NASA TECHNICAL NOTE

NASA TN D-3573

NASA TN D-3573

A FORTRAN IV TWO-DIMENSIONAL DISCRETE ANGULAR SEGMENTATION TRANSPORT PROGRAM

by Clayton E. Barber

Lewis Research Center

Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1966

NASA TN D-3573

A FORTRAN IV TWO-DIMENSIONAL DISCRETE ANGULAR SEGMENTATION TRANSPORT PROGRAM

By Clayton E. Barber

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $3.25

CONTENTS

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 $\bar{z}_k$ normal to $\hat{r}$, cm^2

C_{ik} area of mesh-cell surface at $\bar{r}_i$ and z_k normal to $\hat{k}$, cm^2

c number of neutrons emitted per collision, $(\nu\Sigma^f + \Sigma^s)/\Sigma^t$

f scale factor

G	number of neutron lethargy groups
$\hat{i}, \hat{j}, \hat{k}$	unit vectors for rectangular coordinates
$J_{q_g}(\vec{\rho})$	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_{eff}	neutron multiplication factor
n	order of S_n quadrature
$P_\ell(\xi)$	Legendre polynomial
$P_{\ell m}(\hat{\Omega})$	spherical harmonic
$P_\ell^m(\xi)$	associated Legendre polynomial
$\hat{r}$	unit vector in radial direction
r, z	coordinate variables to specify position $\vec{\rho}$, $\vec{\rho} = r\hat{r} + z\hat{k}$, 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 $\bar{r}_i, \bar{z}_k$, cm^3
$w_{m\ell}$	quadrature weights for angular segment m, ℓ
δ	deviation of two results
θ	angle between $\hat{k}$ and $\hat{\Omega}$
λ	growth factor for production; ratio of total production resulting from outer iteration to similar production at beginning of iteration
ν	average number of neutrons that appear as result of a nuclear fission
ξ	cosine of θ
$\vec{\rho}$	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}

$\Sigma^s(u, \vec{\rho})$	probability per unit distance of neutron travel of neutron at lethargy u and position $\vec{\rho}$ being scattered by a nucleus, cm^{-1}
$\Sigma^s(u' \rightarrow u, \vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega})$	kernal for neutron-nucleus scattering per unit distance of neutron travel from lethargy u' about lethargy u per unit lethargy through angle $\cos^{-1}(\hat{\Omega}' \cdot \hat{\Omega})$ at position $\vec{\rho}$, per cm per unit lethargy
$\Sigma^t(u, \vec{\rho})$	total collision probability per unit distance of neutron travel at lethargy u and position $\vec{\rho}$, $\Sigma^t = \Sigma^a + \Sigma^s$, cm^{-1}
$\Sigma^{tr}(u, \vec{\rho})$	macroscopic transport cross section per unit distance of neutron travel at lethargy u and position $\vec{\rho}$, cm^{-1}
$\Phi_g(\vec{\rho})$	nondirectional scalar neutron flux; number of neutrons in lethargy group g per unit area at point specified by $\vec{\rho}$, neutrons/cm^2
φ	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, $\int \chi(u)du = 1$
$\Psi(u, \vec{\rho}, \hat{\Omega})$	directional flux; number of neutrons per unit lethargy per unit solid angle per unit area at point specified by position vector $\vec{\rho}$ and flowing in direction $\hat{\Omega}$, neutrons/cm^2/sr/unit lethargy
$\hat{\Omega}$	direction of neutron flow at position $\vec{\rho}$
Ω_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 (Ω_z represents same cosine as ξ)
$\hat{\Omega} \cdot \vec{\nabla}\Psi(u, \vec{\rho}, \hat{\Omega})$	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 $\vec{\rho}$, neutrons/cm^3/ sr/unit lethargy
ω	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

ℓ	numerical position from negative r- or x-direction of angular quadrature segment on unit sphere surrounding point i,k
m	numerical position from pole or $\hat{k}$-direction of angular quadrature segment on unit sphere surrounding point i,k
q	represents generalized coordinate such as x, y, z, or r

Superscripts:

a	absorption
act	activity
f	fission
in	inelastic scattering
ℓ	order of spherical harmonic
n → 2n	n → 2n scattering
P	present outer iteration being performed
p	present inner iteration being performed
r	scattering removal
s	scattering
t	total
tr	transport
up	up-scattering, scattering to a group of higher energy
$\bar{(\ )}$	average; over a subscript, midpoint of interval with which subscript is concerned

TRANSPORT EQUATION AND BOUNDARY CONDITIONS FOR r, z GEOMETRY

The time-independent Boltzmann equation of transport theory may be written as

$$\hat{\Omega} \cdot \vec{\nabla}\Psi(u, \vec{\rho}, \hat{\Omega}) + \Sigma^t(u, \vec{\rho})\Psi(u, \vec{\rho}, \hat{\Omega}) = S(u, \vec{\rho}, \hat{\Omega}) \qquad (1)$$

where

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

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

$\Psi(u,\vec{\rho},\hat{\Omega})$ 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_{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

$$\hat{\Omega}\cdot\vec{\nabla}\Psi_{g}(\vec{\rho},\hat{\Omega}) + \Sigma^{t}_{g}(\vec{\rho})\Psi_{g}(\vec{\rho},\hat{\Omega}) = S_{g}(\vec{\rho},\hat{\Omega}) \qquad g = 1, \ldots, G \qquad (3)$$

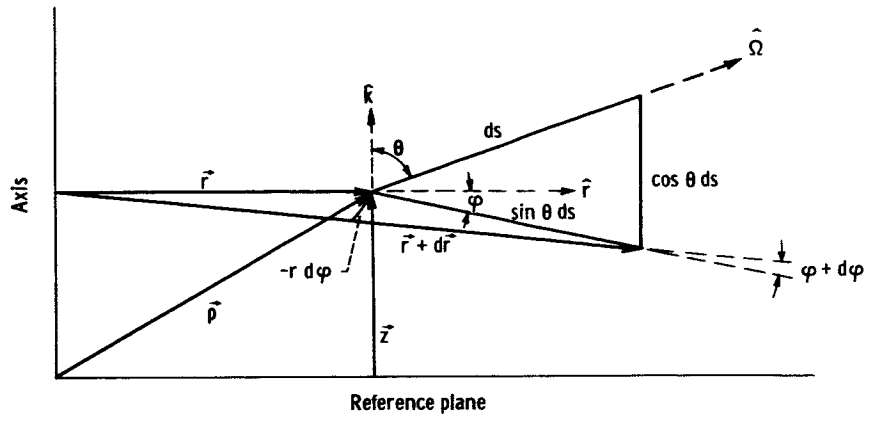
and

$$S_{g}(\vec{\rho},\hat{\Omega}) = \frac{\chi_{g}}{k_{eff}}\sum_{g'=1}^{G}\nu\Sigma^{f}_{g'}(\vec{\rho})\left[\frac{1}{4\pi}\int_{\Omega'}\Psi_{g'}(\vec{\rho},\hat{\Omega}')d^{2}\hat{\Omega}'\right]$$

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

where $\Sigma^{s}_{g'\to g}(\vec{\rho},\hat{\Omega}'\cdot\hat{\Omega})$ 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, Ψ depends on two spatial



(a)

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

$$\frac{d\Psi}{ds} = \frac{\partial\Psi}{\partial r}\frac{\partial r}{\partial s} + \frac{\partial\Psi}{\partial\varphi}\frac{\partial\varphi}{\partial s} + \frac{\partial\Psi}{\partial z}\frac{\partial z}{\partial s}$$

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

$$\frac{dr}{ds} = \sin\theta\cos\varphi = \sqrt{1-\xi^2}\cos\varphi$$

Also,

$$-r\,d\varphi = (\sin\theta\ ds)\sin\varphi$$

or

$$\frac{d\varphi}{ds} = \frac{-\sin\theta\sin\varphi}{r} = \frac{-\sqrt{1-\xi^2}}{r}\sin\varphi$$

and

$$dz = \cos\theta \, ds$$

or

$$\frac{dz}{ds} = \cos\theta = \xi$$

Also note that dr/ds corresponds to Ω_r, and dz/ds corresponds to Ω_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

$$\sqrt{1 - \xi^2} \cos\varphi \frac{\partial\Psi_g}{\partial r} - \frac{\sqrt{1 - \xi^2}}{r} \sin\varphi \frac{\partial\Psi_g}{\partial\varphi} + \xi \frac{\partial\Psi_g}{\partial z} + \Sigma_g^t \Psi_g = S_g \qquad (5)$$

with

$$\Psi_g \equiv \Psi_g(r, z, \xi, \varphi)$$

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

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

and

$$\frac{\partial(r\Psi)}{\partial r} = \Psi + r \frac{\partial\Psi}{\partial r}$$

Using these relations gives

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

and equation (5) becomes

$$\frac{\sqrt{1-\xi^2}}{r}\cos\varphi\frac{\partial(r\Psi_g)}{\partial r}-\frac{\sqrt{1-\xi^2}}{r}\frac{\partial\left[(\sin\varphi)\Psi_g\right]}{\partial\varphi}+\xi\frac{\partial\Psi_g}{\partial z}+\Sigma_g^t\Psi_g=S_g \tag{6}$$

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

$$\Psi(r_b, z, \xi, \varphi) = 0 \qquad 0 \le \varphi \le \frac{\pi}{2}$$

where φ is the angle in the plane perpendicular to the cylindrical axis between the unit vector $\hat{r}$ and the projection of $\hat{\Omega}$.

At the upper or lower boundary z_b,

$$\Psi(r, z_b, \xi, \varphi) = 0$$

with $-1 \le \xi \le 0$ at the upper boundary and $0 \le \xi \le 1$ at the lower boundary, where ξ is the cosine of the angle between the unit vectors $\hat{k}$ and $\hat{\Omega}$.

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

$$\Psi(r_b, z, \xi, \varphi) = \frac{\displaystyle\int_0^{\pi/2}\int_{-1}^{1}\sqrt{1-\xi'^2}\,(\cos\varphi')\Psi(r_b, z, \xi', \varphi')\,d\xi'\,d\varphi'}{\displaystyle\int_0^{\pi/2}\int_{-1}^{1}\sqrt{1-\xi'^2}\cos\varphi'\,d\xi'\,d\varphi'} \qquad \frac{\pi}{2} \le \varphi \le \pi$$

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 φ and ξ.

The 180^{o} rotational symmetry condition may be applied if rotating that portion of a system below one of its diagonals through 180^{o} 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)

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

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

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

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

$$P_{0,0}(\hat{\Omega}) = P_0(\xi) = 1$$

$$P_{1,0}(\hat{\Omega}) = P_1(\xi) = \xi \qquad \equiv \Omega_z$$

$$P_{1,1}(\hat{\Omega}) = \sqrt{1 - \xi^2}\cos\varphi \qquad \equiv \Omega_x$$

$$P_{1,-1}(\Omega) = \sqrt{1 - \xi^2}\sin\varphi \qquad \equiv \Omega_y$$

where

$$\hat{\Omega} = \sqrt{1 - \xi^2}\,(\cos\varphi)\hat{i} + \sqrt{1 - \xi^2}\,(\sin\varphi)\hat{j} + \xi\hat{k}$$

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

$$\Sigma^s_{g'\to g}(\vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega}) = \sum_{\ell=0}^{L} (2\ell + 1)\,\Sigma^{s\ell}_{g'\to g}(\vec{\rho})P_\ell(\hat{\Omega}' \cdot \hat{\Omega})$$

where

$$\Sigma^{s\ell}_{g'\to g}(\vec{\rho}) = \frac{1}{4\pi}\int \Sigma^s_{g'\to g}(\vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega})P_\ell(\hat{\Omega}' \cdot \hat{\Omega})d^2\hat{\Omega}$$

and where $\Sigma^s_{g'\to g}(\vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega})$ 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

$$S_g^L(\vec{\rho}, \hat{\Omega}) = \sum_{g'=1}^{G} \left(\frac{1}{4\pi}\right) \int \Sigma_{g' \to g}^{s}(\vec{\rho}, \hat{\Omega}' \cdot \hat{\Omega}) \Psi_{g'}(\vec{\rho}, \hat{\Omega}') d\hat{\Omega}'$$

$$= \sum_{g'=1}^{G} \sum_{\ell=0}^{L} (2\ell + 1) \Sigma_{g' \to g}^{s\ell}(\vec{\rho}) \sum_{m=-\ell}^{\ell} f_{g'\ell m}(\vec{\rho}) P_{\ell m}(\hat{\Omega}) \qquad (7)$$

where $L = 0$ for P_0 scattering and $L = 1$ for P_1 scattering.

