.
C C C
2400 IF (KFINISH) 2401,2401,2415
2401 IF (TEST1-EPG) 2405,2405,2415
2405 IF (TEST2-EPG) 2410,2410,2415
2410 IF (TEST3-EPG) 2411,2411,2415
2411 IF (TEST4-EPG) 2412,2412,2415
2412 IF (KUP3) 2445,2445,2445
2445 IF ((UP32/XNG(NGP1))-EPS) 2445,2445,2445
2446 IF (ITM1-(ITMIX+ITMIST)) 2425,2420,2420
2420 KONV=1
    GO TO 2440
2425 IF (TIMEZ-TIMAX) 2450,2430,2430
2430 KONV=1
2440 ITMIST=ITMI
    GO TO 2500
2445 IF (KONV=2) 2446,2475,2446
2446 IF (ITMA-1G) 2447,2475,2475
2447 KUNV=2
    GO TO 2455
C C
2450 KONV=0
2455 IF (ITMPK) 1050,1050,2460
2460 WRITE (6,121) ITMA,ITMI,ITMIX,ITU,TIMEZ,EPG1,EPGM,KASOR,KONV,
    1 KFINISH,XNB(NGP1),GLAM,EIGEN
    IF (N2=1) 2466,2466,2467
2466 TEMP=0.0
    GO TO 247C
2467 TEMP=VNLG(NGP1)
2470 WRITE (6,123) UPS2,XNG(NGP1),SING(NGP1),SDG(NGP1),AG(NGP1),
    1 HNLG(NGP1),TEMP,XNLG(NGP1)
    WRITE (6,124)
    GO TO 105C
C C
2475 KUNV=2
C C C
2500 TIMEX=EPG1
C C
2595 IF (KEND) 2575,2575,2560
2560 KEFN=2550
2570 WRITE (6,104) KEFN
C C
2575 KCHAIN=3
C C
RETURN
C
END
FUNCTION ORSCAL ( IJ, SCALE1, SCALE2, XNA, XNN )

REACTOR ANALYSIS SECTION    CLAYTON BARBER

FEBRUARY 11, 1966        VERSION TN

THE DIMENSION STATEMENTS
DIMENSION    XNA(1),    XNN(1)

1000 TEMP=XNN(IJ)+SCALE2*XNA(IJ)
    IF (TEMP) 1005,1010,1005
1005 ORSCAL=SCALE1*XNN(IJ)/TEMP
    RETURN
1010 ORSCAL=1.0
    RETURN

END
SUBROUTINE FISION ( MA, V, C, FG, XKE, XN, XJR, XJZ, F,
1  XNOS, XNOSUP, XNZS )

SUBROUTINE FISION FOR SUBROUTINES SETUP AND OUTER OF TDSN

TO COMPUTE THE FISSION NEUTRON SOURCE DENSITY F(IJ), FISSION
NEUTRON SOURCE GROUP SUMS FG(I), AND THE NEW LAMBDA. ALSO TO
COMPUTE THE EFFECTIVE NEUTRON FISSION SPECTRUM CHI AND THE
FISSION NEUTRON SOURCE GROUP SUMS USING THE NEW LAMBDA. TO
NORMALIZE THE FISSION SOURCES AND THE FLUXES TO A GIVEN
NORMALIZATION FACTOR XNF SUCH THAT THE TOTAL FISSION SOURCE
EQUALS XNF. AND FINALLY TO COMPUTE THE INNER ITERATION EPSILON
EPG.

THE COMMON STATEMENTS

COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED, KBRED, KBGC, KBGR, KRC, KGEO, KSP, KSP, KSP, KSP,
1  KREG, KALC, KGEO, KBBC, KTBC, KLBC, KALC,
2  KREG, KALC, KGEO, KBBC, KTBC, KLBC, KALC,
3  NR, NTPS, NMAT, LBUCK, LGEO, LGEO, LGEO, LGEO,
4  KSISO, KGST, KCTR, KCGG, KDRAD, KFLUX, KFLUX,
5  KACCEL, KACCAV, KACCAV, KACCAV, KACCAV, KACCAV, KACCAV,
6  KBCDUP, KACAV, KACAVE, KNIP, KEDIT, KEDIT, KEDIT,
7  KBCDUP, KACAV, KACAVE, KNIP, KEDIT, KEDIT, KEDIT,
8  RYF, TIMAX, GLAM, EIGEN, EPS, XNF, OMEGA, OMEGAP
COMMON /CALL3/ HZ, HY, BF, KCA, KCF,
1  KUNV, KFINSH, KASOR, KUPS, OMEGA, OMEGA,
2  KUNV, KFINSH, KASOR, KUPS, OMEGA, OMEGA,
3  NBF, NGP1, NZP1, NRP1, OMEGA, OMEGA,
4  NBF, NGP1, NZP1, NRP1, OMEGA, OMEGA,
5  MJJ, NTPS, NDNR, NDNR, NDNR, NDNR, NDNR,
6  MXNZS, NAR, NIG, NIG, NIG, NIG,
7  ITM, ITMI, ITMI, ITMI, ITMI, ITMI,
8  ITM, ITMI, ITMI, ITMI, ITMI, ITMI,
9  TIMEX, TIMEZ, SDGF1, SDGF1, SDGF1, SDGF1,

COMMON /CALL4/ EPG, EPGX, EPGM, SCALUP, SCALUP,
1  OMEGA5, OMEGAPS, OMEGAPS, OMEGAPS, OMEGAPS,
2  EXTRA1, EXTRA2, EXTRA3, EXTRA4, EXTRA5, EXTRA5,
3  KXTA1, KXTA2, KXTA3, KXTA4, KXTA5, KXTA5

 THE DIMENSION STATEMENTS

DIMENSION MA(1), V(1), C(1)
DIMENSION FG(1), XKE(1), XN(1)
1  XJR(1), XJZ(1)
DIMENSION F(1), XNOS(1), XNOSUP(1)
1  XNZS(1)
THE FORMAT STATEMENTS
100 FORMAT (25H TOTAL SOURCE FG(NGP1) OF, E16.8, 24H IS ZERO OR NEGATIVE)

THE INTEGER FUNCTIONS
INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL

1000 FTP=FG(NGP1)
NEGSOR=0

TO COMPUTE THE FISSION NEUTRON SOURCE DENSITY AND ITS VOLUME INTEGRAL BY GROUPS:

1010 DO 1015 J=1,NIJ
1015 F(IJ)=0.0
DO 1040 IG=1,NG
IJG=INDEX(NGP1,IG,0)
IF (KREG) 1020,1C20,1030
1020 DO 1025 IJ=1,NIJ
IJG=IJG+1
K=IABS(IA(IJ))
K1=INDEX(NTYPES,IG,KCF)
K2=INDEX(NTYPES,K,K1)
1025 F(IJ)=F(IJ)+C(K2)*XN(IJG)
GO TO 1040
1030 DO 1035 IJ=1,NIJ
IJG=IJG+1
1035 F(IJ)=F(IJ)+XKE(IG)*XN(IJG)
1040 CONTINUE

FG(NGP1)=0.0
IF (KREG) 1060,1060,1045
1045 DO 1055 IG=1,NG
FG(IG)=0.0
K1=INDEX(NTYPES,IG,KCF)
DO 1050 IJ=1,NIJ
K=IABS(IA(IJ))
K2=INDEX(NTYPES,K,K1)
1050 FG(IG)=FG(IG)+C(K2)*F(IJ)*V(IJ)
1055 FG(NGP1)=FG(NGP1)+FG(IG)
GO TO 1070
1060 TEMP=0.0
DO 1061 IJ=1,NIJ
1061 TEMP=TEMP+V(IJ)*F(IJ)
DO 1065 IG=1,NG
FG(IG)=TEMP*XKE(IG)
1065 FG(NGP1)=FG(NGP1)+FG(IG)
1070 IF (KCALC) 1100,1100,1075
1075 IF (FG(NGP1)) 1080,1080,1100
1080 WRITE (6,1001) FG(NGP1)
NEGSOR=1

TO CALCULATE THE NEW LAMBDA.
NORMALIZE THE FISSION SOURCES AND THE FLUXES TO A GIVEN INPUT
NORMALIZATION FACTOR XNF SUCH THAT THE TOTAL FISSION NEUTRON
SOURCE WILL EQUAL XNF FOR THE PROBLEM SETUP AND TO NORMALIZE
FOR THE GROWTH FACTOR LAMBDA (GLAM) AFTER EACH POWER ITERATION.

IF (KALC-1) 115G, 1130, 1150
TEMP=1.0/GLAM
DO 1135 IG=1,NGP1
XKE(IG)=TEMP*XKE(IG)
1135 FG(IG)=TEMP*FG(IG)
IF (IKREG} 117S,1175,1140
1140 DO 1145 IJ=1,NIJ
1145 F(IJ)=TEMP*F(IJ)
GO TO 1175
C
1150 NFN=0
EIGEN=1.0
IF (KALC} 1155,1155,1175
1155 CONTINUE
GO TO 1350
C
1175 IF (XNF) 1350,1350,1180
1180 TEMP=XNF/FG(NGP1)
IF (ABS(TEMP-1.0)-0.1E-9) 1350,1350,1200
1200 DO 1205 IG=1,NGP1
1205 FG(IG)=TEMP*FG(IG)
DO 1210 IJ=1,NIJ
1210 F(IJ)=TEMP*F(IJ)
C
1225 DO 1250 IJG=1,NIJG
1250 XN(IJG)=TEMP*XN(IJG)
DO 1255 L=1,NDNZNG
1255 XNOS(L)=TEMP*XNOS(L)
IF (NZ-1) 1280,1280,1270
1270 DO 1271 L=1,NDNZNG
1271 XNOSUP(L)=TEMP*XNOSUP(L)
DO 1275 L=1,NXNZS
1275 XNZS(L)=TEMP*XNZS(L)
1280 IF (KS150) 1350,1350,1285
1285 DO 1295 IJG=1,NIJG
XJR(IJG)=TEMP*XJR(IJG)
IF (NZ-1) 1295,1295,1290
1290 XJZ(IJG)=TEMP*XJZ(IJG)
1295 CONTINUE
C
C TO COMPUTE THE EPSILON EPG FOR THE INNER ITERATIONS.
C
1350 TEMP=NG+3
TEMP=40.0*EPS/TEMP
FPG=TEMP*FG(NGP1)
C
C
1400 RETURN
C
END
SUBROUTINE FIXED COMPUTES THAT PORTION OF THE SOURCE THAT REMAINS FIXED WITHIN THE GROUP (INNER) ITERATIONS.

THE COMMON STATEMENTS

COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KACRED
1 KREG, KALC, KGEQ, KBBC, KTBC, KLBC,
2 KRBC, NSN, NG, NZ,
3 NR, NTYPa, NMAT, LBUCK, KCTYP,
4 KSISO, KGST, KCTR, KCGG, KOREAD, KFLUX1,
5 KACCEL, KACCL, KIMIMX, KIMIGM, NZONER, NZONEZ,
6 KBCDUP, KALAV, KIDP, KN1P1, KEDIT,
7 KBCDUP, KALAV, KGAVE, KN1P1, KEDIT,
8 GLAM, EIGEN, EPS, XNF,
9 RYF, TIMAX, TACLC, OMEGA, OMEGAP

COMMON /CALL3/
1 HZ, HY, BF,
2 KCA, KCF,
3 KCONV, KFINSH, KASOR, KUPS,
4 NBF, NGP1, NZP1, NR1,
5 NIJ, NITYP, NDR, NDIJ, NDNZNG,
6 NXNZS, NAK, NC, NIJG, NBUCK,
7 ITMA, ITMI, ITMIST, ITMIGX, ITD,
8 NEGSOR, LSAV,
9 TIMEX, TIZ, SDFG1, SCALUP

COMMON /CALL4/ EPG, EPGX, EPGM, EPGMP,
1 OMEGAS, OMEGPS,
2 EXTRA1, EXTRA2, EXTRA3, EXTRA4, EXTRA5,
3 KTRA1, KTRA2, KTRA3, KTRA4, KTRA5

COMMON /INNER/ ITMIG, IG, IZ, JUP,
1 XITGG, XITGG1, LG, NEGPR, JZ, KK

THE DIMENSION STATEMENTS

DIMENSION XKE(1)
DIMENSION MA(1), C(1)
DIMENSION F(I)
DIMENSION V(I)
DIMENSION XN(1), XJR(I), XJZ(I)
DIMENSION SG(1)
DIMENSION SISO(1), ARSISO(1), AZISO(1)

THE INTEGER FUNCTIONS

INDEX(LENGTH, INDEX0, INDEXL) = LENGTH*(INDEXO-1)+INDEXL
1000 IF (KREG) 1140, 1140, 1130
1130 K1=INDEX(NTPS, IG, KCF)
DO 1135 IJ=1, NIJ
K=IABS(MA(IJ))
K2=INDEX(NTPYNG, K, K1)
1135 SISO(IJ)=C(K2)*F(IJ)
GO TO 1150
1140 DO 1145 IJ=1, NIJ
1145 SISO(IJ)=KXE(IG)*F(IJ)

C
1150 IF (KSISO) 1155, 1155, 1151
1151 DO 1152 IJ=1, NIJ
1152 ARSISO(IJ)=0.0
IF (NZ-IJ) 1155, 1155, 1153
1153 DO 1154 IJ=1, NIJ
1154 ARSISO(IJ)=0.0
C
C UP SCATTERING
C
1155 KSCUP=KCGG-(KCT+K1)
N=IG
ITY=KCGG
IF (KSCUP) 1250, 1250, 1160
1160 IF (KREG) 1161, 1161, 1163
1161 IF ((IG+1)-KUPS) 1250, 1161, 1161
1161 IF (IG-NG) 1162, 1250, 1250
1162 N=N+1
GO TO 1170
1163 IF (IG-KUPS) 1250, 1250, 1164
1164 N=N-1
1170 IITY=NITY-1
IF (KREG) 1171, 1171, 1172
1171 K1=INDEX(NTPS, IG, IITY)
GO TO 1175
1172 K1=INDEX(NTPYNS, N, IITY)
1175 IJG=INDEX(NIJ, N, 0)
DO 1200 IJ=1, NIJ
1200 IJG=IJJG+1
K=IABS(MA(IJ))
KA=K
IF (KSISO) 1180, 1180, 1176
1176 IF (MA(IJ)) 1177, 1180, 1180
1177 KA=K+1
1180 K2=INDEX(NTPYNG, K, K1)
IF (C(K2)) 1185, 1190, 1185
1185 SISO(IJ)=SISO(IJ)+C(K2)*X(IJG)
1190 IF (KSISO) 1200, 1200, 1191
1191 IF (KA-K) 1200, 1200, 1192
1192 K2=INDEX(NTPYNG, KA, K1)
IF (C(K2)) 1195, 1200, 1195
1195 ARSISO(IJ)=ARSISO(IJ)+C(K2)*XJR(IJG)
IF (NZ-IJ) 1200, 1200, 1196
1196 AZSISO(IJ)=AZSISO(IJ)+C(K2)*XJZ(IJG)
1200 CONTINUE
IF (KREG) 1205, 1205, 1215
1205 IF (N-NG) 1210, 1250, 1250
1210 IF ((KCGG-ITY)-KSCUP) 1162,1250,1250
1215 IF (N-KUPPS) 1250,1250,1220
1220 IF ((KCGG-ITY)-KSCUP) 1164,1250,1250
C
C DOWN SCATTERING
C
1250 IF (KREG) 1251,1251,1252
1251 N=0
ITY=KCGG+IG
IF (IG-J) 1350,1350,1260
1252 N=NGP1
ITY=KCGG+(NGP1-IG)
IF (IG-NG) 1260,1350,1350
C
1260 ITY=ITY-1
IF (KREG) 1265,1265,1270
1265 N=N+1
IF (NTYPS-ITY) 1260,1266,1266
1266 KJ=INDEX(NTYPS,IG,ITY)
GO TO 1280
1270 N=N-1
IF (NTYPS-ITY) 1260,1275,1275
1275 KJ=INDEX(NTYPS,N,ITY)
1280 (IG=INDEX(NIJ,N,ITY)
DO 1325 IJ=1,NJ
IJG=IJG+1
K=IABS(MA(IJJ))
KA=K
IF (KSISO) 1290,1290,1286
1286 IF (MA(IJJ)) 1287,1290,1290
1287 KA=K+1
1290 K2=INDEX(NTIYG,K,K1)
IF (IC(K2)) 1295,1300,1295
1295 SISO(IJJ)=SISO(IJJ)+C(K2)*XN(IJJ)
1300 IF (KSISO) 1325,1325,1305
1305 IF (KA-K) 1325,1325,1310
1310 K2=INDEX(NTIYG,KA,K1)
IF (IC(K2)) 1315,1325,1315
1315 ARSISO(IJJ)=ARSISO(IJJ)+C(K2)*XJR(IJJ)
IF (NZ-I) 1325,1325,1320
1320 AZSISO(IJJ)=AZSISO(IJJ)+C(K2)*XZJ(IJJ)
1325 CONTINUE
IF (KREG) 1330,1330,1335
1330 IF (N-(IG-1)) 1260,1350,1350
1335 IF (N-(IG+1)) 1350,1350,1260
C
C FINISH FIXED SOURCE
C
1350 DO 1355 IJ=1,NJ
SISO(IJJ)=SISO(IJJ)+V(IJJ)
1355 SSI(IJJ)=SSG(IJJ)+SISO(IJJ)
IF (KSISO) 1400,1400,1360
1360 DO 1365 IJ=1,NJ
1365 ARSISO(IJJ)=ARSISO(IJJ)+V(IJJ)
IF (NZ-I) 1400,1400,1370
1370 DO 1375 IJ=1,NJ
1375 AZSISO(IJJ)=AZSISO(IJJ)+V(IJJ)
C
C
1400 RETURN
C
END
SUBROUTINE INNER (XKE, MA, C, DR, DZ, MR, W, F,
1 XNOS, XNOSUP, XNZS,
2 VA, AZ, DELZ, KFING, KASORG, SG, ASG, XNG, HNLG,
3 VNLG, XNLG, CT, CGGA, TVLK, SISO, ST, ARSISO, AZSISO, ARST,
4 AZST, DC, XN1, XNO, XNDR, XNDRUP, XNM, XNZ, XNZR,
5 XNR, XJRN, XJZN, XJRA, XJZA, XJRR, XJZR, XNM180, WHITE,
6 BUCKL, BUCKG, MXVAR, ES1JM)

SUBROUTINE INNER PREFORMS THE INNER (OR MINOR) ITERATIONS IN A
GIVEN GROUP IG.