If cross sections for inelastic and $n \to 2n$ scattering are available, they may be included by adding them to $\Sigma_{g' \to g}^{s0}$ to obtain a total scattering matrix defined by

$$\Sigma_{g' \to g}^{st}(\vec{\rho}) = \Sigma_{g' \to g}^{s0}(\vec{\rho}) + \Sigma_{g' \to g}^{in}(\vec{\rho}) + 2\Sigma_{g' \to g}^{n \to 2n}(\vec{\rho})$$

The group scattering cross section $\Sigma_g^s(\vec{\rho})$ becomes

$$\Sigma_g^s(\vec{\rho}) = \sum_{g'=1}^{G} \Sigma_{g \to g'}^{s0}(\vec{\rho}) + \Sigma_{g \to g'}^{in}(\vec{\rho}) + \Sigma_{g \to g'}^{n \to 2n}(\vec{\rho})$$

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

For cylindrical geometry, $\Psi_g(\vec{\rho}, \hat{\Omega})$ is even in φ. The only $P_{\ell m}(\hat{\Omega})$ that is

odd in φ, 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,

$$J_{z_g}(\vec{\rho}) \equiv f_{g,1,0}(\vec{\rho})$$

and similarly for $f_{g,1,1}$, the identification

$$J_{r_g}(\vec{\rho}) \equiv f_{g,1,1}(\vec{\rho})$$

results.

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

$$S_g^0(\vec{\rho}, \hat{\Omega}) = \sum_{g'=1}^{G} \Sigma_{g' \to g}^{st}(\vec{\rho}) \Phi_{g'}(\vec{\rho}) \tag{8}$$

and, for $L = 1$

$$S_g^1(\vec{\rho}, \hat{\Omega}) = \sum_{g'=1}^{G} \left\{ \Sigma_{g' \to g}^{st}(\vec{\rho}) \Phi_{g'}(\vec{\rho}) + 3\Sigma_{g' \to g}^{s1}(\vec{\rho}) \left[\xi J_{z_{g'}}(\vec{\rho}) + \sqrt{1 - \xi^2}\, (\cos \varphi) J_{r_{g'}}(\vec{\rho}) \right] \right\} \tag{9}$$

and equation (4) becomes

$$S_g(\vec{\rho}, \hat{\Omega}) = \frac{\chi_g}{k_{eff}} \sum_{g'=1}^{G} \nu\Sigma_{g'}^{f}(\vec{\rho}) \Phi_{g'}(\vec{\rho}) + S_g^{\ell}(\vec{\rho}, \hat{\Omega}) \tag{10}$$

Equation (9) may be generalized for other geometries by recognizing that $\xi = \Omega_z$ and that $\sqrt{1 - \xi^2} \cos \varphi = \Omega_r$. Then equation (9) becomes

$$S_g^1(\vec{\rho}, \hat{\Omega}) = \sum_{g'=1}^{G} \left[\Sigma_{g' \to g}^{st}(\vec{\rho}) \Phi_{g'}(\vec{\rho}) + 3\Sigma_{g' \to g}^{s1}(\vec{\rho}) \sum_{q} \Omega_q J_{q_{g'}}(\vec{\rho}) \right] \tag{11}$$

for each direction q of the coordinate system.

The source $S_g(\vec{\rho}, \hat{\Omega})$ 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

$$\left.\begin{aligned} F(\vec{\rho}) &= \sum_{g'=1}^{G} \nu\Sigma_{g'}^{f}(\vec{\rho}) \Phi_{g'}(\vec{\rho}) \\ T_g^t(\vec{\rho}) &= \sum_{\substack{g'=1 \\ g' \neq g}}^{G} \Sigma_{g' \to g}^{st}(\vec{\rho}) \Phi_{g'}(\vec{\rho}) \\ T_g^1(\vec{\rho}, \hat{\Omega}) &= 3 \sum_{\substack{g'=1 \\ g' \neq g}}^{G} \Sigma_{g' \to g}^{s1}(\vec{\rho}) \sum_{q} \Omega_q J_{q_{g'}}(\vec{\rho}) \end{aligned}\right\} \tag{12}$$

Then

$$S_g(\vec{\rho}, \hat{\Omega}) = \frac{\chi_g}{k_{eff}} F(\vec{\rho}) + T_g^t(\vec{\rho}) + T_g^1(\vec{\rho}, \hat{\Omega}) + \Sigma_{g\to g}^{st}(\vec{\rho})\Phi_g(\vec{\rho}) + 3\Sigma_{g\to g}^{s1}(\vec{\rho}) \sum_q \Omega_q J_{q_g}(\vec{\rho}) \qquad (13)$$

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:

$$S_g(\vec{\rho}) = \frac{\chi_g}{k_{eff}} F(\vec{\rho}) + T_g^t(\vec{\rho}) + \Sigma_{g\to g}^{st}(\vec{\rho})\Phi_g(\vec{\rho}) \qquad (14)$$

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

$$\Sigma_g^{s1} = \sum_{g'=1}^{G} \Sigma_{g\to g'}^{s1}$$

and Σ_g^{tr} is defined to be $\Sigma_g^t - \Sigma_g^{s1}$.

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

$$\Sigma_g^r = \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \left(\Sigma_{g\to g'}^{s0} + \Sigma_{g\to g'}^{in} + \Sigma_{g\to g'}^{n\to 2n}\right)$$

where Σ_g^a is the cross section for neutron absorption including absorptions that result in fission. Under these circumstances, $\Sigma_{g\to g}^{st}$ may become negative.

For an adjoint problem (refs. 7 and 10), the source is developed differently resulting in

$$S_g^*(\vec{\rho}, \hat{\Omega}) = \nu\Sigma_g^f F^*(\vec{\rho}) + T_g^{t*}(\vec{\rho}) + T_g^{1*}(\vec{\rho}, \hat{\Omega}) + \Sigma_{g\to g}^{st}(\vec{\rho})\Phi_g^*(\vec{\rho}) + 3\Sigma_{g\to g}^{s1}(\vec{\rho}) \sum_q \Omega_q J_{q_g}^*(\vec{\rho}) \tag{15}$$

where

$$\left.\begin{aligned} F^*(\vec{\rho}) &= \sum_{g'=1}^{G} \frac{\chi_{g'}}{k_{eff}} \Phi_{g'}^*(\vec{\rho}) \\ T_g^{t*}(\vec{\rho}) &= \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \Sigma_{g\to g'}^{st}(\vec{\rho})\Phi_{g'}^*(\vec{\rho}) \\ T_g^{1*}(\vec{\rho}, \hat{\Omega}) &= 3 \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \Sigma_{g\to g'}^{s1}(\vec{\rho}) \sum_q \Omega_q J_{q_{g'}}^*(\vec{\rho}) \end{aligned}\right\} \tag{16}$$

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 $\vec{\rho}$ 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 $A_{ik} = 2\pi r_i \Delta z_k$ and $C_{ik} = \pi\left(r_{i+1}^2 - r_i^2\right)$ 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 ξ and φ, a two-dimensional partition of the surface of a unit sphere centered at each mesh cell is required. Since ψ and S are both even functions of φ, it is only necessary to consider the range of φ from zero to π. The directional variable ξ is also even and, if no z variation exists, ξ 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 ξ 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

$$\sum_{m,\ell} w_{m\ell} = 1 \tag{17a}$$

$$\sum_{m,\ell} w_{m\ell}\Omega_{m\ell} = 0 \tag{17b}$$

$$\sum_{m,\ell} w_{m\ell}\Omega_{m\ell}^2 = \frac{1}{3} \tag{17c}$$

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

$$\sum_{m,\ell} w_{m\ell}\left|\Omega_{m\ell}\right| = \frac{1}{2} \tag{17d}$$

also holds. Except for equation (17b), these are all forms of a general condition

$$\sum_{m,\ell} w_{m\ell} \left| \Omega^q_{m\ell} \right| = \frac{1}{q+1}$$

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 ξ. These arcs are defined by dividing the continuous variable ξ into a set of ξ_m. These bands are determined so that the areas of the bands, as ξ decreases, are na, $(n-2)a$, . . ., $4a$, $2a$. The sum of these areas must be π, the area of the first quadrant, or

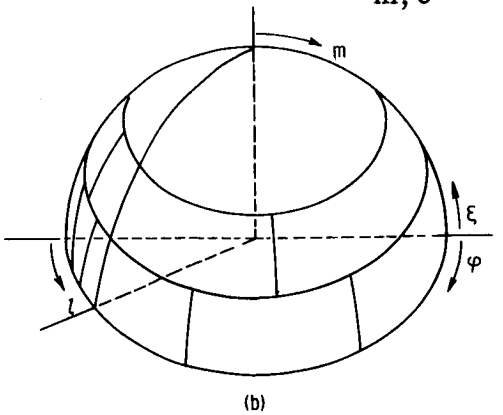
$$\sum_{m=1}^{n/2} 2ma = \pi$$

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 (φ constant) into sections of area a with the bands partitioned into $L = n, n-2, \ldots, 4, 2$ sections. Here, the continuous variable φ is divided into a set of boundaries $\varphi_{m\ell}$ so that

$$\varphi_{m,0} = \pi,\ \varphi_{m,L/2} = \frac{\pi}{2},\ \text{and}\ \varphi_{m,L} = 0$$



(b)

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

With each area is associated an average value $\bar{\xi}_m$ of ξ and $\bar{\varphi}_{m\ell}$ of φ. The average direction $(\bar{\xi}_m, \bar{\varphi}_{m\ell})$ is chosen so that condition (17c) holds (ref. 3). In the quadrature system being considered, this direction is obtained by defining $\bar{\varphi}_{m\ell}$ to be the simple average

$$\frac{\varphi_{m\ell} + \varphi_{m,\ell-1}}{2}$$

and with equal weights $w_{m\ell}$ (representing the areas), while $\bar{\xi}_m$ can be defined by

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

and

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

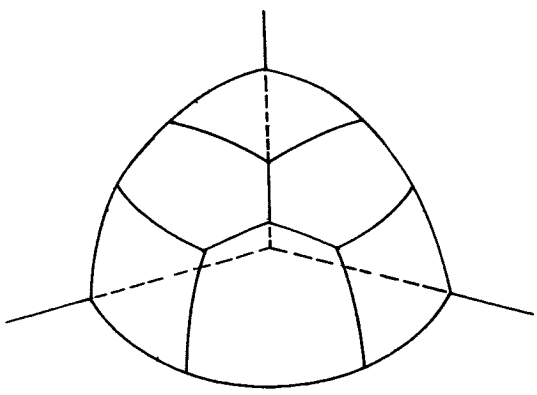
where

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

and

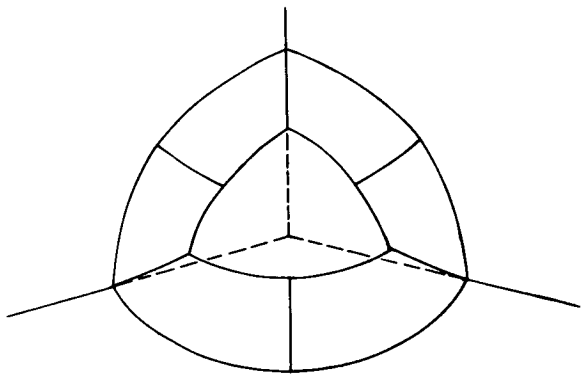
$$\xi_m = -\xi_{n-m} \qquad m = \frac{n}{2} + 1, \ldots, n$$

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 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.



(c)

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.



(d)

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-invarant 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,ℓ 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 φ-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

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

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,ℓ, $\Omega_{r_{m\ell}}$ is the direction cosine of the angle between direction m,ℓ and the radial direction yielding the component of the directional flux normal to the radial surfaces, $A_{i+1,k}$ and A_{ik} are the surface areas at $(r_{i+1}, \bar{z}_k)$ and $(r_i, \bar{z}_k)$, and $\psi_{g,i+1,\bar{k},\bar{m},\bar{\ell}}$ and $\psi_{g,i,\bar{k},\bar{m},\bar{\ell}}$ represent the average directional flux over these surfaces.