THE COMMON STATEMENTS
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBRED, KBRED, KBBC, KTBC, KLBC,
1 KREG, KALC, KGEO, KBBC, NSN, NG, NZ,
2 KRBC, NTPS, NMAT, LBUCK, KCTYP,
3 NSN, NSN, NSN, NSN, NSN,
4 KSISO, KGST, KCTR, KCGG, KREAD, KFLUX,
5 KST, ITMMIX, ITMIGM, NZDNE, NZDNEZ,
6 KACCEL, KALC, NFN, N2N, NBM,
7 KBCDUP, KCAF, KSC, KNEP, KEDIT,
8 GLAM, EIGEN, EPS, XNF,
9 RYF, TIMAX, CALC, OMEGA, OMEGAP
COMMON /CALL3/ HZ, HY, BF,
1 KCA, KCF,
2 NFB, NFB, NFB, NFB, NFB,
3 NJ, NJ, NJ, NJ, NJ,
4 NXPNS, NAR, NC, NI, NBUCK,
5 TM, ITM, ITM, ITM, ITM,
6 NEGS, NEGPRT, JZ, KL,
7 TIME, TIME, TIME, TIME, TIME,
8 COMMON /CALL4/ EPG, EPG, EPG, EPGL, EPGL,
1 OMEGA, OMEGA, OMEGA, OMEGA, OMEGA,
2 EXRA, EXRA, EXRA, EXRA, EXRA,
3 KXTRA, KXTRA, KXTRA, KXTRA, KXTRA,
4 COMMON /INNER/ ITM, ITM, ITM, ITM, ITM,
1 XITGG, XITGG, LG, NEXG, JZ, KK

THE DIMENSION STATEMENTS
DIMENSION XKE(1)
DIMENSION MA(1), C(1)
DIMENSION DR(1), DZ(1), MR(1), W(1)
DIMENSION XNOS(1), XNOSUP(1), XNZS(1)
DIMENSION V(1), AR(1), AZ(1)
DIMENSION KFING(1), KASORG(1)
DIMENSION S(1), ASG(1), XNG(1)
2 VNLG(1), XNLG(1)

120
DIMENSION CT(1), CGG(1), CGGA(1),
2 TVLK(1), SISD(1), ST(1),
3 AZSISO(1), AZSISO(1), ARST(1),
DIMENSION XNI(1), XNO(1), XNOR(1),
2 XNORUP(1), XNM(1), XNZ(1),
DIMENSION XNN(1), XNA(1), XNR(1),
I XJRN(1), XJZN(1), XJRA(1),
2 XJZA(1), XJRR(1), XJZR(1),
DIMENSION XNM180(1), WHITE(I), BUCLK(1),
1 BUCKG(1), MXVARI(1), ESIJM(1)

C THE FORMAT STATEMENTS
150 FORMAT (1X, 6HSCALE:, F10.6, BH FOR IG=, I3, 11H AND ITMIG=, I3,
1 19H AND WAS SET TO 1.0 )
200 FORMAT (2110, 5X, 5E17, 7)
C THE INTEGER FUNCTIONS
INDEX(LENGTH, INDEXO, INDEXL)= LENGTH* (INDEXO-1) + INDEXL
C ITERATIONS FOR A NEW Z DOWN AND UP PASS IN A GIVEN GROUP IG
BEGIN HERE.
990 NEGPRT=0
1000 ITMIG=ITMIG+1
1050 ITD=ITD+1
C TO INITIALIZ THE NEW FLUX AND CURRENTS TO ZERO FOR THE DOWN AND
UP PASS.
1025 DO 1050 IJ=1, N1J
1050 XNA(IJ)=XNN(IJ)
1060 DO 1065 IJ=1, N1J
1065 XJRA(IJ)=XJRN(IJ)
C ITIALIZATIONS FOR THE TOP BOUNDARY AND THE INITIALIZATION TO
ZERO OF SUMS PREFORMED DURING A DOWN AND UP PASS ARE DONE HERE.
1100 ESC=0.0
1101 ESIJM(IG)=0.0
1102 EAM=0.0
1103 XNG(IG)=0.0
1104 IF (KSISO) 1105, 1105, 1101
1105 ASG(IG)=0.0
1106 XITGG=0.0
1107 XITGGA=0.0
1108 LG=0
HNLGR=0.0
HNLGL=0.0
HNLG(IG)=0.0
C
IF (NZ-1) 1105,1105,1110
1105 XNZS(1)=0.0
XNZ(1)=0.0
GO TO 1200
1110 VNLG(IG)=0.0
DO 1111 M=1,ND
1111 DZ(M)=DZ(M)
IF (KTBC) 1115,1115,1125
1115 DO 1120 I=1,NP
DO 1120 M=1,ND
K=INDEX(ND,1,M)
1120 XNZ(K)=0.0
GO TO 1200
1125 DO 1130 I=1,NR
DO 1130 M=1,ND
K=INDEX(ND,1,M)
1130 XNZ(K)=XNZS(K)
C TO COMPUTE THE VERTICAL TOP BOUNDARY NEUTRON LEAKAGE FOR BEFORE
C THE Z DOWN AND UP PASS.
C
1150 IF (KTBC-3) 1185,1155,1185
1155 DO 1180 I=1,NP
TEMP=0.0
TEMP1=0.0
DO 1160 M=1,ND
K=INDEX(ND,1,M)
TEMP=TEMP-W(N)*DZ(M)*XNZ(K)
1160 TEMP1=TEMP1-W(M)*DZ(M)
TEMP1=TEMP/TEMP1
DO 1165 M=1,ND
K=INDEX(ND,1,M)
1165 XNZ(K)=TEMP1
1180 CONTINUE
C
1185 DO 1195 I=1,NR
TLG=0.0
DO 1190 M=1,ND
K=INDEX(ND,1,M)
1190 TLG=TLG+W(M)*DZ(M)*XNZ(K)
1195 VNLG(IG)=VNLG(IG)+TLG*AZ(1)
C
C TO COMPUTE THE TOTAL CROSS SECTION LOSS AND STORE THE WITHIN GROUP
C SCATTERING CROSS SECTION BY MESH INTERVALS AND TO STORE
C EFFECTIVE BUCKLING CROSS SECTIONS IF USED.
C
1200 IF (LG) 1300,1300,1305
C
C C
C ITERATIONS IN A NEW Z INTERVAL IZ IN A GIVEN Z DOWN AND UP PASS
C AND A GIVEN GROUP IG BEGIN HERE.
C C
C 1200 IF (LG) 1300,1300,1305
C C
C TO COMPUTE THE TOTAL CROSS SECTION LOSS AND STORE THE WITHIN GROUP
C SCATTERING CROSS SECTION BY MESH INTERVALS AND TO STORE
C EFFECTIVE BUCKLING CROSS SECTIONS IF USED.
1300 KX=0
GO TO 1310
1305 KX=1
1310 JZ=1Z
CALL SDRAXS ( MA, V, C, CT, CGG,
               1  CGGA, TVLK, BULCK, BUCKG )
IF (LG) 1325,1325,1400
C
TO COMPUTE THE TOTAL BASIC SOURCE BY ADDING THE SELF-SCATTERING
SOURCE TO THE FIXED SOURCE FOR THE GIVEN Z MESH INTERVAL IZ.
C
1325 DD 1326 I=1, NR
IJ=INDEX(NR,IZ,I)
ST(I)=SISO(IJ)+CGG(I)*XNA(IJ)
IF (KSISO) 1400, 1400, 1350
1350 KASRGK(IG)=0
DO 1385 I=1, NR
IJ=INDEX(NR,IZ,I)
ARST(I)=ARSISO(IJ)+CGG(I)*XJRA(IJ)
IF (TEMP-ABS(ARST(I))) 1355, 1360, 1360
1365 KASORG(IG)=KASORG(IG)+1
ARST(I)=SIGN(TEMP,ARST(I))
1370 IF (JUP) 1375, 1375, 1385
1375 ASG(IG)=ASG(IG)+ARSISO(IJ)
IF (NZ-1) 1385, 1385, 1380
1380 ASG(IG)=ASG(IG)+AZSISO(IJ)
1385 CONTINUE
C
TO COMPUTE THE ANGULAR FLUX FOR ND DIRECTIONS (IN EACH HALF OF A
Z DOWN AND UP PASS FOR A TOTAL OF 2*ND DIRECTIONS) BY RADIAL
MESH INTERVALS AND TO SUM IN ORDER TO OBTAIN THE NEW AVERAGED
FLUXES XNN.
C
1400 XLLG=0.0
CALL FLUX ( AR(1), AZ(1), DELZ(1), DR(1), UZ(1), MR(1), W(1),
            1  CT(1), ST(1), ARST(1), AZST(1), DC(1),
            2  XNDSUP(1), XNM(1), XNZ(1),
            3  XNM180(1), WHITE(1), XLLG )
C
TO COMPUTE THE HORIZONTAL NEUTRON LEAKAGES.
C
1450 RLG=0.0
DO 1455 M=1, ND
1455 RLG=RLG+W(M)*DR(M)*XND(M)
RLG=RLG*AR(NRP1)*DELZ(IZ)
HNLG=HNLG+RLG
GO TO (1456, 1460, 1460), KGEO
1456 IF (KLBC) 1457, 1457, 1460
1457 XLLG=XLLG*AR(1)*DELZ(IZ)
HNLG=HNLG+XLLG
C
TO STORE THE RADIAL BOUNDARY ANGULAR FLUXES.

DO 1465 M=I,ND
K=INDEX(ND,IZ,M)
L=INDEX(NDNZ,IG,K)
IF (JUP) 1461,1461,1462
1461 XNOS(L)=XNO(M)
GO TO 1465
1462 XNOSUP(L)=XNO(_)
1465 CONTINUE

TO CONTROL THE Z DOWN AND UP PASS.

IF (NZ-I) 1470,1470,1475
1475 IF (JUP) 1480,1480,1490
1480 IF (IZ-I) 1500,1500,1495
1495 IZ=IZ+1
1490 IF (IZ-NZ) 1495,1540,1540
1500 IZ=IZ-I
1505 IF (LG) 1300,1300,1305
1300 CONTINUE

TO COMPUTE THE VERTICAL BOTTOM BOUNDARY NEUTRON LEAKAGES.

BLG=0.0
IF (KBBC) 1505,1505,1525
1505 DO 1515 I=1,NR
TEMP=0.0
DO 1510 M=I,ND
K=INDEX(ND,I,M)
1510 TEMP=TEMP-W(M)*DZ(M)_XNZ(K)
1515 BLG=BLG+TEMP_AZ(I)
DO 1520 M=I,ND
K=INDEX(ND,I,M)
1520 XNZ(K)=O.

RE-STORE THE OUTWARD DIRECTED ANGULAR FLUXES AT THE BOTTOM BOUNDARY TO BE UPWARD DIRECTED FLUXES AFTER UNDER GOING 180 DEGREE ROTATION AND REFLECTION ABOUT THE FIRST DIRECTION MID-POINT.

IF (KBBC-4) 1525,1526,1538
1526 TEMP2=MDD(NR*2)
1527 TEMP1=TEMP2*2
1528 K=0
1527 TEMP1=TEMP1+1
1528 TEMP=NRPl
1537 I=1,ITEMP1
1537 TEMP=ITEMP-1
1538 K=K
1529 M=1,ND
1529 XND(M)=XNZ(K1)
1530 MM=O
1532 M=1,ND
K=K+1
IF (ABS(W(M))<0.1E-09) 1530,1530,1531
1530 MM=MM+1
L=INDEX(NR,MM,ITEMP)
XNZ(K)+XNZM80(L)
GO TO 1532
1531 M1=MR(M)
K1=INDEX(NR,ITEMP,M1)
XNZ(K)=XNZ(K1)
1532 CONTINUE
IF (ITEMP1) 1533,1533,1537
1533 MM=0
K2=INDEX(NR,ITEMP,0)
DO 1536 M=1,ND
K2=K2+1
IF (ABS(W(M))<0.1E-09) 1534,1534,1535
1534 MM=MM+1
L=INDEX(NR,MM,1)
XNZ(K2)=XNZM80(L)
GO TO 1536
1535 M1=MR(M)
XNZ(K2)=XNZM80(M1)
1536 CONTINUE
1537 CONTINUE
C SET PARAMETERS TO GO FROM BOTTOM TO TOP THROUGH THE MESH.
C
1538 JUP=1
IZ=1
DO 1539 M=1,ND
1539 DZ(M)=-DZ(M)
IF (LG) 1300,1300,1305
C TO COMPUTE THE VERTICAL TOP BOUNDARY NEUTRON LEAKAGE FOR AFTER
C THE Z DOWN AND UP PASS.
C
C 1540 TLG=0.0
DO 1550 I=1,NR
TEMP=0.0
DO 1545 M=1,ND
K=INDEX(NR,I,M)
1545 TEMP=TEMP+W(M)*DZ(M)*XNZ(K)
1550 TLG=TLG+TEMP*AZ(I)
VNLG(I)=VNLG(I)+TLG
C TO STORE THE TOP BOUNDARY ANGULAR FLUXES.
C IF (LZSAV) 1600,1600,1650
1560 DO 1565 I=1,NR
DO 1565 M=1,ND
K=INDEX(NR,I,M)
L=INDEX(NDR,I,G,K)
1565 XNZS(L)=XNZ(K)
C
C 1600 IF (LG) 1610,1610,1650
C
C
TO CHECK FOR THE NECESSITY OF MAKING A NET LEAKAGE CORRECTION PASS AND TO SAVE CERTAIN PARAMETERS WHEN THIS IS REQUIRED.

```
1610 RAHLK=0.0
  HLK=0.0
  IF (KIBC-1) 1613,1611,1611
1611 TEMP=0.0
  DD 1612 J=1,NZ
    IJ=INDEX(NR,J,NR)
  TEMP=TEMP+XNN(IJ)
  TEMP=ABS(TEMP)
  HLK=ABS(HNLRG)
  IF (TEMP) 11612,1613,11612
11612 RAHLK=HLK/TEMP
1613 IF (NZ-1) 1618,1618,1614
1614 IF (KIBC-1) 1618,1618,1615
1615 TEMP=0.0
  DD 1616 J=1,NZ
    IJ=INDEX(NR,NZ,J)
  TEMP=TEMP+XNN(IJ)
  IF (TEMP) 11616,1617,11616
11616 RAVLK=ABS(VNLG(IG)/TEMP)
  IF (RAVLK-(EPG/10.0)) 1617,1617,1620
1617 IF (ABS(VNLG(IG))-(EPG/10.0)) 1618,1618,1620
1618 IF (RAHNL-(EPG/10.0)) 1619,1619,1620
1619 IF (HLK-(EPG/10.0)) 1700,1700,1620
1620 IF (ITMIG-1) 1700,1700,1619
1619 IF (EPGX-0.1) 11620,11620,1700
11620 LG=1
  DO 1623 IJ=1,NJ
    XNR(IIJ)=XNN(IIJ)
    XNN(IIJ)=0.0
    IF (KSISO) 1623,1623,1621
1621 XJRN(IIJ)=XJRN(IIJ)
    XJRN(IIJ)=0.0
    IF (NZ-1) 1623,1623,1622
1622 XJZRN(IIJ)=XJZRN(IIJ)
    XJZRN(IIJ)=0.0
1623 CONTINUE
  DO 1625 J=1,NZ
    M=1,ND
    K=INDEX(ND,J,M)
    L=INDEX(NDNR,IG,K)
    XNR(K)=XNNS(L)
    IF (NZ-1) 1625,1625,1624
1624 XNDRUP(K)=XNDSUP(L)
1625 CONTINUE
  IF (LZSAV) 1631,1631,1626
1626 DO 1630 I=1,NR
    M=1,ND
    K=INDEX(ND,I,M)
    L=INDEX(NDNR,IG,K)
    XNZR(K)=XNZS(L)
    IF (KIBC) 1627,1627,1628
1627 XNZ(K)=0.0
    GO TO 163
1628 XNZ(K)=XNZS(L)
1630 CONTINUE
    GO TO 1635
```
1631 IF (NZ-1) 1635,1635,1632
1632 DO 1633 K=1,NDNR
1633 XNZ(K)=0.0
1634 HNLG(IG)=HNLGR
1635 HNLGR=0.0
1636 HNLGL=HNLGL
1637 DD 1640 I=1,NR
1638 ST(I)=0.0
1639 IF (KSISO) 1640,1640,1636
1640 DO 1641 K=I,NDNR
1641 XNZIK):O,
1642 HNLG(IG)=HNLGR
1643 HNLGR=0.0
1644 HNLGL=O-O
1645 DO 1650 I=I,NR
1650 STl):O.O
1651 IF (KSISO) 1652,1652,1651
1652 ARST(I)=O.O
1653 IF (NZ-1) 1640,1640,1637
1654 DO 1655 I=I,NR
1655 ST:O
1656 DO 1660 M=I,ND
1660 DD 1665 M=I,ND
1661 DZ(M)=-DZ(M)
1662 VNLGR=VNLG(IG)
1663 VNLG(IG)=O.O
1664 BLGR=BLG
1665 JUP=O
1666 IZ=NZ
1667 GO TO 1150
1668 TO AVERAGE THE SAVED QUANTIES WITH THE CORRECTED VALUES SUCH
1669 THAT THE NET CORRECTED PORTION OF THE LEAKAGE WILL BE ZERO.
1670 TEMP1=0.0
1671 TEMP2=0.0
1672 IF (KRBC) 1652,1652,1651
1651 TEMP1=TEMP1+HNLG(IG)
1652 TEMP2=TEMP2+HNLGR
1653 IF (KTBC) 1654,1654,1653
1654 TEMP1=TEMP1+VNLGR
1655 TEMP2=TEMP2+VNLG(IG)
1656 IF (TEMP2) 1660,1655,1660
1657 TEMP2=0.0E-10
1658 TEMP=-TEMP1/TEMP2
1659 HNLGR=TEMP*HNLGR+HNLG(IG)
1660 HNLG(IG)=TEMP*HNGL+HNLGR
1661 IF (NZ-1) 1657,1657,1656
1662 VNLG(IG)=TEMP*VNLG(IG)+VNLGR
1663 VNLGR=TEMP*VNLG(IG)+VNLGR
1664 BLG=TEMP*BLG+BLGR
1665 JUP=O
1666 DO 1670 J=I,NZ
1670 DO 1670 M=I,ND
1671 K=INDEX(ND,J,M)
1672 L=INDEX(NDNZ,IG,K)
1673 XNOS(L)=TEMP*XNOS(L)+XNOR(K)
1674 IF (NZ-1) 1670,1670,1668
1675 XNOSUP(L)=TEMP*XNOSUP(L)+XNORUP(K)
1676 CONTINUE
1677 IF (LZSAV) 1700,1700,1675
1678 DO 1680 I=1,NR
1680 M=1,ND
\[ K = \text{INDEX(ND, I, M)} \]
\[ L = \text{INDEX(NDNR, IG, K)} \]
\[ XNZS(L) = \text{TEMP} \times XNZS(L) + XNZR(K) \]