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

$$w_{m\ell}\Omega_{z_{m\ell}} \frac{\left(C_{i,k+1}\psi_{g,\bar{i},k+1,\bar{m},\bar{\ell}} - C_{ik}\psi_{g,\bar{i},k,\bar{m},\bar{\ell}}\right)}{V_{ik}}$$

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 φ-direction from the quadrature segment but not a loss from the mesh cell. In difference form this term may be written as

$$\frac{\alpha_{m\ell}\psi_{g,\bar{i},\bar{k},\bar{m},\ell} - \alpha_{m,\ell-1}\psi_{g,\bar{i},\bar{k},\bar{m},\ell-1}}{V_{ik}}$$

where $\psi_{g,\bar{i},\bar{k},\bar{m},\ell}$ and $\psi_{g,\bar{i},\bar{k},\bar{m},\ell-1}$ represent average fluxes for the quadrature segment boundaries of direction m,ℓ. The quantities $\alpha_{m\ell}$ and $\alpha_{m,\ell-1}$ are to be determined so that the net angular redistribution in the φ-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:

$$w_{m\ell}\Omega r_{m\ell}(A_{i+1,k}\psi_{g,i+1,\bar{k},\bar{m},\bar{\ell}} - A_{ik}\psi_{g,i,\bar{k},\bar{m},\bar{\ell}}) + (\alpha_{m\ell}\psi_{g,\bar{i},\bar{k},\bar{m},\ell} - \alpha_{m,\ell-1}\psi_{g,\bar{i},\bar{k},\bar{m},\ell-1}) + w_{m\ell}\Omega z_{m\ell}(C_{i,k+1}\psi_{g,\bar{i},k+1,\bar{m},\bar{\ell}} - C_{ik}\psi_{g,\bar{i},k,\bar{m},\bar{\ell}}) + w_{m\ell}\Sigma^{t}_{g,i,k}V_{ik}\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} = w_{m\ell}S_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}}V_{ik} \qquad (18)$$

The $\alpha_{m\ell}$ coefficients must now be evaluated. Because of the cylindrical symmetry, the range of φ to be considered is from zero to π and there cannot be angular redistribution to values of φ 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),

$$w_{m\ell}\Omega r_{m\ell}(A_{i+1,k} - A_{ik}) + (\alpha_{m\ell} - \alpha_{m,\ell-1}) + w_{m\ell}\Omega z_{m\ell}(C_{i,k+1} - C_{ik}) = 0$$

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 ξ-direction.

The conditions on $\alpha_{m\ell}$ thus reduce to

$$\left.\begin{aligned} \alpha_{m,0} &= \alpha_{m,L} = 0 \\ \alpha_{m\ell} - \alpha_{m,\ell-1} &= -w_{m\ell}\Omega r_{m\ell}(A_{i+1,k} - A_{ik}) \end{aligned}\right\} \qquad (19)$$

Another expression for $\alpha_{m\ell}$ can be found by noting that with

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

and

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

$$\alpha_{m,2} = -\left(w_{m,2}\Omega r_{m,2} + w_{m,1}\Omega r_{m,1}\right)(A_{i+1,k} - A_{ik})$$

Substituting $\alpha_{m,2}$ and $\alpha_{m,1}$ into (19) gives

$$\alpha_{m,3} = -\left(w_{m,3}\Omega r_{m,3} + w_{m,2}\Omega r_{m,2} + w_{m,1}\Omega r_{m,1}\right)(A_{i+1,k} - A_{ik})$$

or, in general,

$$\alpha_{m\ell} = -\left(w_{m\ell}\Omega r_{m\ell} + w_{m,\ell-1}\Omega r_{m,\ell-1} + \cdots + w_{m,1}\Omega r_{m,1}\right)(A_{i+1,k} - A_{ik}) \qquad (20)$$

The equations represented by (18) (one equation for each mesh cell i,k and discrete direction m,ℓ and group g) do not couple different values of $\Omega_{z_{m\ell}}$, 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 $\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}}$ 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

$$\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} = \psi_{g,i+1,\bar{k},\bar{m},\bar{\ell}} = \psi_{g,\bar{i},k+1,\bar{m},\bar{\ell}} = \psi_{g,\bar{i},\bar{k},\bar{m},\ell}$$

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

$$\begin{aligned}\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} &= \frac{1}{2}\left(\psi_{g,i+1,\bar{k},\bar{m},\bar{\ell}} + \psi_{g,i,\bar{k},\bar{m},\bar{\ell}}\right)\\ &= \frac{1}{2}\left(\psi_{g,\bar{i},k+1,\bar{m},\bar{\ell}} + \psi_{g,\bar{i},k,\bar{m},\bar{\ell}}\right)\\ &= \frac{1}{2}\left(\psi_{g,\bar{i},\bar{k},\bar{m},\ell} + \psi_{g,\bar{i},\bar{k},\bar{m},\ell-1}\right)\end{aligned} \tag{21}$$

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

$$\begin{aligned}&w_{m\ell}\Omega r_{m\ell}\left[A_{i+1,k}(2\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} - \psi_{g,i,\bar{k},\bar{m},\bar{\ell}}) - A_{ik}\psi_{g,i,\bar{k},\bar{m},\bar{\ell}}\right] + \left[\alpha_{m\ell}(2\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} - \psi_{g,\bar{i},\bar{k},\bar{m},\ell-1}) - \alpha_{m,\ell-1}\psi_{g,\bar{i},\bar{k},\bar{m},\ell-1}\right]\\ &+ w_{m\ell}\Omega z_{m\ell}\left[C_{i,k+1}(2\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} - \psi_{g,\bar{i},k,\bar{m},\bar{\ell}}) - C_{ik}\psi_{g,\bar{i},k,\bar{m},\bar{\ell}}\right]\\ &+ w_{m\ell}\Sigma^{t}_{g,i,k}V_{ik}\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} = w_{m\ell}S_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}}V_{ik}\end{aligned}$$

or

$$\begin{aligned}&2w_{m\ell}\Omega r_{m\ell}A_{i+1,k}\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} - w_{m\ell}\Omega r_{m\ell}(A_{i+1,k} + A_{ik})\psi_{g,i,\bar{k},\bar{m},\bar{\ell}}\\ &+ 2\alpha_{m\ell}\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} - (\alpha_{m\ell} + \alpha_{m,\ell-1})\psi_{g,\bar{i},\bar{k},\bar{m},\ell-1}\\ &+ w_{m\ell}\Omega z_{m\ell}(C_{i,k+1} + C_{ik})\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} - w_{m\ell}\Omega z_{m\ell}(C_{i,k+1} + C_{ik})\psi_{g,\bar{i},k,\bar{m},\bar{\ell}}\\ &+ w_{m\ell}\Sigma^{t}_{g,i,k}V_{ik}\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} = w_{m\ell}S_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}}V_{ik}\end{aligned}$$

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

$$\alpha_{m\ell} + \alpha_{m,\ell-1} = w_{m\ell}\gamma_{m\ell}(A_{i+1,k} - A_{ik}) \tag{22}$$

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 $\ell = 1$,

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

and

$$\gamma_{m,1} = -\Omega r_{m,1}$$

For $\ell = 2$ and using (20) to substitute for $\alpha_{m,\ell-1}$

$$w_{m,2}\gamma_{m,2} = -w_{m,2}\Omega_{r_{m,2}} - w_{m,1}\Omega_{r_{m,1}} - w_{m,1}\Omega_{r_{m,1}}$$

and since $\gamma_{m,1} = -\Omega_{r_{m,1}}$

$$w_{m,2}\gamma_{m,2} - w_{m,1}\gamma_{m,1} = -(w_{m,2}\Omega_{r_{m,2}} + w_{m,1}\Omega_{r_{m,1}})$$

Continuing for $\ell > 2$, the relation

$$\left.\begin{aligned} \gamma_{m,1} &= -\Omega_{r_{m,1}} \\ w_{m,\ell}\gamma_{m,\ell} - w_{m,\ell-1}\gamma_{m,\ell-1} &= -\left(w_{m\ell}\Omega_{r_{m\ell}} + w_{m,\ell-1}\Omega_{r_{m,\ell-1}}\right) \quad \ell > 1 \end{aligned}\right\} \quad (23)$$

can be obtained.

Solving for $\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}}$ gives the difference equation as

$$\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} = \frac{\left[\Omega_{r_{m\ell}}(A_{i+1,k} + A_{ik})\psi_{g,i,\bar{k},\bar{m},\bar{\ell}} + \gamma_{m\ell}(A_{i+1,k} - A_{ik})\psi_{g,\bar{i},\bar{k},\bar{m},\ell-1} + \Omega_{z_{m\ell}}(C_{i,k+1} + C_{ik})\psi_{g,\bar{i},k,\bar{m},\bar{\ell}} + S_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}}V_{ik}\right]}{\left[\Omega_{r_{m\ell}}(A_{i+1,k} + A_{ik}) + \gamma_{m\ell}(A_{i+1,k} - A_{ik}) + \Omega_{z_{m\ell}}(C_{i,k+1} + C_{ik}) + \Sigma^{t}_{g,i,k}V_{ik}\right]} \quad (24)$$

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 $\psi_{g,i,\bar{k},\bar{m},\bar{\ell}}$ or $\psi_{g,\bar{i},k,\bar{m},\bar{\ell}}$ are replaced in equation (18) with $\psi_{g,i+1,\bar{k},\bar{m},\bar{\ell}}$ or $\psi_{g,\bar{i},k+1,\bar{m},\bar{\ell}}$, 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 $\ell = 0$ sweep for which $\gamma_{m,0} = 0$ and $\Omega_{r_{m,0}} = -\sqrt{1 - \Omega^2_{z_{m,0}}}$. Equation (24) is used to obtain an average flux over mesh cell i, k for the discrete direction m, ℓ. Values to be used in the same manner for successive mesh cells and directions are then computed by using equation (21) with $\psi_{g,\bar{i},\bar{k},\bar{m},\ell} = \psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}}$ 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 Φ_{gik} can be obtained by integrating $\psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}}$ over the directions m and ℓ or

$$\Phi_{gik} = \sum_{m,\ell} w_{m\ell} \psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} \tag{25a}$$

while the currents are

$$J_{r_{gik}} = \sum_{m,\ell} w_{m\ell} \Omega_{r_{m\ell}} \psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} \tag{25b}$$

and

$$J_{z_{gik}} = \sum_{m,\ell} w_{m\ell} \Omega_{z_{m\ell}} \psi_{g,\bar{i},\bar{k},\bar{m},\bar{\ell}} \tag{25c}$$

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),

$$\frac{\chi_g}{k_{eff}} F + T_g^t + T_g^1$$

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

$$\nu\Sigma_g^f F^* + T_g^{t*} + T_g^{1*}$$

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

$$\Sigma_{g\to g}^{st}\Phi_g + 3\Sigma_{g\to g}^{s1}\sum_q \Omega_q J_{q_g}$$

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 $\Sigma_{g\to g}^{st}$ 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

$$F_{ik}^{P} = \sum_{g=1}^{G} \nu\Sigma_{gik}^{f}\Phi_{gik}$$

as in equation (12). Then the growth factor for production is obtained by the ratio

$$\lambda^P = \frac{\sum_{i,k} F_{ik}^P V_{ik}}{\sum_{i,k} F_{ik}^{P-1} V_{ik}}$$

The estimate of the multiplication factor k_{eff} resulting from the P^{th} iteration is then given by $k_{eff}^P = \lambda^P k_{eff}^{P-1}$. Using this new value of k_{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_{eff}, and for only one of these values, corresponding to the fundamental mode, will the fluxes be everywhere positive. The latest estimate of k_{eff} is used to obtain the fixed source estimate of the following power iteration. The power iterations are repeated until λ^P converges to 1 (indicating no change in the fission source for the iteration).