To compute the net leakages.

\[ HNLG(IG) = HNLGR + HNLGL \]
\[ XNLG(IG) = HNLG(IG) \]

If \((NZ-1) < 1720\), then:
\[ VNLG(IG) = VNLG(IG) + BLG \]
\[ XNLG(IG) = XNLG(IG) + VNLG(IG) \]

To apply acceleration by over-relaxation using the FLEER method of changing the over-relaxation factor for the rest of a group's iterations after certain conditions are met if \(KACCEL\) is negative.

\[ KACCEL < 1720 \]
\[ ISCALE = 0 \]
Go to 175

\[ OMEGAT = 0 \]
\[ OMEGTP = 0 \]
If \((ITMIG-I) < 1721\), then:
\[ OMEGAT = 0 \]
\[ FLXER = I.O \]
\[ ISCALE = -I \]
If \((ISCALE-2) < 1726\), then:
\[ FLXERP = FLXER \]
\[ FLXER = FLXER + ABS((XNN(IJ) - XNA(IJ)) \times V(IJ)) \]
\[ OMEGTP = OMEGAT \]
\[ OMEGAT = FLXER / FLXERP \]

If \((EPGX - 50.0 \times EPS) < 1731\), then:
\[ IF (ISCALE) < 1732\]
\[ IF (ABS(SCALE-I,0) - 0.05 < 1733\]
\[ IF (KACCEL) < 1734\]
\[ FLXER = EPGX \]
Go to 1740

If \((ABS(OMEGAT - OMEGTP) - 0.005 < 1735\)
\[ IF (OMEGAT - OMEGAS) < 1736\]
\[ IF (OMEGAT - OMEGAS) / (OMEGA**SORT(OMEGAT)) < 1737\]
\[ IF (FACTOR < 1.0) < 1738\]
\[ SCALE1 = 2.0 / (1.0 + SORT(1.0 - (FACTOR**2))) \]
\[ SCALE2 = SCALE1 - 1.0 \]
\[ ISCALE = 2 \]
Go to 1745

Scale1 = OMEGA
Scale2 = OMEGAS
Iscale = 1
Factor = 0.
IF (ITMPRT-2) 1748,1746,1746
write (6,200) IIG,ITMIG,FLXER,DMEGAT,OMEGTP,FACTOR,SCALE1 C
DD 1749 IIG=INDEX(NTYPS,IG,KCGG)
KX=1
call SDRAXS ( MA, V, C, CT, CGG,...
1 CGGX, TVLK, BUCLK, BUCKG
dd 1775 I=1, Nr
IJ=INDEX(NR,JZ,I))
XNVIJ=V(IJ)*XNN(IJ)
XNG(IG)=XNG(IG)+XNVIJ
K=IABS(MAJ(I))
K2G=INDEX(NTYNG,K*KIG)
IF (MA(IJ)) 1751,1752,1752
1751 KA=K+1
K2GA=INDEX(NTYNG,KA,KIG)
1752 TEMP=XNN(IJ)-XNA(IJ)*V(IJ)
ESC=ESC+C(K2G)*TEMP
TEMP=ABS(TEMP)
ESIJ=ABS(C(K2G)*TEMP
IF (MA(IJ)) 1761,1762,1762
1761 ESIJ=ESIJ+ABS(C(K2GA))*TEMP
1762 IF (ESIJ=ESIJM(IG)) 1770,1770,1765
1765 ESIJM(IG)=ESIJ
MXVARI(IG)=ESIJ
1770 ESM=ESM+ABS(C(K2G))*TEMP
IF (MA(IJ)) 1771,1775,1775
1771 ESM=ESM+ABS(C(K2GA))*TEMP
1775 EAM=EAM+((CT(I)/V(IJ))-C(K2G))*TEMP
IF (XNG(IG)) 1785,1780,1785
1780 ESM=0.0
EAM=0.0
ESIJM(IG)=0.0
go to 1790
1785 ESM=ESM/XNG(IG)
EAM=EAM/XNG(IG)
ESIJM(IG)=ESIJM(IG)/XNG(IG)
ESM=ABS(ESM)
EAM=ABS(EAM)
ESIJM(IG)=ABS(ESIJM(IG))
C
1790 SCALE=SG(IG)/(SG(IG)-ESC)
if (SCALE) 1791,1791,1795
1791 write (6,150) SCALE, IG, ITMIG
SCALE=1.0
C
1795 XNG(IG)=XNG(IG)*SCALE
XNLG(IG)=XNLG(IG)*SCALE
HNLG(IG)=HNLG(IG)*SCALE
IF (NZ-1) 1800,1800,1796
1796 VNLG(IG)=VNLG(IG)*SCALE
C
SCALING OTHER QUANTITIES.
C 1800 DO 1810 J=1,NZ
  IJ=INDEX(NR,J,NR)
  IF (ISCALE) 1801,1801,1802
1801 TEMP=1.0
  GO TO 1803
1802 TEMP=DRSCAL(IJ, SCALE1, SCALE2, XNA, XNN)
1803 DO 1810 M=1,ND
  K=INDEX(ND,J,M)
  L=INDEX(NDNZ,JG,K)
  XNOS(L)=XNOSIL)*TEMP*SCALE
  IF (NZ-1) 1810,1810,1806
1806 XNOSUP(L)=XNOSUP(L)*TEMP*SCALE
1810 CONTINUE
  IF (LZSAV) 1825,1825,1811
1811 DO 1820 I=1,NR
  IJ=INDEX(NR,NZ,I)
  IF (ISCALE) 1812,1812,1813
1812 TEMP=1.0
  GO TO 1814
1813 TEMP=DRSCAL(IJ, SCALE1, SCALE2, XNA, XNN)
1814 DO 1820 M=1,ND
  K=INDEX(ND,I,M)
  L=INDEX(NDNR,I,G,K)
1820 XNZS(L)=XNZS(L)*TEMP*SCALE
1825 IF (KSISO) 1845,1845,1830
1830 DO 1840 IJ=1,NIJ
  IF (ISCALE) 1831,1831,1832
1831 TEMP=1.0
  GO TO 1833
1832 TEMP=DRSCAL(IJ, SCALE1, SCALE2, XNA, XNN)
1833 XJRN(IJ)=XJRN(IJ)*TEMP*SCALE
1836 XJZN(IJ)=XJZN(IJ)*TEMP*SCALE
1840 CONTINUE
C SCALING FLUXES
C 1845 DD 1846 IJ=1,NIJ
1846 XNN(IJ)=XNN(IJ)*SCALE
C TO CHECK THE CONVERGENCE OF THE INNER ITERATIONS.
C 1850 IF (ESM=EAM) 1855,1855,1860
1855 EPGX=EAM
  GO TO 1865
1860 EPGX=ESM
1865 IF (KFING(I)-2) 1975,1866,1866
1866 IF (ESIJM(I)-EPG/100) 1870,1870,1900
1870 IF (EPGX-EPG) 1875,1875,1900
1875 IF (KTBC) 1890,1890,1880
1880 IF (NZ-1) 1890,1890,1885
1885 IF (ABS(VNLG(I)-BLG)-(EPG/8)) 1925,1925,1925
1890 IF (KRBC) 1925,1925,1895
1895 IF (ABS(HNLGL(I)-HNLGL)-(EPG/8)) 1925,1925,1900
1900 IF (EPGX-O.05*EPGMP) 1905,1905,1915
1905 KFING(I)=-1
  GO TO 1950
1915 IF (ITMIG-ITMIGM) 1920,1935,1935

130
SUBROUTINE FLUX (AR, AZ, DELZ, DR, DZ, MR, W,
1 CT, ST, ARST, AZST, DC, XN1, XNO, XNDS, XNDUP, XNM, XNZ,
2 XNN, XJRN, XJZN, XNM180, WHITE, XLLG)

SUBROUTINE FLUX FOR SUBROUTINE INNER OF TDSN

TO COMPUTE THE ANGULAR FLUX FOR ND DIRECTIONS BY RADIAL MESH
INTERVALS AND SUM IN ORDER TO OBTAIN THE NEW AVERAGED
FLUXES XNN. THE WEIGHTING FUNCTION W(M) SUMS TO ONE SO THE
AVERAGE IS AUTOMATICALLY OBTAINED. THE CURRENTS ARE ALSO
COMPUTED FOR KSISO=1.

THE COMMON STATEMENTS
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KREG, KBCRED, KGEO, KBBC, KTBC, KLBC,
1 KRCB, NSN, NG, NZ,
2 NR, NBBC, KBBC, KREG, KACAV, KGAVE, KNIP, KEDIT,
3 KSISO, KGST, KNAPS, KSTP, KACAV, KGAVE, KNIP, KEDIT,
4 NSN, NR, NTYPS, NBBC, KBBC, KREG, KACAV, KGAVE, KNIP,
5 KSISO, KGST, KNAPS, KSTP, KACAV, KGAVE, KNIP, KEDIT,
6 KAACC, KACAV, KGAVE, KNIP, KEDIT, XNN, XJRN, XJZN,
7 KBCDUP, KCAM, KACAV, KGAVE, KNIP, KEDIT, WHITE,
8 XLLG, XNMIBO, WHITE,
9 ITMIGM, ITMIGX, ITMIX, XLLG, RH, RF, RFM, RFM1,
COMMON /CALL3/
1 HZ, HY, RF, RFM, RFM1,
2 KCHAIN, KEND, KEFN
3 KREG, KBCRED, KGEO, KBBC, KTBC, KLBC,
4 KRCB, NSN, NG, NZ,
5 NR, NBBC, KBBC, KREG, KACAV, KGAVE, KNIP, KEDIT,
6 KAACC, KACAV, KGAVE, KNIP, KEDIT, XNN, XJRN, XJZN,
7 KBCDUP, KCAM, KACAV, KGAVE, KNIP, KEDIT, WHITE,
8 XLLG, XNMIBO, WHITE,
9 ITMIGM, ITMIGX, ITMIX, XLLG, RH, RF, RFM, RFM1,
COMMON /CALL4/ EPG, EPGX, EPGL, EPGM, EPGMP, 
1 XITGG, XITGGA, LG, NERPT, JZ, JUP, 
2 EXTRA1, EXTRA2, EXTRA3, EXTRA4, EXTRA5, 
3 KXTRA1, KXTRA2, KXTRA3, KXTRA4, KXTRA5, 
COMMON /CINNER/ ITMIG, IG, IZ, JUP, 
1 XITGG, XITGGA, LG, NERPT, JZ, JU, 
2 EXTRA1, EXTRA2, EXTRA3, EXTRA4, EXTRA5, 
3 KXTRA1, KXTRA2, KXTRA3, KXTRA4, KXTRA5, 
THE DIMENSION STATEMENTS 
DIMENSION AR{l), AZ{l), DELZ{l), MR{l), 
2 W(l), 
DIMENSION CT{l), ST{l), ARST{l), 
1 AZST{l), DC{l), 
DIMENSION XNII{l), XNO{l), XNDS{l), 
1 XNDSUP{l), XNM{l), XNZ{l), 
1 XNNSUP{l), XNM{l), XNZ{l), 
DIMENSION XNN{l), XJRN{l), XJZN{l), 
1 XNN180{l), WHITE{l), 
THE FORMAT STATEMENTS 
152 FORMAT (16H ANGULAR FLUX AT,16,9H FIXED TO,EIB.5,SH FROMtEI3.5, 
1 IOH FOR ANGLE,13,1BH AND MESH INTERVAL,I4,gH IN GROUP,13, 
2 14H AND ITERATION,I3) 
153 FORMAT (15H XNANG WAS ZERO,14,90H TIMES OVER ALL ANGLES AND R MESH 
1 INTERVALS WITH NON-ZERO CROSS SECTION OF Z MESH INTERVAL,13, 
2 10H AND GROUP,13) 
THE INTEGER FUNCTIONS 
INDEX(l,LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-I)+INDEXL 
TO CALCULATE THE BOUNDARY ANGULAR FLUX FOR A WHITE RADIAL 
OUTER BOUNDARY. 
1000 IF (KRBC-3) 1006,1001,1006 
1001 IF (JUP) 1002,1002,1006 
1002 TEMP=0.0 
WHITE{lZ)=0.0 
DO 1005 M=lND 
1003 K=INDEX(ND,lZ,M) 
L=INDEX(NDNZ,IG,K) 
TEMP=TEMP+W(M)*DR{M)*XNOS(L) 
WHITE{lZ)=WHITE{lZ)+W(M)*DR{M) 
IF (NZ-I) ICO5,1005,1C04 
1004 TEMP=TEMP+W(M)*DR{M)*XNOSUP(L) 
WHITE{lZ)=WHITE{lZ)+W(M)*DR{M) 
1005 CONTINUE 
WHITE{lZ)=TEMP/WHITE{lZ) 
1006 NZNANG=O 
TO SET UP FOR 180 DEGREE BOTTOM BOUNDARY REFLECTION. 
1007 IF (KKBC-4) 11010,1007,11010 
1008 IF (JUP) 1008,1009,11010 
1009 I180=1 
M180=O 

132
SET ANGULAR FLUX BOUNDARY CONDITIONS.

IF (DR(M)) 1025, 1025, 1010
1010 M=M+M(M)
IF (1180) 11014, 11014, 11011
11011 IF (M-ND) 11012, 11013, 11013
11012 XN(M+11)=0.1E-09 11013, 11013, 11014
11013 J1B=1
11014 GO TO (1011, 1020, 1020), KGEO
1011 IF (KLC=1) 1015, 1020, 1020
1015 XLLG=XLLG-W(M1)*DR(M1)*XNI(M1)
XNI(M)=0.
XNI(M)=0.
GO TO 1100
1020 XNO(M)=XNI(M1)
XNI(M)=XNI(M1)
GO TO 1100
1025 IF (KRB=1) 1030, 1035, 1060
1030 K=INDEX(ND1, IZ, M)
GO TO 1040
1035 M=M+M(M)
K=INDEX(ND1, IZ, M1)
1040 L=INDEX(ND, 1G, K)
IF (JUP) 1045, 1045, 1050
1045 XND(M)=XNDS(L)
XNI(M)=XNDS(L)
GO TO 1100
1050 XNO(M)=XNOSUP(L)
XNI(M)=XNOSUP(L)
GO TO 1100
1060 XNO(M)=WHITE(IZ)
XNI(M)=WHITE(IZ)

COMPUTE THE GEOMETRY FUNCTIONS.

1100 DO 1500 11=1, NR
1105 I=1
GO TO 1115
1110 I=NR+1-I
1115 J=INDEX(NR, IZ, I)

IPI=I+I
MMI=M-I
DAR=ABS(DR(M))*(AR(IPI)+AR(1))*DELZ(I)
DBT=DAR
IF (NZ=1) 1155, 1155, 1160
1155 DAZ=0.0
GO TO 1165
1160 DAZ=ABS(DZ(M))*(2.0*AZ(I))
DBT=DBT+DAZ
1165 IF (ABS(W(M))<0.1E-09) 1170, 1170, 1175
1170 DC(I)=0.0
GO TO 1180
1175 DC(I)=-W(M)*DR(M)+W(MI+1)*DR(MM1)*(AR(IPI)-AR(1))*DELZ(IZ)+
               1 W(MMI)*DC(I)
DC(I)=DC(I)/W(M)

DBT=DBT+DC(I)

1180 DBT=DBT+CT(I)

C CALCULATE THE ANGULAR AVERAGE FLUX XNANG.

C 1200 IF (DR(M)) 1205,1205,1210
1205 TEMP=DR*M XN(M)
1210 GO TO 1215
1210 TEMP=DR*M XN(M)
1215 IF (NZ-1) 1225,1225,1220
1220 K1=INDEXCND,N,M)
1225 TEMP=TEMP+DAZ*XNZ(K1)
1230 IF (ABS(W(M))-0.1E-08) 1235,1235,1230
1230 TEMP=TEMP+DC(I)*XNM(I)
1235 TEMP=TEMP+ST(I)
1240 IF (KSISU) 1245,1245,1240
1240 TEMP=TEMP+DR(M)*ARST(I)
1241 IF (NZ-1) 1245,1245,1241
1241 TEMP=TEMP+DT(M)*AZST(I)
1245 XNANG=TEMP/DBT

C TO CALCULATE THE ANGULAR EXTRAPOLATED FLUXES.

C 1250 TEMP=XNANG+XNANG
1255 XNM(I)=XNANG
1260 GO TO 1265
1260 XNM(I)=TEMP-XNM(I)
1265 IF (NZ-1) 1275,1275,1270
1270 XNZ(K1)=TEMP-XNZ(K1)
1275 IF (DR(M)) 1280,1280,1285
1280 XN(M)=TEMP-XN(M)
1285 GO TO 1285
1285 XNZ(M)=TEMP-XNZ(M)

C TO TEST FOR A NEGATIVE EXTRAPOLATED VALUE FOR AN ANGULAR FLUX.