Several tests are applied to determine the convergence of λ^P. Among these can be a test on the closeness of λ^P to 1, a test on the rate of change of λ, 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

$$f_g = \frac{\sum_{i,k} \left(\frac{\chi_g}{k_{eff}} F_{ik} + T_{gik}^t \right) V_{ik}}{\sum_{i,k} \left(\frac{\chi_g}{k_{eff}} F_{ik} + T_{gik}^t \right) V_{ik} - \sum_{i,k} \Sigma_{g \rightarrow g}^{st} \left(\Phi_{gik}^{new} - \Phi_{gik}^{old} \right) V_{ik}} \qquad (26)$$

where V_{ik} is the volume of the mesh cell i,k. The flux estimate Φ^{new}_{gik}, 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 Φ^{e}_{gik} for mesh interval i,k is given by

$$\Phi^{e}_{gik} = \omega\Phi^{p}_{gik} - (\omega - 1)\Phi^{p-1}_{gik} \tag{27}$$

where p is the present inner iteration. After the extrapolation is performed, Φ^{p}_{gik} will no longer be available to the program having been displaced by Φ^{e}_{gik}. In computing the scaling factor f_g, Φ^{e}_{gik} will become Φ^{new}_{gik} and Φ^{p-1}_{gik} will become Φ^{old}_{gik}. 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 $E_{ik} = \Phi^{e}_{gik} \Big/ \Phi^{p}_{gik}$. Since Φ^{p}_{gik} is no longer available, it must be obtained from the relation $\Phi^{p}_{gik} = \left[\Phi^{e}_{gik} + (\omega - 1)\Phi^{p-1}_{gik}\right] \Big/ \omega$. With f_g being the scaling factor and Φ^{s}_{gik} the scaled flux

$$\Phi^{s}_{gik} = f_g\Phi^{e}_{gik} = f_gE_{ik}\Phi^{p}_{gik} = f_gE_{ik}\,\frac{\Phi^{e}_{gik} + (\omega - 1)\Phi^{p-1}_{gik}}{\omega}$$

and

$$E_{ik} = \frac{\omega \Phi^{e}_{gik}}{\Phi^{e}_{gik} + (\omega - 1)\Phi^{p-1}_{gik}}$$

Using E_{ik} gives the scaled extrapolated currents as

$$J^{s}_{r_{gik}} = f_g E_{ik} J^{p}_{r_{gik}}$$

and

$$J^{s}_{z_{gik}} = f_g E_{ik} J^{p}_{z_{gik}}$$

and similarly the angular boundary fluxes.

The best value of the acceleration factor ω is usually not known (ref. 13). Therefore, estimates of ω 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 ω^{p+1} are computed as convergence proceeds that depend on the original factors ω_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

$$\sigma^2 = \frac{\sum_{i,k} \left(\left| \Phi^{p}_{gik} - \Phi^{p-1}_{gik} \right| V_{ik} \right)}{\sum_{i,k} \left(\left| \Phi^{p-1}_{gik} - \Phi^{p-2}_{gik} \right| V_{ik} \right)}$$

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 $\Phi^{e}_{gik} = \omega\Phi^{P}_{gik} - (\omega - 1)\Phi^{P-1}_{gik}$. 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 ω 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

$$f^{up} = \frac{\sum_{g=1}^{G} \sum_{i,k} \frac{\chi_g}{k_{eff}} F_{ik} V_{ik}}{\sum_{g=1}^{G} \sum_{i,k} \frac{\chi_g}{k_{eff}} F_{ik} V_{ik} - \sum_{g=1}^{G} \sum_{i,k} \Sigma^{up}_{g} \left(\Phi^{new}_{gik} - \Phi^{old}_{gik} \right) V_{ik}} \tag{29}$$

where Σ^{up}_{g} is the total up-scattering removal cross section for group g, and Φ^{old}_{gik} and Φ^{new}_{gik} 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 δ_1 is the first quantity that must satisfy a convergence test and is written as

$$\delta_1 = \frac{\left| \Sigma^{st}_{g \to g} \left(\Phi^{p}_{gik} - \Phi^{p-1}_{gik} \right) V_{ik} \right|_{max}}{\sum_{i,k} \Phi^{p}_{gik} V_{ik}} \tag{30}$$

Second, two integrated quantities are determined. One of these quantities δ_2 is the ratio of the integrated change in the within-group-scattering source occuring as a result of an inner iteration to the total integrated group flux:

$$\delta_2 = \frac{\sum_{i,k} \left| \Sigma^{st}_{g \to g} \left(\Phi^{p}_{gik} - \Phi^{p-1}_{gik} \right) V_{ik} \right|}{\sum_{i,k} \Phi^{p}_{gik} V_{ik}} \tag{31}$$

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

$$\delta_3 = \frac{\sum_{i,k} \left| \left(\Sigma^{t}_{g} - \Sigma^{st}_{g \to g} \right) \left(\Phi^{p}_{gik} - \Phi^{p-1}_{gik} \right) V_{ik} \right|}{\sum_{i,k} \Phi^{p}_{gik} V_{ik}} \tag{32}$$

Both δ_2 and δ_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 λ^P must not differ from 1 or its previous value λ^{P-1} by more than some specified amount ϵ:

$$\delta_4 = \left|\lambda^P - 1\right| < \epsilon \quad \text{and} \quad \delta_5 = \left|\lambda^P - \lambda^{P-1}\right| < \epsilon \tag{33}$$

Third, the change δ_6 in the total integrated scattering loss from one iteration to the next, as given by the ratio

$$\delta_6 = \frac{\sum_{g=1}^{G} \sum_{i,k} \Sigma^r_{gik} \Phi^{P-1}_{gik} V_{ik}}{\sum_{g=1}^{G} \sum_{i,k} \Sigma^r_{gik} \Phi^{P}_{gik} V_{ik}} < \epsilon \tag{34}$$

must satisfy a convergence test. For some problems, for which there is no $\Sigma^{n\rightarrow 2n}_{gik}$, the scattering loss must match the scattering source. Also, if up-scattering scaling is used, the ratio δ_7 of the total integrated change in the up-scattering source as summed over all groups to the total integrated flux must be small:

$$\delta_7 = \frac{\sum_{g=1}^{G} \sum_{i,k} \Sigma^{up}_{gik} \left|\Phi^{P}_{gik} - \Phi^{P-1}_{gik}\right| V_{ik}}{\sum_{g=1}^{G} \sum_{i,k} \Phi^{P}_{gik} V_{ik}} < \epsilon \tag{35}$$

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

Group	Cross sections							
	Σ^a	$\nu\Sigma^f$	Σ^{tr}	$\Sigma_{g+2\rightarrow g}$	$\Sigma_{g+1\rightarrow g}$	$\Sigma_{g\rightarrow g}$	$\Sigma_{g-1\rightarrow g}$	$\Sigma_{g-2\rightarrow g}$
1	0.0	0.1	1.0	0.333333	0.333333	0.333333	0.0	--------
2	0.0	0.1	1.0	0.0	.333333	.333333	.333333	0.0
3	0.0	0.1	1.0	0.0	0.0	.333333	.333333	.333333

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

Number of neutrons emitted per collision, c	Quadrature type, S_{16}	Sphere			Plane			Cylinder		
		Solution		Deviation, δ, percent	Solution		Deviation, δ, percent	Solution		Deviation, δ, percent
		TDSN	Exact (a)		TDSN	Exact (a)		TDSN	Exact (a)	
1.1	One dimensional	4.8723	4.8727	-0.01	2.1153	2.1134	0.09	------	------	-----
	Two dimensional	------	------	-----	2.1146	2.1134	.06	3.5762	3.5783	-0.06
1.6	One dimensional	1.4753	1.4761	-.05	.5132	.5120	.24	------	------	-----
	Two dimensional	------	------	-----	.5139	.5120	.38	1.0191	1.0209	-.18

[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

$$c = \frac{\left(\nu\Sigma_g^f + \Sigma_g^s\right)}{\Sigma_g^t}$$

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 $\nu\Sigma^f$ of 0.6. Table II gives the percent derivation δ 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 δ 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

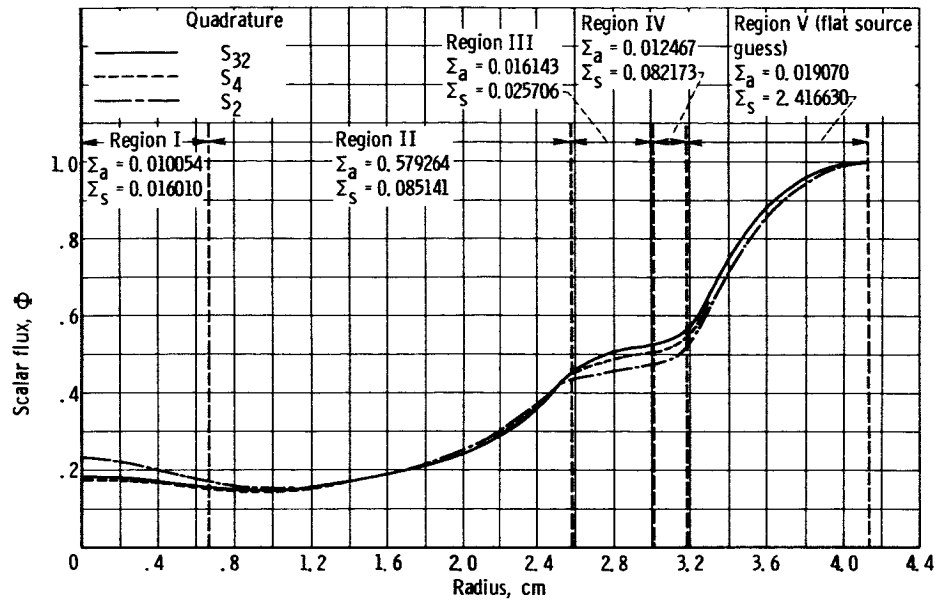


Figure 1. - Relative scalar flux as function of radius for one-group cylindrical-cell problem.

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 ω 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

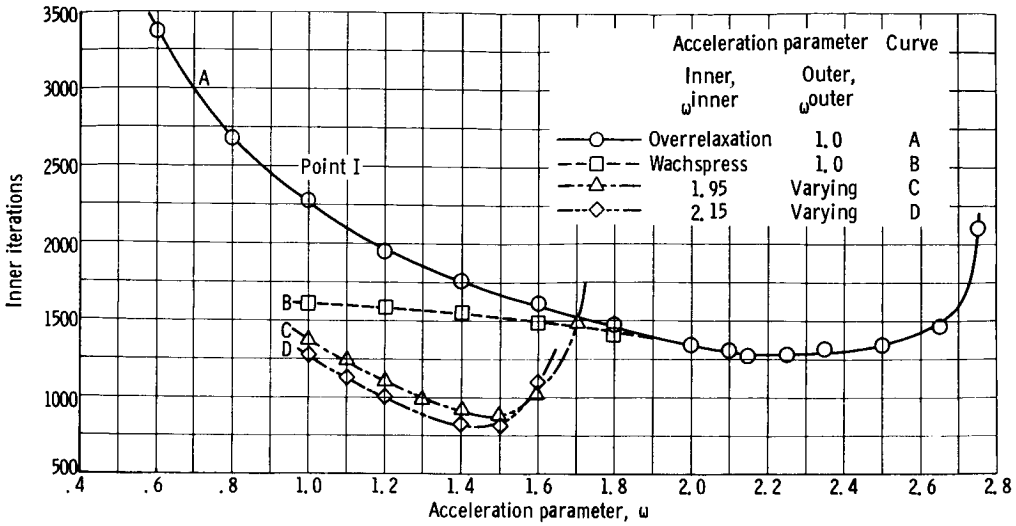


Figure 2. - Number of inner iterations as function of acceleration parameter. Thermal reactor with three groups and three regions.

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 ω 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.

Quantity	Format	Card Columns	Remarks
KBCRED	I10	1-10 = 0	Read in cards III to XIX (new problem).
		= 1	Read in BCDUMP for restart instead of cards III to XIX. [1]
		= 2	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.
		= 3	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.

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.

KREG	I10	1-10 = 0	A regular flux calculation is performed.
		= 1	An adjoint calculation is performed.