C 1300 IF (NL) 1315,1315,1310
1305 IF (XNANG) 1310,1320,133
1310 KEFN=1310
1315 TEMP=XNANG
XNANG=0.0
1320 IF (NEGRT) 1330,1330,1315
1325 WRITE (6,152) KEFN,XNANG,TEMP,M,IG,ITMIG
1330 GO TO 1330
1330 TEMP=XNANG
XNANG=0.0
1340 IF (NEGRT) 1350,1350,1345
1345 WRITE (6,152) KEFN,XNANG,TEMP,M,IG,ITMIG
1350 IF (XNZ(K1)) 1355,1355,1360
1355 KEFN=1355
1360 WRITE (6,152) KEFN,XNANG,TEMP,M,IG,ITMIG
1365 IF (DR(M)) 1370,1375,1380
1370 IF (XNI(M)) 1375,1425,1425
1375 KEFN=1375
   TEMP=XNI(M)
   XNI(M)=0.
   IF (NEGPRT) 1425,1425,1385
1380 WRITE (6,152) KEFN,XNI(M),TEMP,M,IJ,IG,ITMIG
   GO TO 1425
1385 IF (XNO(M)) 1390,1425,1425
1390 KEFN=1390
   TEMP=XNO(M)
   XNO(M)=0.
   IF (NEGPRT) 1425,1425,1395
1395 WRITE (6,152) KEFN,XNO(M),TEMP,M,IJ,IG,ITMIG
C CALCULATE THE NEW NEUTRON FLUX XNN (THE AVERAGE OF THE ANGULAR
C AVERAGE FLUXES).
C 1425 IF (XNANG) 1500,1500,1430
1430 XNN(IJ)=XNN(IJ)+W(M)*XNANG
C TO CALCULATE THE CURRENTS FOR ANISOTROPIC (P(1)) SCATTERING.
C THE MULTIPLYING FACTOR OF (2*L-1) HAS NOT BEEN INCLUDED HERE
C SINCE IT IS IN THE GAM-1I CROSS SECTIONS.
C 1450 IF (KSISD) 1500,1500,1455
1455 XJRN(IJ)=XJRN(IJ)+W(M)*DR(M)*XNANG
   IF (NZ-1) 1500,1500,1460
1460 XJZN(IJ)=XJZN(IJ)+W(M)*DZ(M)*XNANG
C CONTINUE
C STORE SOME ANGULAR FLUXES FOR 180 DEGREE BOTTOM BOUNDARY
C REFLECTION.
C 1500 IF (J180) 1550,1550,1505
1505 DO 1510 I=1,NR
   M180=M180+1
1510 XNM180(M180)=XNM(I)
C CONTINUE
C 1550 IF (NZNANG) 1600,1600,1575
1575 WRITE (6,153) NZNANG,IZ,IG
C RETURN
C END
SUBROUTINE SORAXS (MA, V, C, CT, CGG, CGGA)

SUBROUTINE SORAXS FOR SUBROUTINE INNER OF TDSN

TO LOOK UP THE WITHIN GROUP SCATTERING CROSS SECTION AND THE
TOTAL COLLISION LOSS CROSS SECTION AND COMPUTE ANY BUCKLING
LOSS CROSS SECTIONS TO ADD TO THE TOTAL COLLISION LOSS CROSS
SECTION.

THE COMMON STATEMENTS
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED
1  KREG, KALC, KGEO, KBBC, KBBC, KBCC, KBCC,
2  KRECC, KGST, KCTR, KGCC, KGCC, KGCC, KGCC,
3  NR, NTYPS, NMAI, NBUCK, LBUCK, LBUCK, LBUCK,
4  KSTISO, KSIG, KSIG, KSIG, KSIG, KSIG, KSIG,
5  KACC, KALC, NFN, N2N, N2N, N2N, N2N,
6  KBCDUP, KACAV, KGAVE, KNIP, KNIP, KNIP, KNIP,
7  KARE, KALC, NFN, N2N, N2N, N2N, N2N,
8  GLAM, EIGEN, EPS, XNF, EPS, EPS, EPS,
9  RYF, TIMAX, CALC1, OMEGA, OMEGAP

COMMON /CALL3/ HZ, HY, BF,
1  KCA, KCF, KCF, KCF,
2  KONV, KFINSH, KASOR, KUSP,
3  NBF, NGP1, NZP1, NRPI,
4  KSTISO, NTYPS, NMAI, NBUCK,
5  NNXNZS, NAR, NC, NBUCK,
6  KITMIX, ITMIX, ITMIX, ITMIX,
7  ITMIX, ITMIX, ITMIX, ITMIX,
8  TIMEX, TIMEZ, SDGFG1, SCALUP

COMMON /CALL4/ EPG, EPGX, EPGM, EPGMP,
1  OMEGAS, OMEGPS, EPS, EPS,
2  EXTRAX, EXTRAX, EXTRAX, EXTRAX,
3  KXTRAX, KXTRAX, KXTRAX, KXTRAX,
4  COMMON /CINNER/ ITMIX, IG, IZ, JUP,
1  XITGG, XITGGA, LG, NEGRPT, JZ, KK

THE DIMENSION STATEMENTS
DIMENSION MA(1), V(1), C(1)
DIMENSION CT(1), CGG(1), CGGA(1)
DIMENSION TVLK(1), BUCK(1), BUCK(1)

THE INTEGER FUNCTIONS
INDEX(LENGTH, INDEXO, INDEXL) = LENGTH*(INDEXO-1)+INDEXL

136
TO COMPUTE THE TOTAL CROSS SECTION AND ANY TRANSVERSE LEAKAGE CROSS SECTION FOR A GIVEN GROUP IG AND A GIVEN Z MESH INTERVAL IZ.

1000 DO 1175 I=1, NR
   IJ=INDEX(NR, J, I)
   K=IABS(MA(IJ))
   KA=K
   IF (KSISO) 1015, 1015, 1005
   1005 IF (MA(IJ)) 1010, 1015, 1015
   1010 KA=K+1

        TO COMPUTE THE TOTAL CROSS SECTION AND ANY TRANSVERSE LEAKAGE
        CROSS SECTION FOR A GIVEN GROUP IG AND A GIVEN Z MESH
        INTERVAL IZ.

1015 K1=INDEX(NTYPs, IG, KCTR)
   CL=0.0
   K2=INDEX(NTYNG, K, K1)
   CE=C(K2)
   IF (NBF-I) 1050, 1050, 1020
1020 K2=INDEX(NTYNG, KA, K1)
   CTR=C(K2)
   TEMP=ABS(CTR)
   CL=(IBF/(TEMP*HY+1.42089216)**2)*CTR
   IF (HY-O.1E-7) 1030, 1030, 1025
1025 CL=CL+(IBF/TEMP_HY_42089216_)**2)*CTR
1030 TVLK(IJ)=CL*VI(IJ)
1050 CE=CE+CL
   IF (LBUCK) 1052, 1052, 1051
1051 CE=CE+BUCK(IIG)
1052 IF (NBUCK) 1054, 1054, 1053
1053 K2=INDEX(NTYNG, K, IG)
   CE=CE+BUCKG(K2)
1054 CV(IJ)=CE*VI(IJ)
   IF (K1 1106) 1106, 1100, 1175

        TO STORE THE WITHIN GROUP SCATTERING CROSS SECTIONS BY VOLUME
        MESH INTERVALS.

1100 K1=INDEX(NTYPs, IG, KC GG)
   K2=INDEX(NTYNG, K, K1)
   CGG(I)=C(K2)*VI(IJ)
   XITGG=XITGG+CGG(I)
   IF (KSISO) 1175, 1175, 1125
1125 IF (KA-KJ 1130, 1130, 1139
1130 CGGA(I)=0.0
   GO TO 1175
1135 K2=INDEX(NTYNG, KA, K1)
   CGGA(I)=C(K2)*VI(IJ)
   XITGGA=XITGGA+CGGA(I)

        CONTINUE

1175 CONTINUE

        RETURN

END
THE COMMON STATEMENTS
COMMON X
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED, KBC, KLBC, KBBC, KBCC, KTBC, KLBC,
1 KREG, KALC, KGO, KBC, KBCC, KBCC, KBCC, KBCC,
2 KRBC, KBCC, KBCC, KBCC, KBCC, KBCC, KBCC, KBCC,
3 NR, NTYP, NMAT, LBUCK, KREAD, KFLUX, KCTYP,
4 KSISU, KGST, KCTR, KGG, NG, KZERO, KZERO,
5 KACCEL, KALC1, NFN, NZN, KZERO, KZERO,
6 KBCOUP, KACAV, KGAVE, KNIP, KEDIT, KEDIT,
7 RYF, TIMAX, CALC1, OMEG, EPS, XNF, OMEGAP
COMMON /CALL3/
1 HZ, HY, BF, KCA, KCF,
2 KUNV, KFINSH, KASOR, KUPS, NR1, NDNZG,
3 NBE, NO, NGPL, NZPL, NDR, NDNJ,
4 NCC, NCC, NCC, NCC, NCC, NCC,
5 ITMA, ITMI, ITMIST, ITMIX, ITD,
6 ITMA, ITMI, ITMIST, ITMIX, ITD,
7 TIMEX, TIMEZ, SDGF, OMEX, SCALUP
COMMON /CALL4/
1 HZ, HY, BF, KCA, KCF,
2 KUNV, KFINSH, KASOR, KUPS, NR1, NDNZG,
3 NBE, NO, NGPL, NZPL, NDR, NDNJ,
4 NCC, NCC, NCC, NCC, NCC, NCC,
5 ITMA, ITMI, ITMIST, ITMIX, ITD,
6 ITMA, ITMI, ITMIST, ITMIX, ITD,
7 TIMEX, TIMEZ, SDGF, OMEX, SCALUP
COMMON /COUTER/
1 KNI, KNO, K180, KWHITE, KNOR,
2 KNUKUP, KNR, KNJ, KJ1,
3 KNZ
COMMON /CINPT1/
1 KAS, LAG, LSN, LNS, LSN, LSN, LSN,
2 LSC, LSC, LSN, LSN, LSN, LSN, LSN,
3 LNB, LNB, LNB, LNS, LNS, LNS, LNS,
4 LAC, LAZ, LRM, LRM, LRM, LRM, LRM,
5 LGT, LGG, LGS, LGS, LGS, LGS, LGS,
6 LAPS, LAR, LAR, LAR, LAR, LAR, LAR,
7 LNM, LTVK, LTVK, LTVK, LTVK, LTVK, LTVK,
8 LMR, LMF, LMF, LMF, LMF, LMF, LMF,
9 LCMN /CINPT2/
1 LNA, LJR, LJS, LJS, LJS, LJS, LJS,
2 LJR, LJR, LJS, LJS, LJS, LJS, LJS,
3 LNJ, LNU, LNS, LNS, LNS, LNS, LNS,
4 LJR, LJR, LJS, LJS, LJS, LJS, LJS,
THE DIMENSION STATEMENTS
DIMENSION X(17500)

THE FORMAT STATEMENTS
FORMAT (I=80,END OF BCDUMP END OF BCDUMP END OF BCDUMP END OF BCDUMP)

THE INTEGER FUNCTIONS
INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL

980 KDUP=1
   IF (KBCDUP+1) 990, 1036, 1000
990 KDUP=3
   KBCDUP=-KBCDUP
   GO TO 1020

1000 IF (KBCDUP-1) 1100, 1065, 1015
1005 IF (KCONV=1) 1010, 1015, 1010
1010 IF (KCONV+1) 1100, 1015, 1000
1015 KDUP=2
1020 CALL BCDUMP (KREG, LAST)
   CALL BCDUMP (X(1), X(LBCDUP))

1030 ITEMP1=LN-1
   ITEMP2=LJR-1
   ITEMP3=LJZ-1
   DO 1040 IG=1, NG
      K1=ITEMP1*INDEX(NIJ, IG, 1)
      K2=ITEMP1*INDEX(NIJ, IG, NIIJ)
      CALL BCDUMP (X(K1), X(K2))
      GO TO (1040, 1035, 1035), KDUP
1035 IF (KISO) 1040, 1040, 1036
1036 K1=ITEMP2*INDEX(NIJ, IG, 1)
      K2=ITEMP2*INDEX(NIJ, IG, NIIJ)
      CALL BCDUMP (X(K1), X(K2))
      IF (NL-1) 1040, 1040, 1037
1037 K1=ITEMP3*INDEX(NIJ, IG, 1)
      K2=ITEMP3*INDEX(NIJ, IG, NIIJ)
      CALL BCDUMP (X(K1), X(K2))
1040 CONTINUE
   GO TO (1041, 1045, 1045), KDUP
1041 WRITE (6, 130)
   GO TO 1100
1045 GO TO (1041, 1041, 1050), KDUP
1050 WRITE (6, 130)
   LNP=LF-1
CALL BCDUMP (X(LV), X(LNP))
WRITE (6, 130)
C
C
1100 RETURN
C
END

$1BMAP BCDUMP 100, M94, DECK
7094 RELMOD ASSEMBLY.

$1BLDR BCDUMP
$TEXT BCDUMP
BCDUMP ROUTINE FOR IBSYS

ENTRY BCDUMP

<table>
<thead>
<tr>
<th>CARD ID</th>
<th>BCDU0002</th>
<th>CARD ID</th>
<th>BCDU0003</th>
<th>CARD ID</th>
<th>BCDU0004</th>
</tr>
</thead>
<tbody>
<tr>
<td>000001</td>
<td>000000 G  000005 100012</td>
<td>00001</td>
<td>000000 G  000005 100012</td>
<td>000001</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>000002</td>
<td>000000 G  000005 100012</td>
<td>00002</td>
<td>000000 G  000005 100012</td>
<td>00003</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>000003</td>
<td>000000 G  000005 100012</td>
<td>00004</td>
<td>000000 G  000005 100012</td>
<td>00005</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>000006</td>
<td>000000 G  000005 100012</td>
<td>00007</td>
<td>000000 G  000005 100012</td>
<td>00008</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00009</td>
<td>000000 G  000005 100012</td>
<td>00010</td>
<td>000000 G  000005 100012</td>
<td>00011</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00012</td>
<td>000000 G  000005 100012</td>
<td>00013</td>
<td>000000 G  000005 100012</td>
<td>00014</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00015</td>
<td>000000 G  000005 100012</td>
<td>00016</td>
<td>000000 G  000005 100012</td>
<td>00017</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00018</td>
<td>000000 G  000005 100012</td>
<td>00019</td>
<td>000000 G  000005 100012</td>
<td>00020</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00021</td>
<td>000000 G  000005 100012</td>
<td>00022</td>
<td>000000 G  000005 100012</td>
<td>00023</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00024</td>
<td>000000 G  000005 100012</td>
<td>00025</td>
<td>000000 G  000005 100012</td>
<td>00026</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00027</td>
<td>000000 G  000005 100012</td>
<td>00028</td>
<td>000000 G  000005 100012</td>
<td>00029</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00030</td>
<td>000000 G  000005 100012</td>
<td>00031</td>
<td>000000 G  000005 100012</td>
<td>00032</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00033</td>
<td>000000 G  000005 100012</td>
<td>00034</td>
<td>000000 G  000005 100012</td>
<td>00035</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00036</td>
<td>000000 G  000005 100012</td>
<td>00037</td>
<td>000000 G  000005 100012</td>
<td>00038</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00039</td>
<td>000000 G  000005 100012</td>
<td>00040</td>
<td>000000 G  000005 100012</td>
<td>00041</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00042</td>
<td>000000 G  000005 100012</td>
<td>00043</td>
<td>000000 G  000005 100012</td>
<td>00044</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00045</td>
<td>000000 G  000005 100012</td>
<td>00046</td>
<td>000000 G  000005 100012</td>
<td>00047</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00048</td>
<td>000000 G  000005 100012</td>
<td>00049</td>
<td>000000 G  000005 100012</td>
<td>00050</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00051</td>
<td>000000 G  000005 100012</td>
<td>00052</td>
<td>000000 G  000005 100012</td>
<td>00053</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00054</td>
<td>000000 G  000005 100012</td>
<td>00055</td>
<td>000000 G  000005 100012</td>
<td>00056</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00057</td>
<td>000000 G  000005 100012</td>
<td>00058</td>
<td>000000 G  000005 100012</td>
<td>00059</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00060</td>
<td>000000 G  000005 100012</td>
<td>00061</td>
<td>000000 G  000005 100012</td>
<td>00062</td>
<td>000000 G  000005 100012</td>
</tr>
<tr>
<td>00063</td>
<td>000000 G  000005 100012</td>
<td>00064</td>
<td>000000 G  000005 100012</td>
<td>00065</td>
<td>000000 G  000005 100012</td>
</tr>
</tbody>
</table>

**PICK UP FIRST ARGUMENT**

**PICK UP SECOND ARGUMENT**

**WOI HAS THE FIRST ADDRESS**

**FIRST LOCATION IN INDEX 1**

**THE NO. OF WORDS OUTPUTED IN INDEX 2**

**TRUE WORD COUNT**

**ONLY 22 WORDS OR LESS LEFT**

**CLEAR THE BUFFER**
BINARY CARD ID. BCDUO006

00116 2 00001 1 00101 1001 00174 TIX **-1.1,1
00117 U6C2 00 0 00145 1001 SLW CKSUM
00120 U74C 00 4 00000 1001 RITE TSX WRITE,4
00121 0 00134 0 00020 1010 PZE **.EDF
00122 0 CCC34 0 00144 1001 OCD WD1,.28
00123 UO2C 00 0 00011 1001 TRA TXL
00124 U02D 00 0 00001 1001 RETURN TRA ++1
00125 U74C 00 1 00041 1001 AXT TXL.1
00126 U634 00 1 00123 1001 SAXA RETURN-1.1
81 U0127 U634 00 0 00066 1001 SAXA CNUM
00130 RETURN BCDUMP
00131 U560 00 0 00124 1001 LASTC CLA RETURN
00132 U601 00 0 00123 1001 STO RETURN-1
00133 U602 00 0 00047 1001 TRA TEST4
00134 UCCUUUUG 00010 EDF CALL .FXEM.(EDF2)
00135 U1 00010 0 01003 1001

BINARY CARD ID. BCDUO007

U0136 G CL204 0 00140 10100
U0137 U0000 0 00143 10001
00140 GLLCCUGUUG 00010 CALL EXIT
00140 0C74 00 05000 10011
00141 1 C0C0 0 0102 10011
00142 0 0C204 0 00141 10100
00143 6 0C0C0 0 00042 10000 EDF2 PZE 34
00144 0 00000 0 00000 10000 WDI PZE
00145 2C0C0000027 00001 CKSUM BSS 23
00174 2C0C4004040 10000 GP OCT 420041004040
00175 10C20400000 10000 OCT 104020400000
00176 0 00000 0 00000 10000 WORD3 PZE
00177 0 00000 0 00000 10000 PZE
00200 000000002000 10000 HUNBIT OCT 2000
00201 000C20000000 10000 BITU OCT 20000000
00202 200000000000 10000 BITT OCT 200000000000
00203 0 00000 0 11000 10011 OUT PZE *PCH.
00204 000000000000 10000 *LDIR
00205 222324444447 10000

BINARY CARD ID. BCDU0008 00000 01111 END

CONTROL DICTIONARY
$CUICT BCDUMP

BINARY CARD ID. BCDU0009
000C2060000000 000C2060000000
000C2060000000 222324644447
000C2060000000 222324644447
000C2060000000 222324644447
000C2060000000 060000000000
060000000000 3326525433
3326525433 2C0000000000
2C0000000000 256731636060
256731636060 2C0000100000
2C0000100000 627062364623
627062364623 2C0000100000
2C0000100000 334647254560
334647254560 2C0000100000
2C0000100000 336551316325
336551316325 2C0000100000
2C0000100000 336723363360
336723363360 2C0000100000
2C0000100000