Quantity	Format	Card columns	Remarks
KALC	I10	11-20 = 1	Calculate the multiplication constant k_{eff} as the eigenvalue (only option available).
KGEO	I10	21-30 = 1	Slab x, y geometry
		= 2	Cylindrical r, z geometry
		= 3	Sphere r geometry
KLBC	I10	31-40 = 0	No return current, left boundary condition
		= 1	Plane mirror ("perfect") reflection, left boundary condition
KRBC	I10	41-50 = 0, 1	Same as KLBC for right boundary condition
		= 3	Isotropic ("white") reflection, right boundary condition [2]
KBBC	I10	51-60 = 0, 1	Same as KLBC for bottom boundary condition
		= 4	180^0 rotational symmetry, bottom boundary condition [2]
KTBC	I10	61-70 = 0, 1, 3	Same as KRBC for top boundary condition

IV. Control card.

NSN	I10	1-10	This is the order n of S_n to be used. It must be an even integer.
NG	I10	11-20	The number of energy groups
NR	I10	21-30	The number of mesh intervals for the first direction [3]
NZ	I10	31-40 = 0	A one-dimensional calculation will be performed with one-dimensional quadrature (for slabs and spheres only).
		= 1	A one-dimensional calculation will be performed with two-dimensional quadrature (for slabs and cylinders only).
		> 1	The number of mesh intervals for the second direction of a two-dimensional calculation [3]
NMAT	I10	41-50	The number of different materials to be used [4]
NTYPS	I10	51-60	The number of cross section types per material per group [5]

Quantity	Format	Card columns	Remarks
N2N	I10	61-70 = 1	Read in total out-scattering cross sections along with the other cross sections for KCTYP = 1 or at XVI(4) for KCTYP $\neq$ 1. [6]
V. Control card.			
KCTYP	I10	1-10 = 0	The cross sections will be in TDSN format [5]
		= 1	The cross sections to be supplied in the format of the RP1 diffusion-theory program [7]
		= 2	The cross sections to be supplied in the format of the DTF and DDF programs [5]
KSISO	I10	11-20 = 0	All materials use only P_0 cross sections.
		= 1	At least one material uses P_1 cross sections.
KGST	I10	21-30	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.
KCTR	I10	31-40	The position of Σ_g^t (or Σ_g^{tr}) for g = 1 in the cross-section matrix. [5]
KCGG	I10	41-50	The position of $\Sigma_{g\rightarrow g}^{st}$ in the cross section matrix. [5]
NBUCK	I10	51-60 = 0	No buckling of this type is to be used.
		= -1	Buckling dimensions to be read in by card X and the KCTR field of P_1 scattering cross section sets should contain Σ_g^{tr}. [5] and [8]
		= 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 $\neq$ 1.
LBUCK	I10	61-70 = 0	No buckling of this type is to be used.

Quantity	Format	Card columns	Remarks
		= -1	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.
		= 1	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.
VI. Control card.			
ITMIMX	I10	1-10	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]
ITMIGM	I10	11-20	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]
NZONER	I10	21-30	The number of input zones for the first direction [12]
NZONEZ	I10	31-40	The number of input zones for the second direction (zero if NZ = 0 or 1) [12]
KDREAD	I10	41-50 = 0	Use moment modified quadrature for direction cosines and weights as calculated by program.
		= 1	Read-in direction cosines and weights at XV.
KFLUXI	I10	51-60 = 0	Use an initial flux guess of constant value over all mesh intervals and groups as supplied by the program.
		= -1	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]

Quantity	Format	Card columns	Remarks
		= 1	Supply an initial flux guess at XIX(2)
KALC1	I10	61-70	To be used for KALC > 1 (not available at present).
VII. Control card.			
GLAM	E10.6	1-10	The value of λ to be used for problems other than KALC = 1. On restart, the final value of λ of the previous run is given in the output.
EIGEN	E10.6	11-20	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.
EPS	E10.6	21-30	The convergence criteria for the outer iterations [10]
XNF	E10.6	31-40	The normalization factor; the number of fission neutrons at the beginning of an outer iteration is normalized to XNF.
RYF	E10.6	41-50	The ratio factor used in one of the convergence checks as listed in note [10]
TIMAX	E10.6	51-60	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]
CALC1	E10.6	61-70	To be used in conjunction with KALC1.
VIII. Control card.			
OMEGA	E10.6	1-10	The relaxation parameter to be used for inner iterations. It is not used unless KACCEL $\neq$ 0. OMEGA > 1 for overrelaxation. [11]

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

Quantity	Format	Card columns		Remarks
			= 1	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]
			= 2	Same as KBCDUP = 1, except that the card dump will be given even if the problem is converged. [1] and [10]
KACAV	I10	21-30	= 0	No activity or averaging tables is to be given in the output.
			= -1, 1	Activity tables will be given in the output along with disadvantage factors.
			= -2, 2	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]
			= -3, 3	Both sets of tables will be given

If KACAV > 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 < 0, a separate zone map for each type of table requested must be read in at XXIV.

KGAVE	I10	31-40	= 0	No group collapsed cross sections will be given.
			> 0	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 $\Sigma^{st}_{g \to g}$ will not be correct if n → 2n scattering is present (only given if KREG = 0). [13]
KNIP	I10	41-50	= 0	No flux integrals are punched or effective axial bucklings printed in the output.
			= 1	The fluxes as integrated for each material are printed and punched by group for each material.

Quantity	Format	Card columns	Remarks
		= 2	Same as KNIP = 1 plus the axial leakage probability per unit flux (axial leakage/flux) is printed by group for a two-dimensional problem.
KEDIT	I10	51-60 = 0	EDIT subroutine will not be used. The EDIT subroutine gives activities, flux integrals, and flux-averaged cross sections over groups and mesh intervals to be specified at XXV(2) for cross sections as specified by KEDIT.
		= 1	Cross sections by group to be used by the EDIT subroutine must be read in at XXV(3)
		= 2	The EDIT subroutine will use the absorption cross section of the problem.
		= 3	The fission neutron cross section of the problem will be used.
		= 4	The total (or transport) cross section of the problem will be used.

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.

Quantity	Format	Card columns	Remarks
NMI	I5	1-5	The number of mesh intervals in the zone
RMI	E10.6	6-15	The distance from the bottom boundary of the problem to the upper boundary of the zone

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	7I10	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 χ 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 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:

Quantity	Format	Card columns	Remarks
D	E10.6	1-10	The diffusion coefficient
Σ^a	E10.6	11-20	The absorption cross section that will be put in the C^a position.
Σ^r	E10.6	21-30	If (NTYPS - KCGG) = 1, this is the scattering removal cross section and will be put into the first (and only) C^{down} position. If N2N = 1, this is the total out-scattering cross section CN2N. [6]
$\nu\Sigma^f$	E10.6	31-40	The fission neutron cross section that will be put in the C^f position
BUCKG	E10.6	41-50	If NBUCK 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 NDOWN = NTYPS - KCGG is greater than 1, or if NUP $>$ 0, the second set of cards consists of the down-scattering cross sections for the material and are included here.

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

(c) If NUP = KCGG - (KCTR + 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 - NUP + 1)-th group to the NS = (NG - NUP)-th group.

Σ^{up}	7E10.6	1-10,*	$\Sigma_{g\to g-1}$, $\Sigma_{g\to g-2}$, etc., until $\Sigma_{g\to NS}$ 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 $\Sigma_{g\to g}$ is computed for each material and put in the $C^{g\to g}$ position. No activity cross sections can be included. [7]

(3) If KCTYP = 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 KCTR = 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 $\neq$ 1, the total out scattering cross sections are supplied here before the next material is included under (1) or (3).

Quantity	Format	Card columns	Remarks
CN2N	7E10.6	1-10, *	NG values [6]

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 $\neq$ 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 P_1 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.

Quantity	Format	Card columns	Remarks
ITMIMX	I10	1-10	Same as for card VI [11]
ITMIGM	I10	11-20	Same as for card VI [11]

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

EPS	E10.6	1-10	Same as for card VII
TIMAX	E10.6	11-20	Same as for card VII [11]
OMEGA	E10.6	21-30	Same as for card VIII [11]
OMEGAP	E10.6	31-40	Same as for card VIII [11]
KACCEL	I10	51-60	Same as for card VIII [11]
KUPS	I10	61-70	Same as for card VIII

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.

IDGP	7I10	1-10, *	Include NG numbers identifying the KGAVE collapsed group to which each group belongs.

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

NZONRA	I10	1-10	The number of output zones in the first direction (maximum of 50). [12]
NZONZA	I10	11-20	The number of output zones in the second direction (zero for a one-dimensional problem; (maximum of 30). [12]

(b) Mesh interval map.

Quantity	Format	Card columns	Remarks
NMRA	7I10	1-10, *	The number of mesh intervals for the first direction to be included in each output zone. Include NZONRA values.

(c) Include this card if NZONZA > 0.

NMZA	7I10	1-10, *	The number of mesh intervals for the second direction to be included in each output zone. Include NZONZA values.

(d) The identification numbers for the zones. Include NZONZA sets of cards (one set if NZONZA = 0).

IDM	14I5	1-5, *	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.

(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).

XXV. Input for EDIT subroutine if KEDIT > 0.

NISET	I10	1-10	The number of activity edits (integration sets) to be performed.

(1) The number of integrals to include in a given edit set (INT has a maximum of 200).

INT	I10	1-10	This card will be read in NISET times.

Cards (2) and (3) are included in order INT times following each card (1).

(2) The integration limits for a given integral.

IG1	I10	1-10	The upper limit of the groups to be included in a given integral
IG2	I10	11-20	The lower limit of the groups
IR1	I10	21-30	The left limit of the first direction mesh intervals to be included in a given integral

Quantity	Format	Card columns	Remarks
IR2	I10	31-40	The right limit
IZ1	I10	41-50	The lower limit of the second direction mesh intervals to be included in a given integral (one for a one-dimensional problem).
IZ2	I10	51-60	The upper limit (one for a one-dimensional problem)

(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 χ, XKI.

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:

ITMA	Number of outer iteration
ITMI	Total number of inner iterations performed [11]
ITMIGX	Maximum number of inner iterations performed in a group during outer iteration
ITD	Number of inner iterations per outer iteration
TIME REQ	Total time required to perform iterations in minutes; time starts over on restart [11]
EPG	Convergence criterion for inner iterations [9]
EPGM	Maximum change in within-group-scattering source (or in the collision removal) that has occurred for last inner iteration in a group [9]
KASOR	If KSISO = 1, KASOR > 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
KONV	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]
KFINSH	Number of groups that did not converge during the inner iterations (see XIV. [9] and [11])
NEUTRON BAL	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
GLAM	Eigenvalue λ that must be within EPS of 1 for convergence, and within EPS of previous value of λ
EIGEN	k_{eff}

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 δ_2 and δ_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 $\sum_{i,k} \left\lvert \Phi^{p}_{gik} - \Phi^{p-1}_{gik} \right\rvert V_{ik}$
ERROR RATIO	If KACCEL < 0, this is σ^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 α of equation (28) that is used to compute new overrelaxation factor ω^{p+1}.
OMEGA	Value of overrelaxation factor ω^{p+1} to be used for next inner iteration in group

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 ESIJM occurred for the last inner iteration is listed under MXVARI. [10]

XVII. This mesh interval listing is followed by the values of ESIJM 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 Φ 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 $\Sigma\Phi V$ (ΣJV, if P_1) for each region listed and for each cross section Σ 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 Φ 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

$$f_{g,m} = \frac{\left(\sum_{i,k}^{m} \Phi_{gik} V_{ik} \Big/ \sum_{i,k}^{m} V_{ik} \right)}{\left(\sum_{i,k} \Phi_{gik} V_{ik} \Big/ \sum_{i,k} V_{ik} \right)}$$

where $\sum_{i,k}^{m}$ 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 Σ^t obtained will be a combination of the Σ^t of the P_1 region and the Σ^{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.

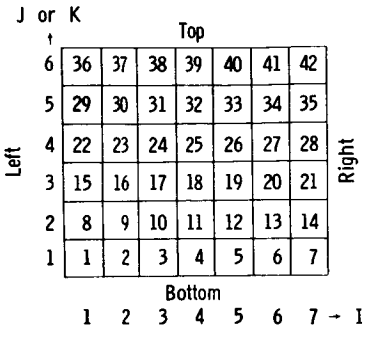


Figure 3. - Mesh layout showing mesh numbering scheme.

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 Σ_g^{act} and are only used in the program output in the tables in flux integrations such as

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

Following the activity cross sections (or the first in order if there are no activity cross sections) comes the absorption cross section Σ_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 ν times the fission cross section $\nu\Sigma_g^f$ followed by the total cross section Σ_g^t (or Σ_g^{tr} if the transport approximation has been used). The field containing Σ_g^t is designated by KCTR.