$DEND BCDUMP 000751 BCDU0011

143
### Symbol Reference Data

#### References to Defined Symbols.

<table>
<thead>
<tr>
<th>Class</th>
<th>Symbol</th>
<th>Value</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCDUMP</td>
<td>0000U</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>BITT</td>
<td>00202</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>BITU</td>
<td>00201</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>CKSUM</td>
<td>00145</td>
<td>57, 63, 117</td>
<td></td>
</tr>
<tr>
<td>CLA</td>
<td>00062</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>CLEAR</td>
<td>00057</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNUM</td>
<td>00066</td>
<td>106, 112, 127</td>
<td></td>
</tr>
<tr>
<td>EOF</td>
<td>00143</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td>EOF2</td>
<td>00134</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>UG</td>
<td>00174</td>
<td>73, 115</td>
<td></td>
</tr>
<tr>
<td>MUNBIT</td>
<td>00200</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>IX1</td>
<td>00041</td>
<td>23, 55, 123, 125</td>
<td></td>
</tr>
<tr>
<td>IX2</td>
<td>00042</td>
<td>24, 45</td>
<td></td>
</tr>
<tr>
<td>.0001</td>
<td>00003</td>
<td>7, 10, 11</td>
<td></td>
</tr>
<tr>
<td>.0002</td>
<td>00004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.0003</td>
<td>00005</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LASTC</td>
<td>00131</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>LOOP</td>
<td>00054</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUN</td>
<td>00040</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>OUT</td>
<td>00203</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>RETURN</td>
<td>00124</td>
<td>126, 131, 132</td>
<td></td>
</tr>
<tr>
<td>RITE</td>
<td>00120</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>LCTR</td>
<td>BLCIR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUAL</td>
<td>UNGS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCTR</td>
<td>//</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST4</td>
<td>00047</td>
<td>133</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>00043</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WDI3</td>
<td>00176</td>
<td>100, 105</td>
<td></td>
</tr>
<tr>
<td>WDI4</td>
<td>00144</td>
<td>16, 17, 20, 50, 53, 114, 122</td>
<td></td>
</tr>
<tr>
<td>LCTR</td>
<td>//</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXIT</td>
<td>5</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>.FXEM.</td>
<td>4</td>
<td>134</td>
<td></td>
</tr>
<tr>
<td>.UPEN</td>
<td>7</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>.PCH.</td>
<td>9</td>
<td>203</td>
<td></td>
</tr>
<tr>
<td>.RIE.</td>
<td>8</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>SYSLUC</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

#### References to Virtual Symbols.

- EXIT 5 140
- .FXEM. 4 134
- .UPEN 7 37
- .PCH. 9 203
- .RIE. 8 120
- SYSLUC 6 5

$1BMAP .PCH. 6 .DECK

7094 KELMU ASSEMBLY.

$1EELD .PCH. 02/25/66 .PCH0000

$FILE .PCH. *PCH *PP*READY*OUTPUT,BLK=28,MULTIKEEL,BIN,NOLID .PCH0001
FILE DICTINARY

BUILD *PCH.

BINARY CARD ID ..PCMP003
                2500200003 PCH FILE *PCH
                BINARY OUTPUT, NOCVN, BLK=28
                0000000000
                7233666600
                6600666600
                6600000000
                $TEXT *PCH.

ENTRY *PCH.

BINARY CARD ID ..PCMP005
                0000000000 0000000000 0000000000
                FILE *PCH PJE PCH FILE *PCH
                PREPARED, OUTPUT, BLK=28, MULTIREL, BINARY

END

CONTROL DICTINARY

$END *PCH.

BINARY CARD ID ..PCMP007
                0000000000
                0000000000
                1547233333360
                0000000000
                3247233333360
                0000000000
                $END *PCH.

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS SYMBOL VALUE REFERENCES

        *PCH.  0000 GEFILE PCH 1 0
        LCTR  BLCTR WITH UNGS
        LCTR  //
SUBROUTINE OUTPUT IS THE OVERLAY (OR CHAIN) SUBROUTINE OF TDSN
THAT WRITES OUT THE FINAL VALUES OF THE ITERATED QUANTITIES
AND COMPUTES AND WRITES OUT EXTRA OUTPUT BASED ON THE
FINAL ITERATED VALUES.

THE COMMON STATEMENTS
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED, KITBC, KLBC,
1 KREG, KALC, KGEO, KBBC, KTBC, KCTYP,
2 KRBC, NNTPS, LBUCK, NLBC, KCTYP,
3 NR, NTS, KCR, KCCG, KREAD, KFLUXI,
4 KSISO, KGST, KCTR, KCGG, KFLUXI,
5 KACCEL, KALC1, NFN, M2N, XIMAX,
6 KBCDUP, KACAV, KAVE, KNIP, KEDIT,
7 KRF, TMAX, EPS, XZF
COMMON /CALL3/
1 HZ, HY, BF,
2 KCA, KCF,
3 KNG, KFINN, KASOR, KUPS,
4 NBF, NGP1, NFP1, NRP1,
5 N1, NTLY, NDNZ, NDNR, NDNIJ, NZDENZ,
6 NNXS, NAR, NC, NIJG, MBUC,
7 TIMA, TIMI, TIMIST, TIMIGX, ITD,
8 ITMC, NEGSOR, LZSAV,
9 TIMEX, TIMEZ, SDFG1, SCALUP
COMMON /CALL4/ KVEC, KFLUX, EOMEG, EPG, EPGX, EPNG, EPNGM,
1 EXRA1, EXRA2, EXRA3, EXRA4, EXRA5,
2 KXTRA1, KXTRA2, KXTRA3, KXTRA4, KXTRA5

THE DIMENSION STATEMENTS
DIMENSION MA(I)
DIMENSION FG(I), F(I)
DIMENSION XN(I), XJR(I), XJZ(I)
DIMENSION V(I)
DIMENSION KFIN(I), KASOR(I)
DIMENSION AG(I), FNG(I),
1 SG(I), ASG(I), SCG(I),
2 SDG(I), SING(I), XNG(I),
3 XJRG(I), XJZG(I), XNB(I),
4 HNLG(I), VNLG(I), XNLG(I),
5 BNLG(I)
DIMENSION ES1JM(1), BUCKL(1), MXVARI(1), MXSTATE(1)
DIMENSION STG(1)
DIMENSION XNORMII(1)

THE FORMAT STATEMENTS
100 FORMAT (1I1)
103 FORMAT (1H )
111 FORMAT (8E14.6)
113 FORMAT (8E14.6)
120 FORMAT (126H ITMA ITMI ITMIX ITD TIME REQ EPG)
121 FORMAT (I1, I6, 14, IZ, 3E14.6, 14, I4)
122 FORMAT (I1, I6, 14, IZ, 3E14.6, 14, I4)
123 FORMAT (I1, 28HTOTAL NEUTRON FLUX (XNJG(1G)))
124 FORMAT (I1, 23HFIISSION SOURCE (FG(1G)))
125 FORMAT (I1, 31H SCATTERING SOURCE (SING(1G)))
126 FORMAT (I1, 29H OUT SCATTERING LOS (SDG(1G)))
127 FORMAT (I1, 32H SELF SCATTERING SOURCE (SG(1G)))
128 FORMAT (I1, 35H NON-SELF SCATTERING SOURCE (SG(1G)))
129 FORMAT (I1, 22HTOTAL SOURCE (STG(1G)))
130 FORMAT (I1, 26HFIISSION NEUTRONS (FNG(1G)))
131 FORMAT (I1, 20H ABSORPTIONS (AG(1G)))
132 FORMAT (I1, 29H HORIZONTAL LEAKAGE (MNLLG(1G)))
133 FORMAT (I1, 27H VERTICAL LEAKAGE (VNLG(1G)))
134 FORMAT (I1, 22H NET LEAKAGE (XMLG(1G)))
135 FORMAT (I1, 42HN3T LEAKAGE IN (XNB(1G)) (SOURCE - LOSS))
136 FORMAT (I1, 48HTotal FIRST DIRECTION NEUTRON CURRENT (XJRG(1G)))
137 FORMAT (I1, 74HA NON-SL3T LEAKAGE SOURCE WITHOUT DIRECTIO III COSINES (ASG(1G)))
138 FORMAT (I1, 2X, 46HTHE FISSION RATE PER UNIT VOLUME (FI(1G)))
139 FORMAT (I1, 49HTotal SECOND DIRECTION NEUTRON CURRENT (XJZG(1G)))
140 FORMAT (I1, 2X, 66HTHE NET FIRST DIRECTION CURRENT PER UNIT VOLUME
141 BY GROUP (XJRG(1G)))
142 FORMAT (I1, 2X, 66HTHE NET SECOND DIRECTION CURRENT PER UNIT VOLUME
1 BY GROUP (XJZ(1G)))
143 FORMAT (I1, 2X, 58HTHE NON-DIRECTIONAL FLUX PER UNIT VOLUME BY GROUP
1 P (XN(1G)))
144 FORMAT (I1, 2X, 38H THE NUMBER OF MESH INTERVALS AT WHICH A PI SOURCE C
1 ORRECTION WAS MADE (KASORG(1G)))
145 FORMAT (I1, 51H THE MAXIMUM RATE OF CHANGE IN THE SELF-SCATTERING SOURCES
1 OCCURS AT THE MESH INTERVAL (MXVARI(1G)))
146 FORMAT (I1, 25HBUCKLING LOS (BUCKL(1G)))
147 FORMAT (I1, 24HBUCKLING LOSS (BNL(1G)))
148 FORMAT (I1, 68HTHE MAXIMUM RATE OF CHANGE IN THE SELF-SCATTERING SOURCES
1 OCCURS AT (ESIJM(1G)))
149 FORMAT (I1, 43H AVERAGE OVER ENTIRE VOLUME)
150 FORMAT (I1, 43H AVERAGE PARALLEL OVER ENTIRE VOLUME)
175 FORMAT (1HS*,43HAVERAGE PARALLEL (D*BSW) OVER ENTIRE VOLUME)
C
C THE INTEGER FUNCTIONS
INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C 1000 WRITE (6,100)
WRITE (6,120)
WRITE (6,121) ITMA,ITIM1,ITMI,GX,ITD,TIMEZ,TIMEX,EPGM,KASOR,KONV,
1 KFINSh,XNB(NGP1),Glam,Eigen
IF (KFINSh) 1010,1015,1010
1010 WRITE (6,122)
WRITE (6,111) (KFINI(GI), IG=1,NG)
1015 IF (KASOR) 1030,1036,1020
1020 WRITE (6,147)
WRITE (6,111) (KASORI(GI), IG=1,NG)
1030 WRITE (6,149)
WRITE (6,111) (MXVARI(GI), IG=1,NG)
WRITE (6,152)
WRITE (6,113) (ESIJM(I), IG=1,NG)
C
C 1050 SG(NGP1)=0.0
FNGSUM=0.0
IF (KSISO) 1052,1052,1051
1051 ASGSUM=0.0
XJRGSM=0.0
XJZGSM=0.0
C
1052 DO 1056 IG=1,NG
IF (KSISO) 1055,1055,1053
1053 XJRG(I)=6.0
IF (NZ-1) 1056,1058,1057
1057 XJZG(I)=6.0
1058 IJG=INDEX(NI,J,IG,0)
DO 1054 J=1,NIJ
IJG=IJG+1
XJRG(I)=XJRG(I)+V(IJ)*XJR(IJG)
IF (NZ-1) 1054,1059,1061
1059 XJZG(I)=XJZG(I)+V(IJ)*XJZ(IJG)
1054 CONTINUE
ASGSUM=ASGSUM+ASGI(IG)
XJRGSM=XJRGSM+XJRG(I)
XJZGSM=XJZGSM+XJZ(I)
IF (NZ-1) 1056,1058,1061
1061 XJRGSM=XJRGSM+XJRG(I)
1055 SG(NGP1)=SG(NGP1)+SG(I)
1056 FNGSUM=FNGSUM+FGN(I)
DO 1060 IG=1,NGP1
1060 SG(I)=SG(I)+SCG(I)
IF (LBUCK) 1070,1070,1068
1068 BUCKS=0.0
DO 1069 IG=1,NG
BUCK(I)=BUCK(I)+XNG(IG)
1069 BUCKS=BUCKS+BUCK(I)
C
1070 WRITE (6,123)
WRITE (6,113) (XNG(IG), IG=1,NGP1)
IF (KSISO) 1080,1080,1075
1075 WRITE (6,136)
WRITE (6,113) (XJRG(IG), IG=1,NG), XJRGSM
IF (NZ-1) 1080,1080,1076
1076 WRITE (6,141)
WRITE (6,113) (XJZG(IG), IG=1,NG), XJZGSM
1080 WRITE (6,124)
WRITE (6,113) (FG(IG), IG=1,NGP1)
WRITE (6,125)
WRITE (6,113) (SG(IG), IG=1,NGP1)
WRITE (6,126)
WRITE (6,113) (SDG(IG), IG=1,NGP1)
WRITE (6,127)
WRITE (6,113) (SCG(IG), IG=1,NGP1)
WRITE (6,128)
WRITE (6,113) (SG1(IG), IG=1,NGP1)
WRITE (6,129)
WRITE (6,113) (STG(IG), IG=1,NGP1)
IF (KSISO) 1090,1090,1085
1085 WRITE (6,138)
WRITE (6,113) (ASG(IG), IG=1,NG), ASGSM
1090 WRITE (6,130)
WRITE (6,113) (FNG(IG), IG=1,NG), FNGSM
WRITE (6,131)
WRITE (6,113) (AG(IG), IG=1,NGP1)
WRITE (6,132)
WRITE (6,113) (MNLG(IG), IG=1,NGP1)
IF (NZ-1) 1092,1092,1091
1091 WRITE (6,133)
WRITE (6,113) (WNLG(IG), IG=1,NGP1)
1092 IF (NBF-1) 1093,1093,1094
1093 IF (NBUCK) 1055,1095,1094
1094 WRITE (6,151)
WRITE (6,113) (BNLG(IG), IG=1,NGP1)
1095 IF (LBUCK) 1097,1097,1096
1096 WRITE (6,150)
WRITE (6,113) (BUCLK(IG), IG=1,NG), BUCLKS
1097 WRITE (6,134)
WRITE (6,113) (XNLG(IG), IG=1,NGP1)
WRITE (6,135)
WRITE (6,113) (XNB(IG), IG=1,NGP1)
C
C
1100 XNMAX=0.0
00 1105 JG=1,NIJG
IF (XNMAX-XN(JG)) 1101,1105,1105
1101 XNMAX=XN(JG)
1105 CONTINUE
1106 IF (XNMAX .LT. 1106) Go To 1110
1107 YNORM=1.0/XNMAX
1110 WRITE (6,148) YNORM
C
C
125 DO 1130 J=1,NIJ
1130 XNORM(IJ)=YNORM*F(IJ)
    WRITE (6,140)
    DO 1135 J=1,NZ
    WRITE (6,165) J
    K1=INDEX(NR,J,1)
    K2=INDEX(NR,J,NR)
    WRITE (6,113) (XNORM(IJ), IJ=K1,K2)
1135 WRITE (6,103)
C
C
1150 WRITE (6,145)
    IJG=0
    DO 1160 IJG=1,NG
    IJG=IJG+1
1155 XNORM(IJ)=YNORM*XN(IJG)
    WRITE (6,146) IG
    DO 1160 J=1,NZ
    WRITE (6,165) J
    K1=INDEX(NR,J,1)
    K2=INDEX(NR,J,NR)
    WRITE (6,113) (XNORM(IJ), IJ=K1,K2)
1160 WRITE (6,103)
C
C
1165 WRITE (6,143)
    IJG=0
    DO 1170 IJG=1,NG
    IJG=IJG+1
1166 XNORM(IJ)=YNORM*XJR(IJG)
    WRITE (6,146) IG
    DO 1170 J=1,NZ
    WRITE (6,165) J
    K1=INDEX(NR,J,1)
    K2=INDEX(NR,J,NR)
    WRITE (6,113) (XNORM(IJ), IJ=K1,K2)
1170 WRITE (6,103)
C
C
1171 WRITE (6,144)
IJK=0
DO 1173 IG=1,NG
DO 1172 IJ=1,NIJ
IJK=IJK+1
1172 XNORM(IJ)=YNORM*XJZ(IJG)
WRITE (6,146) IG
DO 1173 J=1,NJ
WRITE (6,165) J
K1=INDEX(NR,J,1)
K2=INDEX(NR,J, NR)
WRITE (6,113) (XNORM(IJ), IJ=K1,K2)
1173 WRITE (6,103)
C
C
C
1200 IF (KKNIP) L300,1300,1205
1205 WRITE (6,171)
WRITE (6,172)
DO 1280 N=1,NMAT
DO 1270 IG=1,NG
STG(IJ)=0.0
IJK=INDEX(NIJ,IG,0)
DO 1260 IJ=1,NIJ
IJK=IJK+1
IF (N-IABS(MA(IJJ))) 1225,1245,1225
1225 IF (MA(IJJ)) 1230,1260,1260
1230 IF (N+(MA IJJ))1260,1235,1260
1235 TEMP=V(IJJ)*XJZ(IJG)
IF (NZ-1) 1250,1250,1240
1240 TEMP=TEMP+V(IJJ)*XJZ(IJG)
GO TO 1250
1245 TEMP=V(IJJ)*XN(IJG)
1250 STG(IJ)=STG(IJ)+TEMP
1260 CONTINUE
1270 CONTINUE
WRITE (6,113) (STG(GI), IG=1,NG)
WRITE (6,170) (STG(IG), IG=1,NG)
1280 CONTINUE
C
IF (KKNIP-2) L300,1285,1300
1285 WRITE (6,173)
DO 1290 IG=1,NG
1290 STG(IG)=VNLG(IG)/XNG(IG)
WRITE (6,113) (STG(IG), IG=1,NG)
C
1300 IF (LBUCK) L305,1400,1305
1305 WRITE (6,174)
WRITE (6,175)
DO 1310 IG=1,NG
1310 STG(IG)=HNLG(IG)/XNG(IG)
WRITE (6,113) (STG(IG), IG=1,NG)
WRITE (6,170) (STG(IG), IG=1,NG)
C
C
1400 IF (KACAV) L405,1401,1405
1401 IF (KGAVE) L410,1410,1405
1405 KCHAIN=4
RETURN
C
1410 IF (KEDIT) L420,1420,1415
1415 KCHAIN=5
RETURN
C
1420 KCHAIN=1
RETURN
C
END
**$IBFIC TABLES DECK**

**SUBROUTINE TABLES** (MA,C, CN2N, XN,XJR,XJZ, V,  
1 NMRA,NMZA,IDM,IDMAP,IDGP)

**SUBROUTINE TABLES** IS THE OVERLAY (OR CHAIN) SUBROUTINE OF TOSN  
THAT COMPUTES AND PRINTS OUT THE ACTIVITY AND AVERAGING TABLES.