The Σ_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 Σ_g^t field unless a buckling cross section based on the reactor dimensions (NBUCK = -1) is to be obtained. Then this field should contain Σ_g^{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 $\Sigma_{g\to g}$ 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 NUP = KCGG - (KCTR + 1) with the $\Sigma_{g+1\to g}$ cross section occuring in the field just before $\Sigma_{g\to g}$, the $\Sigma_{g+NUP\to g}$ cross section occuring just after Σ_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 $\Sigma_{g-1\to g}$ cross section occuring in the field immediately following $\Sigma_{g\to g}$. 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 Σ_g^{act}, Σ_g^a, $\nu\Sigma_g^f$, Σ_g^t, and the total scattering array for all groups, while the second set of cross sections contains Σ_g^{act} (to be integrated over the current) and the P_1 scattering array and Σ_g^{tr} if NBUCK = -1. For KCTYP = 0, the $2\ell + 1$ multiplier must be included in the P_1 scattering matrix. For KCTYP = 2, the P_1 scattering matrix will be multiplied by $2\ell + 1$ by the program before being stored in the C array, and the $2\ell + 1$ multiplier must not be supplied in the cross-section matrix.

[6] If a problem contains materials that involve $n \to 2n$ scattering, the total scattering removal cross section Σ_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 $n \to 2n$ 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 Σ_g^r is defined by

$$\Sigma_g^r = \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \left(\Sigma_{g\to g'}^{s0} + \Sigma_{g\to g'}^{in} + \Sigma_{g\to g'}^{n\to 2n} \right)$$

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), Σ_g^{tr} and $\Sigma_{g\to g}$ must be computed and the whole array rearranged before being stored in the C array. Both these operations are performed by TDSN: Σ_g^{tr} is calculated as $1/(3D_g)$, while $\Sigma_{g\to g}$ is computed to give balance as $\Sigma_{g\to g} = \Sigma_g^{tr} - \Sigma_g^a - \Sigma_g^r$, where Σ_g^r is the scattering removal from the group. If there is no $n \to 2n$ scattering (N2N = 0), Σ_g^r is obtained by summing the down- and up-scattering

out of group g. If there is $n \rightarrow 2n$ 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 δ_2 and δ_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), δ_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 δ_2 and δ_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 δ_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 δ_4, δ_5, and δ_7 compared with EPS. However, a form of δ_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 maintained 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 convergence 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 proportional to the average number of neutron collisions per neutron lifetime (ref. 2). Convergence 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 $\overline{\Sigma}_g$ are obtained so that

$$\overline{\Sigma}_g = \frac{\sum_{i,k} \Sigma_{gik} \Phi_{gik} V_{ik}}{\sum_{i,k} \Phi_{gik} V_{ik}}$$

for those mesh intervals i,k over which the average is to be obtained. For the scattering array, the average cross section $\overline{\Sigma}_{g' \to g}$ is obtained by replacing Σ_{gik} by $\Sigma_{g' \to g, i, k}$ and Φ_{gik} by $\Phi_{g', i, k}$. The average scattering removal cross section is ob-

tained in the same way as $\bar{\Sigma}_g$, and the within-group-scattering cross section is obtained by neutron balance so that

$$\bar{\Sigma}_{g\to g} = \bar{\Sigma}^t_g - \bar{\Sigma}^a_g - \bar{\Sigma}^r_g$$

The cross section $\bar{\Sigma}_{g\to g}$ is obtained so that $\bar{\Sigma}^s_g = \bar{\Sigma}^r_g + \bar{\Sigma}_{g\to g}$ and will be incorrect for computing the scattering source if $n \to 2n$ 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 $n \to 2n$ scattering is present. The within-collapsed-group scattering is computed similar to that for flux-weighted cross sections by neutron balance. If $n \to 2n$ 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 invarient 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)

Quadrature	One-dimensional set		Two-dimensional set		
	DR	W	DR	DZ	W
S_2	-1.0	0	-0.81649660	0.57735027	0
	-.57735027	.5	-.57735027	.57735027	.5
	.57735027	.5	.57735027	.57735027	.5
S_4	-1.0	0	-0.47140452	0.88191710	0
	-.88191710	.1666667	-.33333333	.88191710	.1666667
	-.33333333	.3333333	.33333333	.88191710	.1666667
	.33333333	.3333333	-.94280904	.33333333	0
	.88191710	.1666667	-.88191710	.33333333	.1666667
			-.33333333	.33333333	.1666667
			.33333333	.33333333	.1666667
			.88191710	.33333333	.1666667
S_6	-1.0	0	-0.36514837	0.93094934	0
	-.93094934	.0916667	-.25819889	.93094934	.09166667
	-.68313005	.150	.25819889	.93094934	.09166667
	-.25819889	.2583333	-.73029674	.68313005	0
	.25819889	.2583333	-.68313005	.68313005	.075
	.68313005	.150	-.25819889	.68313005	.075
	.93094934	.0916667	.25819889	.68313005	.075
			.68313005	.68313005	.075
			-.96609178	.25819889	0
			-.93094934	.25819889	.09166667
			-.68313005	.25819889	.075
			-.25819889	.25819889	.09166667
			.25819889	.25819889	.09166667
			.68313005	.25819889	.075
			.93094934	.25819889	.09166667

Quadrature	One-dimensional set		Two-dimensional set		
	DR	W	DR	DZ	W
S_8	-1.0	0	-0.30860672	0.95118972	0
	-.95118972	.06349070	-.21821790	.95118972	.06349070
	-.78679579	.09138352	.21821790	.95118972	.06349070
	-.57735027	.12676087	-.61721341	.78679579	0
	-.21821790	.21836491	-.57735027	.78679579	.04569176
	.21821790	.21836491	-.21821790	.78679579	.04569176
	.57735027	.12676087	.21821790	.78679579	.04569176
	.78679579	.09138352	.57735027	.78679579	.04569176
	.95118972	.06349070	-.81649658	.57735027	0
			-.78679579	.57735027	.04569176
			-.57735027	.57735027	.03537735
			-.21821790	.57735027	.04569176
			.21821790	.57735027	.04569176
			.57735027	.57735027	.03537735
			.78679579	.57735027	.04569176
			-.97590007	.21821790	0
			-.95118972	.21821790	.06349070
			-.78679579	.21821790	.04569176
			-.57735027	.21821790	.04569176
			-.21821790	.21821790	.06349070
			.21821790	.21821790	.06349070
			.57735027	.21821790	.04569176
			.78679579	.21821790	.04569176
			.95118972	.21821790	.06349070

PROJECTION INVARIANT SET B (REF. 2)

Quadrature	One-dimensional set		Two-dimensional set		
	DR	W	DR	DZ	W
S_2	-1.0	0	-0.81649660	0.57735027	0
	-.57735027	.5	-.57735027	.57735027	.5
	.57735027	.5	.57735027	.57735027	.5
S_4	-1.0	0	-0.71007688	0.70412415	0
	-.70412415	.33333333	-.70412415	.70412415	.16666667
	-.09175171	.16666667	-.09175171	.70412415	.16666667
	.09175171	.16666667	.09175171	.70412415	.16666667
	.70412415	.33333333	.70412415	.70412415	.16666667
			-.99578192	.09175171	0
			-.70412415	.09175171	.16666667
			.70412415	.09175171	.16666667

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.

Deck name	Language	Purpose
TDSN	FORTRAN IV	To call the overlays
TIME1	MAP	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.

The first overlay is as follows:

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

The second overlay is as follows:

OUTER	FORTRAN IV	Contains the outer (power) iteration loop and convergence tests.

Deck name	Language	Purpose
ORSCAL	FORTRAN IV	To compute the correction factor to be applied to the currents and angular boundary fluxes if overrelaxation has been applied
FISION	FORTRAN IV	To compute the production neutron density and the new λ and to renormalize to correct the power level after each outer iteration
FIXED	FORTRAN IV	To compute that portion of the source that remains fixed within the inner iterations
INNER	FORTRAN IV	Contains the inner iteration loop and convergence tests
SORAXS	FORTRAN IV	To look up the within-group-scattering cross section and the total collision-loss cross section
FLUX	FORTRAN IV	To compute the angular fluxes and the new scalar flux and current

The third overlay is as follows:

DUMPBC	FORTRAN IV	To give restart and flux and current binary card dumps
BCDUMP	MAP, Version 12	To punch data in absolute binary (maximum of 22 words per card)
.PCH.	MAP, Version 12	Used by BCDUMP
OUTPUT	FORTRAN IV	To give the output after the final outer iteration consisting of integrated quantities and the production rate, fluxes, and currents

The fourth overlay is as follows:

TABLES	FORTRAN IV	To set up to give the activity and averaging tables
IDACAV	FORTRAN IV	To read in output identification maps to be used by TABLES
TABLE	FORTRAN IV	To call subroutines required and to list the results for the tables requested
SUM	FORTRAN IV	To compute sums of activities and fluxes over the output regions requested

Deck name	Language	Purpose
AVERAG	FORTRAN IV	To compute flux-weighted average cross sections
COLAPS	FORTRAN IV	To compute flux-weighted collapsed-group cross sections

The fifth overlay is as follows:

EDIT	FORTRAN IV	To compute activities, flux integrals, and average activities for a specified cross section and over specified groups and regions