**THE COMMON STATEMENTS**

| COMMON          | /CALL1/ KCHAIN, KEND, KEFN |
| COMMON          | /CALL2/ KBCRED, KBCRED   |
| KREG           | KALC, KGOE, KBGC, KTBC, KLBC |
| KRUC           | NSN, NG, NZ, KCTYP, KFLUX, |
| NR             | NTYPS, NMAT, LBUCK, KDRAD, |
| KSISO          | KGST, KCTR, KGGE, KDREAD, |
| KACCEL         | ITMIMX, ITMIG, KZONE, KZONEZ,  |
| KBCoup         | KACAL, KIN, N2N, KMR,  |
| XN             | XJR, XJZ, XMK, XMB, XMB  |
| V              | 152

**THE DIMENSION STATEMENTS**

| DIMENSION MA(1), C(1) |
| DIMENSION CN2N(1) |
| DIMENSION NMX(1), XJR(1), XJZ(1) |
| DIMENSION V(1) |
| DIMENSION NMRA(1), NMZA(1), IDM(1) |
| DIMENSION IDMAP(1), IDGP(1) |
| DIMENSION XTAB(5000) |

**THE FORMAT STATEMENTS**

| 104 FORMAT (1HL,21H,PROGRAM STOP AT KEFN=,18) |
| 110 FORMAT (71110) |
120 FORMAT (1HL,36HTABLES ARRAY SIZE EXCEEDED. LASTAB=.18,31M WITH 5
1000 STORAGE AVAILABLE.)
160 FORMAT (1H1,16HACTIVITY TABLES)
161 FORMAT (1H1,17HAVERAGING TABLES)

C
C
C   NZM1=NZ-1
   NTYG1=NTYPS*NGP1
1000 IF (KACAV) 1005,1001,1005
1001 IF (KGAVE) 2000,2000,1005
1005 KUP=KCNG-(KCTR+1)
   KDBW=NTYPS-KCGG
   IF (KGAVE) 1100,1100,1010
1010 READ (9,110) (IDGP(IG), IG=1,NG)
   KDBW=KGAVE-1
   IF (KDBW=KDBAR) 1015,1020,1020
1015 KDBAR=KDBW
1020 NTYBAR=(KCTR+1)+KDBAR
   KUPBAR=0
   IF (KUP) 1075,1075,1025
1025 LBAR=NGP1-KUP
   IGB=1
   IG=1
1030 IF (LBAR-IG) 1035,1035,1040
1035 KUPBAR=(KGAVE+1)-IGB
   GO TO 1075
1040 IG=IG+1
   IF (NG-IG) 1075,1075,1045
1045 IF (IDGP(IG)-IGB) 1040,1040,1050
1050 IGB=IGB+1
   GO TO 1030
1075 NTYBAR=NTYBAR+KUPBAR
   KGBAR=(KCTR+1)+KUPBAR
C
C
C   COMPUTE THE ACTIVITY TABLES.
C
C
C1100 IF (KACAV) 1120,1502,1105
1105 IF (KACAV-2) 1110,1501,1110
1110 WRITE (6,160)
   DU 1115 IJ=1,NIJ
1115 IDMAP1(IJ)=MA(IJ)
   NACAV=NMAT
   GO TO 1130
1120 IF (KACAV+2) 1125,1501,1125
1125 WRITE (6,160)
   CALL IDACAV (NR,NZ,NACAV,NZONRA,NZONZA, NMRA(1),NMZA(1),IDM(1),
   1 IDMAP(1))
C
1130 NACAV1=NACAV+1
   NACAVT=NACAV1
   IF (KSISO) 1150,1150,1135
1135 NACAV2=NACAV+2
   NACAVT=NACAV2

153
C
1150 LKSI=1
  LVSUM=LKSI*NACAVT
  LFX=LVSUM*NACAVT
  LCX=LFX*NACAVT*NGP1
  LCOUT=LCX*NACAVT*NGP1*NTYPS
  LGISAD=LCOUT+1
  LFXBAR=LGISAD+NG
  LCXBAR=LFXBAR+1
  LGEOBAR=LCXBAR+1
  LASTAB=LGEOBAR
  IF (LASTAB=5000) 1200, 1200, 1155
1155 WRITE (6, 120) LASTAB
  KEND=1
  KEFN=1155
  GO TO 2050
C
1200 XTALE=1
  CALL TABLE (MA, CN2N, XN, XJR, XJJ, V, IDMAP, IDGP, 1)
  XTAB(LKSI), XTAB(LVSUM), XTAB(LFX), XTAB(LCX), XTAB(LCOUT), 2
  XTAB(LGISAD), XTAB(LFXBAR), XTAB(LCXBAR), XTAB(LGEOBAR)
  IF (KEND) 1500, 1500, 2050
C
  COMPUTE THE AVERAGING TABLES.
C
1500 IF (KREG) 1501, 1501, 2000
1501 IF (IABS(KACAV)-1) 1503, 1503, 1502
1502 WRITE (6, 161)
  GO TO 1505
1503 IF (KGAVE) 2000, 2000, 1505
1505 IF (KACAV-2) 1506, 1510, 1510
1506 IF (IABS(KACAV)-1) 1510, 1510, 1520
1510 DO 1515 1J=1, NIJ
1515 IDMAP(IJ)=MA(IJ)
  NACAV=NMAT
  GO TO 1530
1520 CALL DACAV (NR, NZ, NACAV, NZONRA, NZONZA, NMRA(I), NMZA(I), 1)
  IDM(1)
C
1530 NACAV1=NACAV+1
  NACAVT=NACAV1
  IF (KSI0) 1550, 1550, 1535
1535 NACAV2=NACAV+2
  NACAVT=NACAV2
C
1550 LKSI=1
  LVSUM=LKSI*NACAVT
  LFX=LVSUM*1
  LCX=LFX*NACAVT*NGP1
  LCOUT=LCX*NACAVT*NGP1*NTYPS
  LGISAD=LCOUT+NACAVT*NGP1
  LFXBAR=LGISAD+1
  LCXBAR=LFXBAR*NTYBAR*KGAVE
  LGEOBAR=LCXBAR*NTYBAR*KGAVE

154
LASTAB=LCOBAR+KGAVE-1
IF (LASTAB<5000) 1600, 1600, 1555
1555 WRITE (6,120) LASTAB
KEND=1
KEFN=1555
GO TO 2050
C
1600 KTABLE=2
CALL TABLE (MA,C,CN2N, XN,XJR,XJZ, V, IDMAP, IDGP,
 1  XTABILKSI),XTAB(LVSUM), XTAB(LFX),XTAB(LCX),XTAB(LCOUS),
 2  XTAB(LUISAD), XTAB(LFXBAR),XTAB(LCXBAR),XTAB(LCOBAR)
 1 IF (KEND) 2000, 2000, 2050
C
C
2000 KCHAIN=5
RETURN
C
2050 KCHAIN=1
WRITE (6,104) KEFN
RETURN
C
END
SUBROUTINE IDACAV(INR, NZ, NACAV, NZONRA, NZONZA, NMRA, NMZA, IDM, I, IDMAP)
C
SUBROUTINE IDACAV OBTAINS THE IDENTIFICATION NUMBER MAP TO USE FOR ACTIVITY TABLES AND AVERAGING TABLES.
C
C THE DIMENSION STATEMENTS
DIMENSION NMRA(I), NMZA(I), IDM(I)
DIMENSION IDMAP(I)
C
C THE FORMAT STATEMENTS
110 FORMAT (7110)
115 FORMAT (1415)
116 FORMAT (4313)
120 FORMAT (1HK, 15HMAP FOR TABLES)
C
C THE INTEGER FUNCTIONS
INDEX(LENGTH, INDEXO, INDEXL) = LENGTH*(INDEXO-1)+INDEXL
C
C 1000 READ (5, 110) NZONRA, NZONZA
READ (5, 110) (NMRA(I), I=1, NZONRA)
IF (NZONZA) 1005, 1005, 1010
1005 NZONZA=1
NMZA(1)=1
GO TO 1025
1010 READ (5, 110) (NMZA(J), J=1, NZONZA)
C
1025 K=0
J=0
NACAV=0
DO 1050 JJ=1, NZONZA
READ (5, 115) (IDM(II), II=1, NZONRA)
K=K+NMZA(JJ)
1030 J=J+1
I=0
DO 1045 II=1, NZONRA
L=NMRA(II)
DO 1035 KK=1, L
I=I+1
IJ=INDEX(INR, J, I)
1035 IDMAP(IJ)=IDM(II)
IF (IDM(II)) 1036, 1040, 1040
1036 IF (NACAV+IDM(II)-1) 1037, 1045, 1045
1037 NACAV=-IDM(II)+1
GO TO 1045
1040 IF (NACAV-IDM(II)) 1041, 1045, 1045
1041 NACAV=IDM(II)
1045 CONTINUE
  IF (J-K) 1030,1050,1050
1050 CONTINUE
C
C
1075 WRITE (6,120)
   DD 1080 J=1,NZ
   II=INDEX(NK,J,1)
   INR=INDEX(INR,J,INR)
1080 WRITE (6,116) (IDMAP(IJ), IJ=II,INR)
C
C
1100 RETURN
C
   END
$1BFTC

TABLE DECK

SUBROUTINE TABLE (MA,C, CA2N, XN,XJR,XJZ, V,

1 IUMAP,IDGP, KSI,VSUM, FX,CX,COUT, DISAD,

2 FXBAR,CXBAR,CUBAR )

C C C C C C

THE COMMON STATEMENTS

COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED,
1 KREG, KALC, KGB, KBBC, KTBC, KLBC,
2 KBBC, NTS, NMAT, LBUCK, KCTYP,
3 KSISO, KGST, KCTR, KCGG, KOREAD, KFLUX1,
4 KACL, KACV, KGAKE, KNIP, KELT,
5 FRE, TIMAX, CALCI, OMEGA, EPS,
6 KACAV, KGAVE, KGN, KFLUX2,
7 COMMON /CALL3/ H2, HY, BF, KCA, KCF,
8 TIMEX, TIMEZ, L2SAV, SCALUP
COMMON /CALL4/ OMEGA, OMEGAPS,
1 EXTRA1, EXTRA2, EXTRA3, EXTRA4, EXTRA5,
2 COMMON /CTABLE/ KTABLE, NACAVT, NTYBAR, KGGBAR
COMMON /CSUM/ KSUM, NACAV2, NACAV1, NACAV3, N2N,
1 COMMON /CAVER/ ID, KF, KC, NACAV

C C

THE DIMENSION STATEMENTS

DIMENSION MA(I), C(I)
DIMENSION CA2N(I)
DIMENSION XN(I), XJR(I), XJZ(I)
DIMENSION V(I)
DIMENSION IUMAP(I), IDGP(I)
DIMENSION KSI(I), VSUM(I)
DIMENSION FX(I), CX(I), COUT(I)
DIMENSION DISAD(I)
DIMENSION FXBAR(I), CXBAR(I), CUBAR(I)

C C

THE FORMAT STATEMENTS

104 FORMAT (1PL,21H)
111 FORMAT (7116)
113 FORMAT (7E16.8)
162 FORMAT (1H1,17HAVERAGING TABLES )
163 FORMAT (1H1,17HDISADVANTAGE FACTORS )
164 FORMAT (1HL)
165 FORMAT (1HL,32HZONE IDENTIFICATION NUMBER (ID)=,I3)
166 FORMAT (1HL,30HGROUP COLLAPSED CROSS SECTIONS)
167 FORMAT (1HL, 33H)D= NUMBERS FOR COLLAPSED GROUPS )
168 FORMAT (1H5, 30HGROUP COLLAPSED CROSS SECTIONS)
169 FORMAT (1H5, 7F10.6)
170 FORMAT (1H )
171 FORMAT (1HL,28HTOTAL OF P(0) AND P(1) ZONES)
172 FORMAT (1HL,19HTOTAL OF P(1) ZONES)
173 FORMAT (12H GROUP TOTAL)
174 FORMAT (1HL,128HN TO 2N REACTIONS ARE NOT ACCOUNTED FOR IN THE WHIRE
181 HIN GROUP SCATTERING CROSS SECTION WHICH HAS BEEN COMPUTED BY NEUT
189 RN BALANCE ./5X,92HFROM SCATTERING THAT CONTAINS THE EXTRA NEUTRO
196 NS AND THE TOTAL COLLISION LOSS WHICH DOES NOT)

C THE INTEGER FUNCTIONS
199 INDEX(LNTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL

C
C 1000 GO TO (1150,1550), KTABLE
C
C COMPUTE THE ACTIVITY TABLES.
C
C
1150 KF=0
211 KC=0
261 DO 1165 ID=1,NACAVT
271 DO 1160 IG=1,NGPI
301 KF=KF+1
311 DO 1155 N=1,NITYPS
321 KC=KC+1
1155 CX(KC)=0.0
361 1160 FX(KF)=0.0
401 KSII(1D)=0
411 1165 VSUM(1D)=0.0
421 DO 1170 IJ=1,NIJ
431 1170 VSUM(NACAV1)=VSUM(NACAV1)+V(IJ)
441 IF (KSISO) 1200,1200,1175
1175 VSUM(NACAV2)=VSUM(NACAV1)
451 KSII(NACAV2)=1
C
C 1200 DO 1250 IG=1,NG
C
C 1225 KSUM=1
1226 CALL SUM (MA,C, CN2N, XN,XJR,XJZ, V, IDMAP, KSI,VSUM, )
1227 1 FX,CX,COUT )
1228 IF (KEND) 1250,1250,2050
1250 CONTINUE
C
C
1300 KF=0
KC=0
DO 1335 ID=1,NACAVT
   IF (ID=NACAV1) 1305,1310,1315
1305 WRITE (6,165) ID
   GO TO 1325
1310 WRITE (6,171)
   GO TO 1325
1315 WRITE (6,172)
C
1325 DO 1330 IG=1,NGP1
   IF (IG-NGP1) 1327,1326,1326
1326 WRITE (6,173)
1327 KF=KF+1
   K1=KC+1
   K2=KC+NTPS
   WRITE (6,113) (CX(K), K=K1,K2), FX(KF)
   WRITE (6,170)
1330 KC=K2
1335 CONTINUE
C
C   COMPUTE THE DISADVANTAGE FACTORS.
C
1400 WRITE (6,163)
   KF=0
   DO 1475 ID=1,NACAV
      WRITE (6,165) ID
      IF (KSI(ID)) 1405,1405,1410
1405 KFI=INDEX(NGP1,NACAV1,0)
      GO TO 1425
1410 KFI=INDEX(NGP1,NACAV2,0)
1425 DO 1450 IG=1,NG
      KFI=KF+1
      KF=KF+1
      DISAD(IG)=FX(KF)/VSUM(ID)
      IF (KSI(ID)) 1430,1430,1435
1430 DISAD(IG)=DISAD(IG)/(FX(KFI)/VSUM(NACAV1))
      GO TO 1450
1435 DISAD(IG)=DISAD(IG)/(FX(KFI)/VSUM(NACAV2))
1450 CONTINUE
   KF=KF+1
C
   WRITE (6,113) (DISAD(IG), IG=1,NG)
1475 WRITE (6,170)
C
   RETURN
C
C   COMPUTE THE AVERAGING TABLES.
C
1550 KF=0
KC=0
   DO 1565 ID=1,NACAVT
   DO 1560 IG=1,NGP1

160
KF=KF+1
DO 1555 N=1,NTYPS
KL=KC+1
1555 CX(KC)=0.0
COUT(KF)=C.0
1560 FX(KF)=0.0
1565 KSII(ID)=0
IF (KSISO) 1600,1600,1570
1570 KSISO=1
1600 DO 1650 IG=1,NG
C
1625 KSUM=2
CALL SUM (MA,C, CN2N, XN,XJR,XJZ, V, IDMAM, KSI,VSUM,
1 FX,CX,COUT )
IF (KEND) 1650,1650,2050
1650 CONTINUE
C
1700 KC=0
DO 1750 ID=1,NACAVT
IF (IABS(KACAVT)) 1740,1740,1705
1705 IF (ID=NACAVT) 1710,1715,1720
1710 WRITE (6,165) ID
IF (KACAV) 1730,1740,1740
1715 WRITE (6,171)
GO TO 1730
1720 WRITE (6,172)
1730 WRITE (6,162)
C
1740 CALL AVERAG ( KSI, FX,CX,COUT )
1750 CONTINUE
C
COMPUTE THE COLLAPSED GROUP CROSS SECTIONS.
C
1800 IF (KGAVE) 2000,2000,1805
1805 WRITE (6,166)
WRITE (6,168)
WRITE (6,167)
WRITE (6,111) (IDGP(IG), IG=1,NG)
IF (N2N) 1807,1807,1806
1806 WRITE (6,174)
1807 KGAVE=0
1807 DO 1875 IG=1,KGAVE
IF (ID=NACAVT) 1810,1815,1820
1810 WRITE (6,165) ID
GO TO 1825
1815 WRITE (6,171)
GO TO 1825
1820 WRITE (6,172)
C
1825 CALL COLAPS ( KGAVE,NTYBAR,KGGBAR, IDGP, KSI, FX,CX,
1 FXBAR,CXBAR,COBAR )
1850 DO 1660 IG=1,KGAVE
K1=INDEX(NTYBAR,IG,1)
K2=INDEX(NTYBAR,IG,NTYBAR)
WRITE (6,113) (CXBARR(K), K=K1,K2), FXBAR(K)
WRITE (6,170)
1860 WRITE (6,169) (CXBARR(K), K=K1,K2)
C
1875 CONTINUE
C
2000 RETURN
C
2050 WRITE (6,104) KEFN
KEFN=2050
RETURN
C
END
SUBROUTINE SUM (XAC, CN2N, XNR, XJR, XJZ, V, IDMAP, KSI, VSUM,  
1 FAX, CXL, COUT  )

C
C
C
C
C
C
C
C
C
C
C
C
C
C

COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED
1  KRCG, KLEC, KGEO, KBBG, KTBG, KLBC,  
2  KRCN, KLEO, NSN, NG, NZ,  
3  NKR, NTYP, NMAK, LBUC, KCTYP,  
4  KSIS0, KGST, KTK, KCG, KORE, KFLEX,  
5  ACCEL, KCALC, NFN, N2N, ITMPRT,  
6  KALCP, KACA, KAVE, KNIP, KEDIT,  
7  KCy, TIMAX, GLAM, EIGEN, EPS, XNF,  
8  KCy, TIMAX, CALC1, OMEGA, OMEGAP