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

```
$IBFTC TDSN    DECK
C
C      PROGRAM TDSN IS A TWO DIMENSIONAL S(N) TRANSPORT CODE.
C
C
C
C
C
C
C
C      THE COMMON STATEMENTS
       COMMON         X
       COMMON         /CALL1/     KCHAIN,    KEND,      KEFN
       COMMON         /CALL2/     KBCRED,
      1   KREG,       KALC,       KGEO,      KBBC,      KTBC,      KLBC,
      2   KRBC,                                NSN,       NG,        NZ,
      3   NR,         NTYPS,      NMAT,      LBUCK,                KCTYP,
      4   KSISO,      KGST,       KCTR,      KCGG,      KDREAD,    KFLUXI,
      5                           ITMIMX,    ITMIGM,    NZONER,    NZONEZ,
      6   KACCEL,     KALC1,      NFN,       N2N,                  ITMPRT,
      7   KBCDUP,     KACAV,      KGAVE,     KNIP,      KEDIT,
      8                           GLAM,      EIGEN,     EPS,       XNF,
      9   RYF,        TIMAX,      CALC1,                OMEGA,     OMEGAP
       COMMON         /CALL3/
      1                           HZ,        HY,        BF,
      2                                                 KCA,       KCF,
      3   KONV,       KFINSH,     KASOR,     KUPS,
      4   NBF,        ND,         NGP1,      NZP1,      NRP1,
      5   NIJ,        NTYNG,      NDNZ,      NDNR,      NDNIJ,     NDNZNG,
      6   NXNZS,      NAR,        NC,        NIJG,      NBUCK,
      7   ITMA,       ITMI,       ITMIST,    ITMIGX,    ITD,
      8                           NEGSOR,               LZSAV,
      9   TIMEX,      TIMEZ,                            SDGFG1,    SCALUP
       COMMON         /CALL4/     EPG,       EPGX,      EPGM,      EPGMP,
      1                           OMEGAS,    OMEGPS,
      2                EXTRA1,    EXTRA2,    EXTRA3,    EXTRA4,    EXTRA5,
      3                KXTRA1,    KXTRA2,    KXTRA3,    KXTRA4,    KXTRA5
       COMMON         /COUTER/    KDR,       KDZ,       KMR,       KW,
      1   KNI,        KNO,                   K180,      KWHITE,    KNOR,
      2   KNORUP,     KNZR,       KNR,       KJRR,      KJZR,
      3               KNZ
       COMMON         /CINPT1/    LKI,       LKE,       LFG,       LKF,
      1   LKAS,       LAG,        LFNG,      LSG,       LASG,      LBNLG,
      2   LSCG,       LSDG,       LSING,     LXNG,      LJRG,      LJZG,
      3   LXNB,       LHNLG,      LVNLG,     LXNLG,     LBUC,      LMXVAR,
      4   LAR,        LAZ,        LR,        LRM,       LRA,       LRAV,
      5   LCT,        LCGG,       LCGGA,     LSISO,     LST,       LARSO,
      6   LAZSO,      LARST,      LAZST,     LCUPS,     LESIJM,    LDC,
      7   LNM,        LTVLK,      LDELZ,     LZ,        LDR,       LDZ,
      8   LMR,        LW,         LNI,       LNO,       LN2N
       COMMON         /CINPT2/    LMA,       LV,        LF,        LNN,
      1   LNA,        LJRN,       LJZN,      LJRA,      LJZA,      LJRR,
      2   LJZR,       L180,       LNOR,      LNR,       LNORUP,    LNZR,
      3   LNZ,        LNOSUP,     LNZS,      LNOS,      LN,
      4   LJR,        LJZ,                   LC,        LNBUC,
      5   LWHITE,     LBCDUP,                           LMID,      LNMI,
      6   LNMR,       LRMI,       LDZ1,      LC1,
      7   LSTG,                                                    LNORM,
      8                           LNMRA,     LNMZA,     LIDM,      LIDMAP,
      9   LIDGP
       COMMON         /CINPT3/                          LAST1,     LAST2,
```

```
     1   LAST3,    LAST4,    LAST
C
C
C      THE DIMENSION STATEMENTS
       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(LJZ),   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(LTVLK),X(LSISO),X(LST),X(LARSO),X(LAZSO),X(LARST),X(LAZST),
     6   X(LDC),   X(LBNLG),   X(LCUPS),X(LN2N),     X(LNM),    X(LNN),
     7          X(LNA),       X(LJRN),X(LJZN),X(LJRA),X(LJZA),
     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),                          X(LFG),X(LF),
     1             X(LN), X(LJR),X(LJZ),     X(LV),
     2   X(LKF),X(LKAS),      X(LAG),X(LFNG),X(LSG),X(LASG),X(LSCG),
     3   X(LSDG),X(LSING),X(LXNG),X(LJRG),X(LJZG),X(LXNB),X(LHNLG),
     4   X(LVNLG),X(LXNLG),X(LBNLG),                X(LSTG),
     5                                        X(LNORM),     X(LBUC),
     6   X(LMXVAR),X(LESIJM)    )
      GO TO 1100
C
 1405 CALL TABLES (X(LMA),X(LC),  X(LN2N),   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(LN),X(LV)     )
      GO TO 1000
C
C
C
      END
```

```
$IBFTC INPUT   DECK
      SUBROUTINE INPUT
C
C     SUBROUTINE INPUT IS THE OVERLAY (OR CHAIN) SUBROUTINE OF TDSN
C        THAT READS IN THE INPUT PARAMETERS, COMPUTES OTHER PARAMETERS,
C        AND CALLS THE READSV, SETUP, AND FISION SUBROUTINES.
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        X
      COMMON        /CALL1/     KCHAIN,     KEND,       KEFN
      COMMON        /CALL2/     KBCRED,
     1   KREG,      KALC,       KGEO,       KBBC,       KTBC,       KLBC,
     2   KRBC,                                NSN,        NG,         NZ,
     3   NR,        NTYPS,      NMAT,       LBUCK,                  KCTYP,
     4   KSISO,     KGST,       KCTR,       KCGG,       KDREAD,     KFLUXI,
     5                          ITMIMX,     ITMIGM,     NZONER,     NZONEZ,
     6   KACCEL,    KALC1,      NFN,        N2N,                    ITMPRT,
     7   KBCDUP,    KACAV,      KGAVE,      KNIP,       KEDIT,
     8                          GLAM,       EIGEN,      EPS,        XNF,
     9   RYF,       TIMAX,      CALC1,                  OMEGA,      OMEGAP
      COMMON        /CALL3/
     1                          HZ,         HY,         BF,
     2                                                  KCA,        KCF,
     3   KONV,      KFINSH,     KASOR,      KUPS,
     4   NBF,       ND,         NGP1,       NZP1,       NRP1,
     5   NIJ,       NTYNG,      NDNZ,       NDNR,       NDNIJ,      NDNZNG,
     6   NXNZS,     NAR,        NC,         NIJG,       NBUCK,
     7   ITMA,      ITMI,       ITMIST,     ITMIGX,     ITD,
     8                          NEGSOR,                 LZSAV,
     9   TIMEX,     TIMEZ,                              SDGFG1,     SCALUP
      COMMON        /CALL4/     EPG,        EPGX,       EPGM,       EPGMP,
     1                          OMEGAS,     OMEGPS,
     2                EXTRA1,   EXTRA2,     EXTRA3,     EXTRA4,     EXTRA5,
     3                KXTRA1,   KXTRA2,     KXTRA3,     KXTRA4,     KXTRA5
      COMMON        /COUTER/    KDR,        KDZ,        KMR,        KW,
     1   KNI,       KNO,                    K180,       KWHITE,     KNOR,
     2   KNORUP,    KNZR,       KNR,        KJRR,       KJZR,
     3              KNZ
      COMMON        /CINPT1/    LKI,        LKE,        LFG,        LKF,
     1   LKAS,      LAG,        LFNG,       LSG,        LASG,       LBNLG,
     2   LSCG,      LSDG,       LSING,      LXNG,       LJRG,       LJZG,
     3   LXNG,      LHNLG,      LVNLG,      LXNLG,      LBUC,       LMXVAR,
     4   LAR,       LAZ,        LR,         LRM,        LRA,        LRAV,
     5   LCT,       LCGG,       LCGGA,      LSISO,      LST,        LARSO,
     6   LAZSO,     LARST,      LAZST,      LCUPS,      LESIJM,     LDC,
     7   LNM,       LTVLK,      LDELZ,      LZ,         LDR,        LDZ,
     8   LMR,       LW,         LNI,        LNO,        LN2N
      COMMON        /CINPT2/    LMA,        LV,         LF,         LNN,
     1   LNA,       LJRN,       LJZN,       LJRA,       LJZA,       LJRR,
     2   LJZR,      L180,       LNOR,       LNR,        LNORUP,     LNZR,
```

```
     3   LNZ,       LNOSUP,   LNZS,     LNOS,     LN,
     4   LJR,       LJZ,                LC,       LNBUC,
     5   LWHITE,    LBCDUP,                       LMID,     LNMI,
     6   LNMR,      LRMI,     LDZ1,     LC1,
     7   LSTG,                                              LNORM,
     8                         LNMRA,    LNMZA,    LIDM,     LIDMAP,
     9   LIDGP
      COMMON        /CINPT3/                      LAST1,    LAST2,
     1   LAST3,     LAST4,    LAST
      COMMON        /CSETUP/  KALCP1,   AVF,      PI,       GLAMP,
     1   EXTRP
C
C      THE DIMENSION STATEMENTS
       DIMENSION    X(17500)
       DIMENSION    TITLE(14)
C
C
C      THE FORMAT STATEMENTS
  100 FORMAT (1H1)
  101 FORMAT (1HL)
  102 FORMAT (1HK)
  103 FORMAT (1H )
  104 FORMAT (1HL,21HPROGRAM STOP AT KEFN=,I8)
  105 FORMAT (11,1X,14A5)
  106 FORMAT (2X,14A5)
  107 FORMAT (1HL,6X,7HKBCRED=,I2)
  108 FORMAT (1HL,29HPROGRAM SIZE EXCEEDED.  LAST=,I8,32H  WITH 17500 ST
     1ORAGES AVAILABLE.)
  110 FORMAT (7I10)
  111 FORMAT (7I16)
  112 FORMAT (7E10.6)
  113 FORMAT (7E16.8)
  114 FORMAT (2E16.8,48X,2I16)
  115 FORMAT (5E10.6,2I10)
  120 FORMAT (1HL,10X,4HKREG,12X,4HKALC,12X,4HKGEO,12X,4HKLBC,12X,
     1    4HKRBC,12X,4HKBBC,12X,4HKTBC)
  121 FORMAT (1HL,11X,3HNSN,14X,2HNG,14X,2HNR,14X,2HNZ,12X,4HNMAT,11X,
     1    5HNTYPS,13X,3HN2N)
  122 FORMAT (1HL,9X,5HKCTYP,11X,5HKSISO,12X,4HKGST,12X,4HKCTR,12X,
     1    4HKLGG,11X,5HNBUCK,11X,5HLBUCK)
  123 FORMAT (1HL,8X,6HITMIMX,10X,6HITMIGM,10X,6HNZONER,10X,6HNZONEZ,
     1    10X,6HKDREAD,10X,6HKFLUXI,11X,5HKALC1)
  124 FORMAT (1HL,8X,6HITMPRT,10X,6HKBCDUP,11X,5HKACAV,11X,5HKGAVE,
     1    12X,4HKNIP,11X,5HKEDIT)
  125 FORMAT (1HL,10X,4HGLAM,11X,5HEIGEN,13X,3HEPS,13X,3HXNF,13X,3HRYF,
     1    11X,5HTIMAX,11X,5HCALC1)
  126 FORMAT (1HL,12X,2HHZ,14X,2HHY,14X,2HBF)
  127 FORMAT (1HL,9X,5HOMEGA,10X,6HOMEGAP,58X,6HKACCEL,12X,4HKUPS)
  131 FORMAT (1HL,10X,4HLAST,11X,5HLAST1,11X,5HLAST2,11X,5HLAST3,11X,
     1    5HLAST4)
C
C
C
C      READ INPUT PARAMETERS.
C
 1000 WRITE (6,100)
```

```
 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 (X(1),X(LBCDUP))
      IF (KBCRED-2) 1100,1016,1016
 1016 READ (5,110) ITMIMX,ITMIGM
      READ (5,115) EPS,TIMAX,OMEGA,OMEGAP,TEMP,KACCEL,KUPS
      IF (KBCRED-3) 1060,1019,1060
 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,KSISO,KGST,KCTR,KCGG,NBUCK,LBUCK,
     3              ITMIMX,ITMIGM,NZONER,NZONEZ,KDREAD,KFLUXI,KALC1




      READ (5,112) GLAM,EIGEN,EPS,XNF,RYF,TIMAX,CALC1
      READ (5,115) OMEGA,OMEGAP,TEMP,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-0.1E-07) 1035,1035,1040
 1035 BF=1.8137994
 1040 NBF=NR
      IF (NZ) 1060,1060,1045
 1045 GO TO (1050,1060,1060), KGEO