C
C
C
C
C
C
C
C
C
C
C

COMMON /CALL3/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED
1  KRCG, KLEC, KGEO, KBBG, KTBG, KLBC,  
2  KRCN, KLEO, NSN, NG, NZ,  
3  NKR, NTYP, NMAK, LBUC, KCTYP,  
4  KSIS0, KGST, KTK, KCG, KORE, KFLEX,  
5  ACCEL, KCALC, NFN, N2N, ITMPRT,  
6  KALCP, KACA, KAVE, KNIP, KEDIT,  
7  KCy, TIMAX, GLAM, EIGEN, EPS, XNF,  
8  KCy, TIMAX, CALC1, OMEGA, OMEGAP

THE COMMON STATEMENTS
DIMENSION MAT(1), CN2N(1)
DIMENSION XNR(1), XJR(1), XJZ(1)
DIMENSION FX(1), IDMAP(1)
DIMENSION VSUM(1)
DIMENSION COUT(1)

THE DIMENSION STATEMENTS
THE FINTMAT STATEMENTS
104 FORMAT (1H4,21HPRGRAM STOP AT KEFN=,I8)
THE INTEGER FUNCTIONS
INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*INDEXO+INDEXL
1000 IJU=INDEX(INIJ,IG,O)
DO 1200 IJ=1,INIJ
IJG=IJG+1
ID=IABS(IDMAP(IJ))  

163
ICA=ID
K=|ABS(MA(IJ)|
KA=K

TEMP1=V(IJ)*XN(IJG)
KF=INDEX(NGP1,ID,IG)
GO TO (1010,1005), KSUM

1005 KOUT=INDEX(NG,K,IG)
TEMP=CN2N(KOUT)*TEMP1
COUT(KF)=COUT(KF)+TEMP

1010 FX(KF)=FX(KF)+TEMP1
KF=INDEX(NGP1,NACAV1,IG)
GO TO (1020,1015), KSUM

1015 COUT(KF)=COUT(KF)+TEMP1
KF=INDEX(NGP1,ID,NGP1)
GO TO (1030,1025), KSUM

1025 COUT(KF)=COUT(KF)+TEMP1
KF=INDEX(NGP1,ID,NGP1)
GO TO (1040,1035), KSUM

1035 COUT(KF)=COUT(KF)+TEMP1
KF=INDEX(NGP1,NACAV1,ID,IG)
GO TO (1050,1045), KSUM

1040 COUT(KF)=COUT(KF)+TEMP1
KF=INDEX(NGP1,NACAV1,NGP1)
GO TO (1060,1055), KSUM

1050 KEND=1
KCFN=1050
GO TO 1275

1055 KA=KA+1
TEMP2=V(IJ)*ABS(XJR(IJG))
IF (NZM1) 1065,1065,1060

1060 TEMP2=TEMP2+V(IJ)*ABS(XJZ(IJG))

1065 KF=INDEX(NGP1,ID,IG)
FX(KF)=FX(KF)+TEMP2
KF=INDEX(NGP1,NACAV2,IG)
FX(KF)=FX(KF)+TEMP2
KF=INDEX(NGP1,ID,NGP1)
FX(KF)=FX(KF)+TEMP2
KF=INDEX(NGP1,NACAV2,NGP1)
FX(KF)=FX(KF)+TEMP2

C

1075 DD 1125 N=1,NTYPS
IF (KREG) 1076,1076,1080
1076 IF (N-KCTR) 1080,1080,1085
1080 IGN=IG
GO TO 1100

1085 IGN=IG-KCG+1
IF (IGN) 1125,1125,1090

1090 IF (IGN-NG) 1100,1100,1125

1100 K1=INDEX(NTYPS,IGN,N)
K2=INDEX(NTYPES,IGN,K1)
KC=INDEX(NTYG1,IG,K1)
KCS=INDEX(NTYG1,NACAV1,IG)
TEMP=C(K2)*TEMP1
CX(KC)=CX(KC)+TEMP
CX(KCS)=CX(KCS)+TEMP
IF (KREG) 1101, 1101, 1102
1101 IF (IG-IGN) 111C, 1105, 1110
1102 IF (N-KCTR) 1105, 1105, 1103
1103 IF (N-KCGG) 111C, 1105, 1110
1105 KT = INDEX(NTPS, NGP1, N)
   KC = INDEX(NTPG1, ID, KT)
   KC$ = INDEX(NTPG1, NACAVL, KT)
   CX(KC) = CX(KC) + TEMP
   CX(KCS) = CX(KCS) + TEMP
1110 IF (ID-IDA) 1115, 1125, 1125
1115 K2 = INDEX(NTPG1, IDA, K1)
   KC = INDEX(NTPG1, IDA, K1)
   KC$ = INDEX(NTPG1, NACAVP, K1)
   TEMP = C(K2) * TEMP2
   CX(KC) = CX(KC) + TEMP
   CX(KCS) = CX(KCS) + TEMP
   IF (KREG) 1116, 1116, 1117
1116 IF (IG-IGN) 1125, 1120, 1125
1117 IF (N-KLTR) 1120, 1120, 1118
1118 IF (N-KCGG) 1125, 1120, 1125
1120 KC = INDEX(NTPG1, IDA, KT)
   KC$ = INDEX(NTPG1, NACAVP, KT)
   CX(KC) = CX(KC) + TEMP
   CX(KCS) = CX(KCS) + TEMP
1125 CONTINUE
C
1150 IF (IG-1) 1155, 1155, 1200
1155 IF (ID-IDA) 1165, 1160, 1160
1160 GO TO (1175, 120C), KSUM
1165 KSI(IDA) = 1
   GO TO (117G, 120G), KSUM
1170 VSUM(IDA) = VSUM(IDA) + V1(J)
1175 VSUM(ID) = VSUM(ID) + V1(J)
C
1200 CONTINUE
C
1250 RETURN
C
1275 WRITE (6, 104) KEFN
   RETURN
C
END
$IBFIC AVERAG

SUBROUTINE AVERAG ( KSI, FX, CX, COUT )

THE COMMON STATEMENTS
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED, KREG, KSI
1 KREG, KALC, KGEO, KBBE, KTBC, KLBC,
2 KBBE, NSN, NG, NI,
3 NR, NTYPES, NMAT, LBUCR, KCTYP, KSI0,
4 KSI0, KSISO, KCTR, KCGG, KDREAD, KFLUXI,
5 KACCEL, KAC, NFN, N2N, ITMPRT,
6 ITMIX, ITMIGM, NZONE, NZONEZ,
7 KBGDUP, KACAV, KGAVE, KNIP, KEDIT,
8 GLAM, EIGEN, EPS, XNF,
9 RYF, TIMAX, CALC1, OMEGA, OMEGAP
COMMON /CALL3/ HZ, HY, BF,
1 KCA, KCF,
2 KUNV, KFINSH, KASOR, KUPS,
3 NBF, ND, NGP1, NZP1, NRP1,
4 NIJ, NTYNG, NDNZ, NDNR, NDNIJ, NONZNG,
5 NXXN, NAR, NC, NIJG, NBUCR,
7 ITMA, ITMI, ITMIST, ITMIGX, ITD,
8 NEGSOR, LZSAV,
9 TIMEX, TIMEZ, SOFG1, SCALUP
COMMON /CAVEER/ ID, KF, KC, NACAV

THE DIMENSION STATEMENTS
DIMENSION KSI(1), FX(1), COUT(1)

THE FORMAT STATEMENTS
113 FORMAT (7E16.8)
169 FORMAT (1H$, 7F10.6)
170 FORMAT (1H )
173 FORMAT (1HJ, 2H COLLAPSED TO ONE GROUP)
174 FORMAT (24H OUT SCATTERING BY GROUP)
175 FORMAT (1H$, 24H OUT SCATTERING BY GROUP)

THE INTEGER FUNCTIONS
INDEXLENGTH, INDEXO, INDEXL = LENGTH * (INDEXO - 1) + INDEXL

1000 DO 1125 IG=1, NG
   K1 = KC + 1
   K2 = KC + NTYPES
DO 1100 N=1, NTYPES
   KC = KC + 1
   IF (N-KCTR) 1005, 1005, 1010

166
1005 IGN=IG
GO TO 1025
1010 IGN=IG+KCGG-N
IF (IGN) 1100,1100,1015
1015 IF (IGN-NG) 1025,1025,1100
1025 KF=INDEX(NGP1,ID,IGN)
IF (N-KCTR) 1050,1030,1035
1030 KC1=KC
KF1=KF
COUT(KF)=COUT(KF)/FX(KF)
GO TO 1050
1035 IF (KSI(ID)) 1040,1040,1050
1040 IF (N-KCG) 1050,1075,1050
1050 (X(KC)=X(KC)/FX(KF)
GO TO 1100
1075 CX(KC)=CX(KC)+CX(KC)-COUT(KF)
1100 CONTINUE
C
1105 WRITE (6,113) (CX(K), K=K1,K2), FX(KF1)
WRITE (6,170)
IF (ID--NACAV) 1110,1110,1115
1110 IF (KACAV) 1115,1125,1125
1115 WRITE (6,169) (CX(K), K=K1,K2)
1125 CONTINUE
C
1150 KF=INDEX(NGP1,ID,NGP1)
K1=KC+1
K2=KC+NTYPES
DO 1155 N=1,KCTR
KC=KC+1
1155 CX(KC)=CX(KC)/FX(KF)
KC1=KC
ITEMP=KCTR+1
DO 1160 N=ITEMP,KCGG
KC=KC+1
IF (KSI(ID)) 1170,1170,1165
1165 CX(KC)=CX(KC)/FX(KF)
GO TO 1175
1170 CX(KC)=CX(KC)+CX(KC)-2
1175 IF (IABS(KACAV)=1) 1185,1185,1180
1180 WRITE (6,173)
WRITE (6,113) (CX(K), K=K1,K2), FX(KF)
WRITE (6,170)
1185 KC=K2
C
WRITE (6,174)
KF1=INDEX(NGP1,ID,1)
KF2=INDEX(NGP1,ID,NG)
WRITE (6,113) (OUT(KF), KF=KF1,KF2)
IF (ID--NACAV) 1190,1190,1195
1190 IF (KACAV) 1195,1200,1200
1195 WRITE (6,175)
WRITE (6,169) (OUT(KF), KF=KF1,KF2)
C
C
1200 RETURN
C
END
SUBROUTINE COLAPS IS TO GIVE COLLAPSED GROUP CROSS SECTIONS.
(THIS VERSION MUST BE CALLED BY TABLES.)

THE COMMON STATEMENTS
COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KBCRED,
1 KREG, KALC, KGD, KBBC, KTBC, KLBC,
2 KEBC, NSN, NG, NZ,
3 NTR, NTYPS, NMAT, LBUCK, KCY,
4 KISO, KGST, KCTR, KGG, KOREAD, KFLUXI,
5 KIG, KACCEL, KCALC1, NFN, N2N, K.elementAt
6 KBCDUP, KACAV, KGAVE, KNIP, KEDIT,
7 KREM, GLAM, EIGEN, EPS, XNF,
8 RYF, TIMAX, CALC1, OMEGA, OMEGAP
COMMON /CALL3/
1 HZ, HY, BF, KCA,
2 DNC, KINS, KASOR, KUPS,
3 KEBC, ND, NGP, N2P, NRP,
4 KACCEL, NTYNG, NDNZ, NDNR, NDNIJ, NDNZNG,
5 KSA, NAR, NCI, NIJG, NBUCK,
6 ITMA, ITMI, ITMIST, ITMIG, ITM,
7 GLAM, EIGEN, EPS, XNF,
8 TIMEX, TIMEZ, SDGFG1, SCALUP
COMMON /CAVER/ 1D, KF, KC, NACAV

THE DIMENSION STATEMENTS
DIMENSION KEG1)
DIMENSION KEG(1)
DIMENSION FS(1), CS(1)
DIMENSION FOLAP(1), COLAP(1), OUTLAP(1)

THE INTEGER FUNCTIONS
INDEX(L.EN, INDEX, INDEXL) = LENGTH*INDEX - INDEX + INDEXL

1000 K=0
Do 1010 IG=1,KOLAPS
Do 1005 IH=1,NTYLP
K=K+1
COLAP(K)=0.0
1005 FOLAP(K)=0.0
1010 OUTLAP(IM)=0.0
1025 DO 1060 I=1,NG
   KF=KF+1
   IGK=IG(I)
   K=INDEX(NTYLAP, IGK, 0)
   DO 1055 N=1,NTYPS
   KC=KC+1
   IF (N-KCTR) 1030,1030,1035
1030 K=K+1
   FOLAP(K)=FOLAP(K)+FS(KF)
   L=KF
   GO TO 1050
1035 IGN=IG*KGLG-N
   IF (IGN) 1055,1055,1050
1040 IF (IGN-NG) 1045,1045,1055
1045 IGNLAP=KGI(IGN)
   NLAP=IGK-IGNLAP*KGGLAP
   IF (NLAP-KCTR) 1055,1055,1046
1046 K=INDEX(NTYLAP, IGK, NLAP)
   L=INDEX(NGP1, ID, IGN)
1050 COLAP(K)=COLAP(K)+CS(KC)*FL(L)
1055 CONTINUE
1060 CONTINUE
   KF=KF+1
   KC=KC+NTYPS
C
1100 KCTR1=KCTR+1
   DO 1115 I=1,KOLAPS
   DO 1115 N=1,KCTR1,NTYLAP
   IGN=IG*KGLG-N
   IF (IGN) 1115,1115,1105
1105 IF (IGN-KOLAPS) 1110,1110,1115
1110 K=INDEX(NTYLAP, IG, N)
   L=INDEX(NTYLAP, IGN, KCTR)
   FOLAP(K)=FOLAP(K)+FOLAP(L)
1115 CONTINUE
C
1150 DO 1180 I=1,KOLAPS
   DO 1180 N=1,NTYLAP
   IF (N-KCTR) 1155,1155,1160
1155 IGN=IG
   GO TO 1170
1160 IGN=IG-N*KGLG
   IF (IGN) 1180,1180,1155
1165 IF (IGN-KOLAPS) 1170,1170,1180
1170 K=INDEX(NTYLAP, IG, N)
   COLAP(K)=COLAP(K)+FOLAP(K)
   IF (IGN-1G) 1175,1175,1170
1175 OUTLAP(IGN)=OUTLAP(IGN)+COLAP(K)
1180 CONTINUE
   IF (KSI(ID)) 1185,1185,1200
1185 DO 1190 I=1,KOLAPS
   L=INDEX(NTYLAP, IG, KCTR)
   K=INDEX(NTYLAP, IG, KGGLAP)
1190 COLAP(K)=COLAP(L)+COLAP(L-2)-OUTLAP(IG)
C
C
C
1200 RETURN
C
END
$18FTG EDIT DECK

SUBROUTINE EDIT (MA,C,XN,Y)

SUBROUTINE EDIT COMPUTES ACTIVITIES, FLUX INTEGRALS, AND AVERAGE
ACTIVITIES FOR A SPECIFIED CROSS SECTION OVER GIVEN SETS OF
GROUPS AND REACTOR REGIONS.

THE COMMON STATEMENTS

COMMON /CALL1/ KCHAIN, KEND, KEFN
COMMON /CALL2/ KGCRD, KGCRD
1 KREG, KCALC, KGEU, KBC, KTBC, KLBC, 2 KREG, KCALC, KGEU, KBC, KTBC, KLBC, 3 NR, NTYPS, NMA, KBUCK, KCTYP, 4 KSID, KGST, KCIH, KCGU, KREAD, KFLUXI, 5 TIMIMX, TIMS, N2R, N2Z, ITMTR, 6 KACCEL, KALC1, NFN, N2N, KEDIT, 7 KBCDUP, KACAV, KSN, KNIP, KEDIT, 8 GLAM, EIGEN, EPS, XNF, 9 RYF, TIMAX, CALCI, OMEGA, OMEGA

THE DIMENSION STATEMENTS

DIMENSION MA(1,1), C(1), XN(1), 1 VI(1)
DIMENSION ACTINT(201), FINT(201), PAVE(201)

THE FORMAT STATEMENTS

100 FORMAT (1H1)
101 FORMAT (1HL)
110 FORMAT (7H10)
112 FORMAT (7F10.6)
113 FORMAT (7E16.8)
200 FORMAT (1HL,9HD19NTRE, 13H HAVING LIMITS FROM 12.4H TO 12.31H ON RADIAL MESH INTERVALS, AND 1 12H ON GRUUPS, 12.4H TO 12.31H ON AXIAL MESH INTERVALS)
201 FORMAT (1HL,19H SGMA EQUALS, E16.8,14H FOR OPTION 1)

170
format (1hl,32hactivity integrals for edit set ,13)
format (1hl,28hflux integrals for edit set ,13)
format (1hl,32haverage activities for edit set ,13)
C

the integer functions
index(length,indexo,indexlj)=length*(indexo-1)*indexl
C

1000 write (6,100)
read (5,110) ni
set
do 1175 iset=1,ni
set
write (6,200) iset
read (5,11c) int
inpi=int+1
actint(inpt1)=0.0
fint(inpt1)=0.0
C

do 1150 in=1,ini
read (5,11c) ig1,ig2,ir1,ir2,iz1,iz2
write (6,201) in,ig1,ig2,ir1,ir2,iz1,iz2
actint(in)=0.0
fint(in)=0.0
C

1020 do 1125 i=ig1,ig2
go to (1025,105g,1055,1060), kedit
1025 read (5,112) sigma
write (6,202) sigma
go to 1075
1050 k1=index(nyps,ig,kca)
go to 1075
1055 k1=index(nyps,ig,kcf)
go to 1075
1060 k1=index(nyps,ig,kctr)
1075 do 1125 j=iz1,iz2
do 1125 j=ir1,ir2
ij=index(nr,j,i)
ijg=index(nij,ig,ij)
go to (111c,110c,1100,1100), kedit
1100 k=abs(max(ij))
k2=index(nypg,k,k1)
sigma=c(k2)
1110 actint(in)=actint(in)+sigma*xn(ijg)*v(ij)
1125 fint(in)=fint(in)+xn(ijg)*v(ij)
C

actint(inpt1)=actint(inpt1)+actint(in)
fint(inpt1)=fint(inpt1)+fint(in)
1150 pavel(in)=actint(in)/fint(in)
pavel(inpt1)=actint(inpt1)/fint(inpt1)
C

write (6,203) iset
write (6,113) (actint(in), in=1,intp1)
write (6,204) iset
write (6,113) (fint(in), in=1,intp1)
write (6,205) iset
write (6,113) (pavel(in), in=1,intp1)
write (6,101)
1175 continue
C
C

return
C

end
APPENDIX E
SAMPLE PROBLEM

<table>
<thead>
<tr>
<th>TITLE</th>
<th>TITLE</th>
<th>CONTROL</th>
<th>CONTROL</th>
<th>CONTROL</th>
<th>CONTROL</th>
<th>CONTROL</th>
<th>CONTROL</th>
<th>CONTROL</th>
<th>CONTROL</th>
<th>CONTROL</th>
<th>CONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LISTING OF SAMPLE PROBLEM OUTPUT
OVERLAY ORIGIN CARDS AND ASSIGNED LINK NUMBERS

<table>
<thead>
<tr>
<th>$ORIGIN</th>
<th>TO54CO</th>
<th>IS LINK</th>
<th>1, PARENT LINK IS 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>2, PARENT LINK IS 0</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>3, PARENT LINK IS 2</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>4, PARENT LINK IS 3</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>5, PARENT LINK IS 0</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>6, PARENT LINK IS 0</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>7, PARENT LINK IS 6</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>8, PARENT LINK IS 6</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>9, PARENT LINK IS 8</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>10, PARENT LINK IS 8</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>11, PARENT LINK IS 8</td>
</tr>
<tr>
<td>$ORIGIN</td>
<td>TO54CO</td>
<td>IS LINK</td>
<td>12, PARENT LINK IS 0</td>
</tr>
</tbody>
</table>
MEMORY MAP

SYSTEM

FILE BLOCK ORIGIN

NUMBER OF FILES - 4

1. READ5
2. PCH
3. UNIT05
4. UNIT06

FILE LIST ORIGIN

PRE-EXECUTION INITIALIZATION

CALL ON OBJECT PROGRAM

OBJECT PROGRAM

1. DECK 'TDSN ' *
   OVERLAY COMMUNICATION
   03046
   0

2. DECK 'INPUT ' *
3. DECK 'READSV ' *
4. DECK 'XSTDSN ' *
5. DECK 'SETUP ' *
6. DECK 'BCREAD ' *
7. DECK 'READ5 ' *
8. DECK 'OUTER ' *
9. DECK 'OKSCAL ' *
10. DECK 'FISI9NN ' *
11. DECK 'FIXED ' *
12. DECK 'INNER ' *
13. DECK 'FLUX ' *
14. DECK 'SURAXS ' *
15. DECK 'DUMPBL ' *
16. DECK 'BCDUMP ' *
17. DECK 'PCH ' *
18. DECK 'OUTPUT ' *
19. DECK 'TABLES ' *
20. DECK 'IOAGAV ' *
21. DECK 'TABLE ' *
22. DECK 'SLM ' *
23. DECK 'AVEKAV ' *
24. DECK 'GULAPS ' *
25. DECK 'EDIT ' *
26. SUBR 'JBSYS ' *
27. SUBR 'LRCON ' *
28. SUBR 'IOEX ' *
29. SUBR 'JBCON ' *
30. SUBR 'LXCON ' *
31. SUBR 'IOQEF ' *
32. SUBR 'JOCSE ' *

00000 THRU 02717
02720
03000
03010
03041
03046 THRU 34262

173
33. SUBR 'LUVKY' * 10227 0
34. SUBR 'LXSL' 10606 0
35. SUBR 'FPTRP' 10745 0
36. SUBR 'FOUT' 11413 0
37. SUBR 'FLNV' 11461 0
38. SUBR 'FILS' 13263 0
39. SUBR 'FIOH' 13360 0
40. SUBR 'FSEL' 14331 0
41. SUBR 'FBRD' 14416 0
42. SUBR 'FRUD' 14445 0
43. SUBR 'UNOS' 14477 0
44. SUBR 'UNOS' 14500 0
45. SUBR 'FSQN' 14503 0
46. SUBR 'FSQN' 14557 0
47. SUBR 'FUIE' 14723 0
48. SUBR 'FUE56' 14744 0
49. SUBR 'KDOLE' 14765 0
50. SUBR 'SINL' 15004 0
51. SUBR 'SOKTN' * 15004 0
52. SUBR 'TIMEI' * 15007 0
53. SUBR ' // ' 35644 0

(*) - INSERTIONS OR DELETIONS MADE IN THIS DECK)

INPUT - OUTPUT BUFFERS
34263 THRU 35626

UNUSED CORE
35627 THRU 35643

BEGIN EXECUTION.
CARLSUN TEST PROBLEM  SLAB  C=1.1
3 WHUP'S  S=6  32 INTERVALS
2-24-66

K8RED= 0

KREG  KALC  KGED  KLBC  KKBC  KBBC  KTBC
0    1    1    1    0    0    0

NSN  NG  NK  NZ  NMAT  NTIPS  N2N
4    3    0    0    2    9    0

KCTYP  KS150  KUST  KCTA  KCGG  NBUCK  LBUC
0    0    1    4    7    0    0

ITMIX  ITMIX  NIZUNER  NIZUNL  KOREAU  KFLUXI  KALL1
4000  7    0    2    0    0    0

GLAM  EPS  XNS  KRF  TIMAX  CALCI
0    0    0    0.949999999E-04  1.000000000E+00  0.150000000E+01

UMEGA  OMEGAP
0.120000000E+01  0.165000000E+01

O.120000000E+01  0.165000000E+01

KACCA  KACAV  KGAVF  KNIK  KEDUT
-2    1    1    1    1

LAST  LAST1  LAST2  LAST3  LAST4
854  619  854  809  144

K8H1(I)  K8H2(I)  MU8(I)
16  0.105670000E+01  1
16  0.211300000E+01  2

THE FIRST DIKELTION MESH CELL BOUNDARY POSITIONS (R(INI))
0.6643747E-01  0.720675E-00  0.1913125E-00  0.2641750E-00
0.526850000E+00  0.5439575E-00  0.6094379E-00  0.7264612E-00
0.3905905E-00  0.1905905E-00  0.1637006E+01  0.1122743E-01
0.1180784E-01  0.3654832E-01  0.1590700E+01  0.1651093E-01

THE MATERIAL MAP IMAP(NZ;NR)
1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2

THE FISSION SPECTRUM (XKL(INI))
0.33333333E-00  0.33333333E-00  0.33333333E-00
THE CROSS SECTIONS (CMAT, ING, NTYPES)
N= 1
G.50000000E 00 0. 0.9999999E-01 1.0000000E 00 0.3333333E-00 0.3333333E-00
0. 1.0000000E 00 0. 0.9999999E-01 1.0000000E 00 0. 0.3333333E-00 0.3333333E-00
0. 0.3333333E-00 0. 0.9999999E-01 1.0000000E 00 0. 0. 0.3333333E-00
0. 0.15000000E 00 0. 0.9999999E-01 1.0000000E 00 0. 0. 0.3333333E-00
0. 0.3333333E-00 0. 0.9999999E-01 1.0000000E 00 0. 0. 0.3333333E-00
N= 2
0. 0.50000000E 00 0. 0.9999999E-01 1.0000000E 00 0. 0. 0.50000000E 00
0. 1.00000000E 00 0. 0.9999999E-01 1.0000000E 00 0. 0. 0.50000000E 00
0. 0.50000000E 00 0. 0.9999999E-01 1.0000000E 00 0. 0. 0.50000000E 00
0. 0.50000000E 00 0. 0.9999999E-01 1.0000000E 00 0. 0. 0.50000000E 00

THE UUI SCATTERING REMOVAL CROSS SECTIONS (CMAT, ING)
N= 1
0.66666666E 00 0.66666666E 00 0.66666666E 00
N= 2
0.50000000E 00 0.50000000E 00 0.50000000E 00

THE MESH CELL VOLUMES (VINZ, AR)
J= 1
0.66043749E-01 0.66043749E-01 0.66043749E-01 0.66043749E-01 0.66043749E-01
0.66043749E-01 0.66043749E-01 0.66043749E-01 0.66043749E-01 0.66043749E-01
0.66043749E-01 0.66043749E-01 0.66043749E-01 0.66043749E-01 0.66043749E-01
0.66043749E-01 0.66043749E-01 0.66043749E-01 0.66043749E-01 0.66043749E-01

DIRECTIONAL FUNCTIONS
-1.00000000E 00 -0.8019176E 00 -0.62844078E 00 -0.37796447E 00 -0.1253818E 00
0.62844078E 00 0.8019176E 00
0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00
0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00
0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00 0.1253818E 00
THE MAXIMUM RATE OF CHANGE IN THE SELF-SCATTERING SOURCE OCCURS AT THE MESH INTERVAL (AXVAR(11G))
13 1 1

THE MAXIMUM RATE OF CHANGE IN THE SELF-SCATTERING SOURCE (ESJH(11G))
0.144694E-06 0.2032E-07 0.184394E-07

TOTAL NEUTRON FLUX (XNGL11G))
0.254001E+11 0.395070E+11 0.39593E+11 0.94953E+11

FISSION SOURCE (FGL1(11G))
0.333333E-06 0.333333E-06 0.00000E+00

IN SCATTERING SOURCE (SIGL1(11G))
0.141877E+01 0.239309E+01 0.218628E+01

OUT SCATTERING LOSS (SGL11G))
0.157808E+01 0.228647E+01 0.212706E+01

SELF SCATTERING SOURCE (SGL11G))
0.962014E+00 0.196643E+01 0.185170E+01

NON-SELF SCATTERING SOURCE (SGL11G))
0.179211E+01 0.242642E+01 0.231320E+01

TOTAL SOURCE (STGL11G))
0.271412E+01 0.429072E+01 0.394469E+01

FISSION NEUTRONS (FNU1(11G))
0.254001E-06 0.395070E-06 0.39593E-06 0.94953E-06

ABSORPTIONS (AGL11G))
0. 0. 0.

HORIZONTAL LEAKAGE (HNLG11G))
0.174108E-06 0.439952E-06 0.389934E-06 0.999933E-06

NET LEAKAGE (XNLG11G))
0.174108E-06 0.439952E-06 0.389934E-06 0.999933E-06

NEUTRON BALANCE (ALN11G)) (SOURCE - LOSS)
0.759954E-06 0.864267E-06 0.834465E-06 0.245809E-05

NORMALIZATION FACTOR FOR F, XN, XJH, AND XJL EQUALS 0.47185151E-06

THE FISSION RATE PER UNIT VOLUME (F11J))

<table>
<thead>
<tr>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>0.290005E-06 0.28953E-06 0.28853E-06 0.287507E-06 0.286545E-06 0.285582E-06 0.284609E-06 0.283625E-06 0.282630E-06</td>
</tr>
<tr>
<td>0.275106E-06 0.274617E-06 0.274127E-06 0.273637E-06 0.273147E-06 0.272656E-06 0.272164E-06 0.271673E-06 0.271181E-06</td>
</tr>
<tr>
<td>0.254810E-06 0.254312E-06 0.253814E-06 0.253316E-06 0.252817E-06 0.252317E-06 0.251817E-06 0.251317E-06 0.250816E-06</td>
</tr>
<tr>
<td>0.171502E-06 0.171004E-06 0.169507E-06 0.168009E-06 0.166512E-06 0.165015E-06 0.163518E-06 0.162020E-06 0.160523E-06</td>
</tr>
</tbody>
</table>

179
YGDO1A2 CLAY BARNER
THE NON-DIRECTIONAL FLUX PER UNIT VOLUME BY GROUP (XLLJ1)

IG= 1

\begin{align*}
&J=1 \\
&0.847108E+00 & 6.28741E+00 & 0.450934E+00 & 0.412548E+00 & 0.927581E+00 & 0.656962E+00 & 0.847040E+00 & 0.663854E+00 & 0.615798E+00 \\
&0.552896E+00 & 0.490915E+00 & 0.450215E+00 & 0.412548E+00 & 0.784285E+00 & 0.656962E+00 & 0.847040E+00 & 0.663854E+00 & 0.615798E+00 \\
&0.279216E+00 & 0.257139E+00 & 0.235135E+00 & 0.217064E+00 & 0.199198E+00 & 0.181226E+00 & 0.162311E+00 & 0.146374E+00 & 0.142621E+00
\end{align*}

IG= 2

\begin{align*}
&J=1 \\
&0.000000E+00 & 0.999928E+00 & 0.932458E+00 & 0.898218E+00 & 0.916055E+00 & 0.927581E+00 & 0.927581E+00 & 0.916055E+00 & 0.903212E+00 \\
&0.998016E+00 & 0.905834E+00 & 0.951028E+00 & 0.918404E+00 & 0.940584E+00 & 0.972689E+00 & 0.893214E+00 & 0.804192E+00 & 0.898218E+00 \\
&0.848645E+00 & 0.990314E+00 & 0.990314E+00 & 0.962662E+00 & 0.905834E+00 & 0.972689E+00 & 0.893214E+00 & 0.804192E+00 & 0.898218E+00 \\
&0.769429E+00 & 0.729727E+00 & 0.687248E+00 & 0.641819E+00 & 0.593554E+00 & 0.540249E+00 & 0.482226E+00 & 0.417065E+00 & 0.417065E+00
\end{align*}

IG= 3

\begin{align*}
&J=1 \\
&0.992846E+00 & 0.979734E+00 & 0.977241E+00 & 0.973611E+00 & 0.962699E+00 & 0.955506E+00 & 0.947215E+00 & 0.947215E+00 & 0.955506E+00 \\
&0.932786E+00 & 0.907556E+00 & 0.894927E+00 & 0.887186E+00 & 0.879809E+00 & 0.872689E+00 & 0.865798E+00 & 0.857741E+00 & 0.857741E+00 \\
&0.868315E+00 & 0.836613E+00 & 0.821182E+00 & 0.802360E+00 & 0.783541E+00 & 0.765567E+00 & 0.748834E+00 & 0.722826E+00 & 0.697269E+00 \\
&0.696365E+00 & 0.633139E+00 & 0.590499E+00 & 0.557275E+00 & 0.515119E+00 & 0.469431E+00 & 0.419225E+00 & 0.362903E+00 & 0.362903E+00
\end{align*}

FLUX INTEGRALS

\[0.184796E+01 \quad 0.216653E+01 \quad 0.208994E+01 \quad 0.168426E+01 \quad 0.146093E+01\]
### Averaging Tables

#### Zone Identification Number (ID) = 1

<table>
<thead>
<tr>
<th>Zone Identification</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
<th>Upper Limit (2)</th>
<th>Lower Limit (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.500000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.333333333E 00</td>
<td>0.333333336E 00</td>
<td>0.333333336E 00</td>
</tr>
<tr>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.333333332E 00</td>
<td>0.333333334E 00</td>
<td>0.333333334E 00</td>
</tr>
<tr>
<td>0.150000000E 01</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.333333334E 00</td>
<td>0.333333335E 00</td>
<td>0.333333335E 00</td>
</tr>
<tr>
<td>0.333333333E 00</td>
<td>0.333333333E 00</td>
<td>0.333333333E 00</td>
<td>0.209084133E 01</td>
<td>0.333333338E 00</td>
<td>0.333333339E 00</td>
<td>0.333333339E 00</td>
</tr>
</tbody>
</table>

**Collapsed to one group**

<table>
<thead>
<tr>
<th>Group</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
</tbody>
</table>

**Out Scattering by Group**

<table>
<thead>
<tr>
<th>Group</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.666666664E 00</td>
<td>0.666666664E 00</td>
<td>0.666666665E 00</td>
<td>0.666666665E 00</td>
<td></td>
</tr>
</tbody>
</table>

#### Zone Identification Number (ID) = 2

<table>
<thead>
<tr>
<th>Zone Identification</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
<th>Upper Limit (2)</th>
<th>Lower Limit (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.500000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
<tr>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
<tr>
<td>0.500000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
<tr>
<td>0.150000000E 01</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
<tr>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
</tbody>
</table>

**Collapsed to one group**

<table>
<thead>
<tr>
<th>Group</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
</tbody>
</table>

**Out Scattering by Group**

<table>
<thead>
<tr>
<th>Group</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.400000000E 00</td>
<td>0.400000000E 00</td>
<td>0.400000000E 00</td>
<td>0.400000000E 00</td>
<td></td>
</tr>
</tbody>
</table>

#### Total of P(0) and P(1) Zones

<table>
<thead>
<tr>
<th>Zone Identification</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
<th>Upper Limit (2)</th>
<th>Lower Limit (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.500000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
<tr>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
<tr>
<td>0.500000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
<tr>
<td>0.146999998E 01</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
<tr>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
</tbody>
</table>

**Collapsed to one group**

<table>
<thead>
<tr>
<th>Group</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>0.000000000E 00</td>
<td>1.000000000E 00</td>
<td>0.000000000E 00</td>
</tr>
</tbody>
</table>

**Out Scattering by Group**

<table>
<thead>
<tr>
<th>Group</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.62125631E 00</td>
<td>0.59376999E 00</td>
<td>0.59376999E 00</td>
<td>0.59376999E 00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>Upper Limit (0)</th>
<th>Lower Limit (0)</th>
<th>Upper Limit (1)</th>
<th>Lower Limit (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.62125631E 00</td>
<td>0.59376999E 00</td>
<td>0.59376999E 00</td>
<td>0.59376999E 00</td>
<td></td>
</tr>
</tbody>
</table>

---

**Note:** The table continues with similar entries for different zones and groupings, showing the distribution of values across the specified limits.
ZONE IDENTIFICATION NUMBER (ID)= 1
0.10198124E 01 0.
0.99999999E-01 1.00000000E 00 0.00000000E 00 0.00000000E 00

ZONE IDENTIFICATION NUMBER (ID)= 2
0.11001176E 01 0.
0.99999999E-01 1.00000000E 00 1.00000000E 00 0.00000000E 00

TOTAL OF PLUS AND MINUS ZONES
0.10911461E 01 0.
0.99999999E-01 1.00000000E 00 1.00000000E 00 0.00000000E 00

EDIT SET 1

INTEGRALS 1 HAVE LIMITS FROM 2 TO 3 ON GROUPS, 5 TO 30 ON RADIAL MESH INTERVALS, AND 1 TO 1 ON AXIAL MESH INTERVALS

ACTIVITY INTEGRALS FOR EDIT SET 1
0.60641600E 01 0.00000000E 01 0.00000000E 01

FLUX INTEGRALS FOR EDIT SET 1
0.60641600E 01 0.00000000E 01

AVERAGE ACTIVITIES FOR EDIT SET 1
1.00000000E 00 1.00000000E 00 1.00000000E 00
REFERENCES