 1050 NBF=NBF*NZ
C
C
 1060 IF (ITMIMX) 1065,1065,1070
 1065 ITMIMX=4000
 1070 IF (ITMIGM) 1075,1075,1080
 1075 ITMIGM=20
 1080 IF (TIMAX-0.1E-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,NR,NZ,NMAT,NTYPS,N2N
      WRITE (6,122)
```

```
      WRITE (6,111) KCTYP,KSISO,KGST,KCTR,KCGG,NBUCK,LBUCK
      WRITE (6,123)
      WRITE (6,111) ITMIMX,ITMIGM,NZONER,NZONEZ,KDREAD,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,KBCOUP,KACAV,KGAVE,KNIP,KEDIT
      IF (NBF-1) 1125,1125,1105
 1105 WRITE (6,126)
      WRITE (6,113) HZ,HY,BF
      NBUCK=0
C
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
      OMEGPS=OMEGAP-1.0
C
 1150 IF (KBCRED) 1155,1155,1240
 1155 GO TO (1160,1165,1161), KGEO
 1160 IF (NZ-1) 1161,1165,1165
 1161 ND=NSN+1
      GO TO 1170
 1165 ND=(NSN*(NSN+4))/4
 1170 NGP1=NG+1
      NRP1=NR+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 NXNZS=NDNRNG
      LZSAV=1
      GO TO 1185
 1180 AVF=1.0
      NZ=1
 1181 NXNZS=1
 1185 NIJ=NR*NZ
      NZP1=NZ+1
      NAR=NRP1*NZ
      NTYNG=NTYPS*NG
      NDNZ=ND*NZ
      NDNIJ=ND*NIJ
      NDNZNG=NDNZ*NG
      NC=NTYNG*NMAT
      NIJG=NIJ*NG
      IF (NBUCK) 1189,1189,1186
 1186 NBUCK=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
      LFG=LKE+NGP1
      LKF=LFG+NGP1
      LKAS=LKF+NG
      IF (KSISO) 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 LFNG=LAG+NGP1
      LSG=LFNG+NG
      LSCG=LSG+NGP1
      LSDG=LSCG+NGP1
      LSING=LSDG+NGP1
      LXNG=LSING+NGP1
      LXNB=LXNG+NGP1
      LHNLG=LXNB+NGP1
      IF (NZ-1) 11186,11186,11187
11186 LVNLG=LHNLG+1
      GO TO 11188
11187 LVNLG=LHNLG+NGP1
11188 LXNLG=LVNLG+NGP1
      LBNLG=LXNLG+NGP1
      IF (NBUCK) 11189,11189,11191
11189 IF (NBF-1) 11190,11190,11191
11190 LMXVAR=LBNLG+1
      GO TO 11192
11191 LMXVAR=LBNLG+NGP1
11192 LESIJM=LMXVAR+NG
      LBUC=LESIJM+NG
C
      IF (LBUCK) 11194,11193,11194
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+ND
      IF (NZ-1) 1194,1194,1195
 1194 LNZS=LNOSUP+1
      GO TO 1196
 1195 LNZS=LNOSUP+NDNZNG
 1196 LNOS=LNZS+NXNZS
      LN=LNOS+NDNZNG
      LJR=LN+NIJG
      IF (KSISO) 1197,1197,1199
 1197 LJZ=LJR+1
 1198 LN2N=LJZ+1
      GO TO 1201
 1199 LJZ=LJR+NIJG
      IF (NZ-1) 1198,1198,1200
 1200 LN2N=LJZ+NIJG
 1201 LCUPS=LN2N+NMAT*NG
      IF ((KCTR+1)-KCGG) 11198,11197,11197
11197 KUPS=0
      LNBUC=LCUPS+1
      GO TO 11199
11198 LNBUC=LCUPS+NMAT*NG
C
11199 IF (NBUCK) 11200,11200,11201
11200 LAR=LNBUC+1
      GO TO 11202
11201 LAR=LNBUC+NBUCK
11202 LAZ=LAR+NRP1
      LR=LAZ+NR
      LRM=LR+NRP1
      LRA=LRM+1
      LRAV=LRA+NRP1
      LDELZ=LRAV+NR
      LZ=LDELZ+NZ
C
      LMA=LZ+NZP1
      LV=LMA+NIJ
      LF=LV+NIJ
      LC=LF+NIJ
      LBCDUP=LC+NC-1
C
C
C
      LCT=LC+NC
      LCGG=LCT+NR
      LSISO=LCGG+NR
      LST=LSISO+NIJ
      LDC=LST+NR
      LNM=LDC+NR
      LTVLK=LNM+NR
      LCGGA=LTVLK+NBF
      IF (KSISO) 1202,1202,1204
```

```
 1202 LARSO=LCGGA+1
      LARST=LARSO+1
      LAZSO=LARST+1
 1203 LAZST=LAZSO+1
      LJRA=LAZST+1
      GO TO 1206
 1204 LARSO=LCGGA+NR
      LARST=LARSO+NIJ
      LAZSO=LARST+NR
      IF (NZ-1) 1203,1203,1205
 1205 LAZST=LAZSO+NIJ
      LJRA=LAZST+NR
C
 1206 IF (KSISO) 1207,1207,1209
 1207 LJRN=LJRA+1
      LJZN=LJRN+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 (KBBC-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=L180+ITEMP
      KWHITE=K180+ITEMP
11213 IF (KRBC-3) 1212,1213,1212
 1212 LNORUP=LWHITE+1
      KNORUP=KWHITE+1
      GO TO 1214
 1213 LNORUP=LWHITE+NZ
      KNORUP=KWHITE+NZ
 1214 IF (KRBC) 1215,1215,1220
 1215 IF (KTBC) 1216,1216,1220
 1216 LNZR=LNORUP+1
      KNZR=KNORUP+1
      LNOR=LNZR+1
      KNOR=KNZR+1
      LNR=LNOR+1
      KNR=KNOR+1
      LJRR=LNR+1
      KJRR=KNR+1
 1217 LJZR=LJRR+1
```

```
      KJZR=KJRR+1
 1218 LNZ=LJZR+1
      KNZ=KJZR+1
      IF (NZ-1) 1219,1219,1228
 1219 LAST2=LNZ+1
      GO TO 1230
 1220 IF (NZ-1) 1221,1221,1223
 1221 LNZR=LNORUP+1
      KNZR=KNORUP+1
 1222 LNOR=LNZR+1
      KNOR=KNZR+1
      GO TO 1225
 1223 LNZR=LNORUP+NDNZ
      KNZR=KNORUP+NDNZ
      IF (LZSAV) 1222,1222,1224
 1224 LNOR=LNZR+NDNR
      KNOR=KNZR+NDNR
 1225 LNR=LNOR+NDNZ
      KNR=KNOR+NDNZ
      LJRR=LNR+NIJ
      KJRR=KNR+NIJ
      IF (KSISO) 1217,1217,1226
 1226 LJZR=LJRR+NIJ
      KJZR=KJRR+NIJ
      IF (NZ-1) 1218,1218,1227
 1227 LNZ=LJZR+NIJ
      KNZ=KJZR+NIJ
 1228 LAST2=LNZ+NDNR
C
C
 1230 NMXZON=NZONER
      IF (NZONEZ-NMXZON) 1232,1232,1231
 1231 NMXZON=NZONEZ
 1232 LMID=LCT
      LNMI=LMID+NMXZON
      LNMR=LNMI+NMXZON
      IF (KFLUXI) 1235,1235,1233
 1233 IF (NMXZON-5) 1234,1235,1235
 1234 NMXZON=5
 1235 LRMI=LNMR+NMXZON
      LDZ1=LRMI+NMXZON
      LC1=LDZ1+(NSN/2)+1
      LAST1=LC1+NG*NTYPS
C
C
 1237 LSTG=LARSO
      LNORM=LSTG+NGP1
      LAST3=LNORM+NIJ
C
C
 1240 LNMRA=LCT
      IF (KACAV) 1244,1241,1244
 1241 IF (KGAVE) 1242,1242,1244
 1242 LNMZA=LNMRA+1
      LIDM=LNMZA+1
      LIDMAP=LIDM+1
```

```
      LIDGP=LIDMAP+1
 1243 LAST4=LIDGP+1
      GO TO 1250
 1244 LNMZA=LNMRA+50
      LIDM=LNMZA+30
      LIDMAP=LIDM+50
      LIDGP=LIDMAP+NIJ
      IF (KGAVE) 1243,1243,1245
 1245 LAST4=LIDGP+NG
C
C
 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,LAST1,LAST2,LAST3,LAST4
      IF (LAST-17500) 1275,1275,1261
 1261 WRITE (6,108) LAST
      KEND=1
      KEFN=1261
      GO TO 1450
C
C
C
 1275 CALL TIME1(TIMEX)
      KALCP1=KALC+1
C
C     READ INPUT SUBSCRIPTED VARIABLES.
C
 1300 CALL READSV (
     1   X(LKI),X(LMA),X(LC),X(LDR),X(LDZ),X(LW),X(LNOS),X(LNOSUP),
     2   X(LNZS),X(LN),X(LJR),X(LJZ),X(LBUC),X(LNBUC), X(LCUPS),X(LN2N),
     3   X(LMID),X(LNMI),X(LNMR),X(LRMI),X(LR),X(LZ),       X(LC1)    )
      IF (KEND) 1350,1350,1305
 1305 WRITE (6,104) KEFN
      KEFN=1305
      GO TO 1450
C
C     SETUP QUANTITIES NEEDED FOR EVERY ITERATION.
C
 1350 CALL SETUP (   X(LKI),   X(LKE),   X(LMA),
     1   X(LV),X(LAR),X(LAZ),X(LDELZ),X(LDR),X(LDZ),X(LMR),X(LW),
     2   X(LR),X(LZ),X(LRM),X(LRA),X(LRAV),     X(LDZ1)    )
      IF (KEND) 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 1400
 1375 HZ=2.0*X(LZP)
C
C
C
 1400 KCHAIN=2
      RETURN
C
C     IF KEND=1
C
 1450 WRITE (6,104) KEFN
      KCHAIN=1
      RETURN
C
      END
```

```
$IBFTC READSV  DECK
      SUBROUTINE READSV (  XKI, MA, C, DR, DZ, W,  XNOS, XNOSUP, XNZS,
     1   XN, XJR, XJZ,    BUCLK, BUCKG,  CUPS, CN2N,
     2   MID,NMI,NMR,RMI,R,Z,C1)
C
C      SUBROUTINE READSV OF SUBROUTINE INPUT OF TDSN IS TO READ THE
C         INPUT SUBSCRIPTED VARIABLES.
C
C
C
C
C
C
C
C      THE COMMON STATEMENTS
       COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
       COMMON        /CALL2/   KBCRED,
      1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
      2   KRBC,                          NSN,      NG,       NZ,
      3   NR,        NTYPS,    NMAT,     LBUCK,               KCTYP,
      4   KSISO,     KGST,     KCTR,     KCGG,     KOREAD,   KFLUXI,
      5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
      6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
      7   KBCDUP,    KALCAV,   KGAVE,    KNIP,     KEDIT,
      8                        GLAM,     EIGEN,    EPS,      XNF,
      9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
       COMMON        /CALL3/
      1                        HZ,       HY,       BF,
      2                                            KCA,      KCF,
      3   KONV,      KFINSH,   KASOR,    KUPS,
      4   NBF,       ND,       NGP1,     NZP1,     NRP1,
      5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
      6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
      7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
      8                        NEGSOR,             LZSAV,
      9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
       COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
      1                        OMEGAS,   OMEGPS,
      2                EXTRA1,   EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
      3                KXTRA1,   KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
C
C      THE DIMENSION STATEMENTS
       DIMENSION     XKI(1),          MA(1),          C(1),
      1              DR(1),           DZ(1),          W(1)
       DIMENSION     XNOS(1),         XNOSUP(1),      XNZS(1),
      1              XN(1),           XJR(1),         XJZ(1),
      2              BUCLK(1),        BUCKG(1),       CUPS(1),
      3              CN2N(1)
       DIMENSION     MID(1),          NMI(1),         NMR(1),
      1              RMI(1),          R(1),           Z(1)
       DIMENSION     C1(1)
C
C      THE FORMAT STATEMENTS
  105 FORMAT (1H )
  110 FORMAT (7I10)
  112 FORMAT (7E10.6)
```

```
  113 FORMAT (7E16.8)
  114 FORMAT (I5,E10.6,I5)
  115 FORMAT (I16,E16.8,I16)
  116 FORMAT (40I3)
  118 FORMAT (I5,E10.6,I5,E10.6,I5,E10.6,I5,E10.6,I5,E10.6)
  119 FORMAT (1HL,35HTOO MANY VALUES FOR FLUX XN READ IN    )
  120 FORMAT (1HL,2X,45H      NMI(I)          RMI(I)          MID(I)   )
  121 FORMAT (1HL,56HTHE FIRST DIRECTION MESH CELL BOUNDARY POSITIONS (R
     1(NR)))
  122 FORMAT (1HL,57HTHE SECOND DIRECTION MESH CELL BOUNDARY POSITIONS (
     1Z(NZ)))
  123 FORMAT (1HL,28HTHE MATERIAL MAP (MA(NZ,NR)))
  124 FORMAT (1HL,30HTHE FISSION SPECTRUM (XKI(NG)))
  125 FORMAT (1HL,37HTHE CROSS SECTIONS (C(NMAT,NG,NTYPS)))
  130 FORMAT (6F12.7)
  131 FORMAT (1HL,70HTHE MATERIAL INDEPENDENT EFFECTIVE BUCKLING CROSS S
     1ECTIONS (BUCLK(NG)))
  132 FORMAT (1HL,73HTHE MATERIAL DEPENDENT EFFECTIVE BUCKLING CROSS SEC
     1TIONS (BUCKG(NMAT,NG)))
  134 FORMAT (1HL,57HTHE OUT SCATTERING REMOVAL CROSS SECTIONS (CN2N(NMA
     1T,NG)))
  135 FORMAT (6X,2HN=,I5)
C
C      THE INTEGER FUNCTIONS
       INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 IF (KBCRED) 1001,1001,1040
 1001 WRITE (6,120)
       DO 1005 I=1,NZONER
       READ (5,114) NMR(I),RMI(I),MID(I)
 1005 WRITE (6,115) NMR(I),RMI(I),MID(I)
C
 1010 I=1
       I1=1
       R(I1)=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
       IF (I2-NMR(I)) 1030,1030,1035
 1030 I1M1=I1
       I1=I1+1
       R(I1)=R(I1M1)+DELR
       MA(I1M1)=MID(I)
       GO TO 1025
 1035 I=I+1
       IF (I-NZONER) 1015,1015,1040
 1040 WRITE (6,121)
       WRITE (6,113) (R(I), I=1,NRP1)
C
 1050 IF (NZ-1) 1180,1180,1055
```

```
 1055 IF (KBCRED) 1060,1060,1140
 1060 IF (NZONEZ-1) 1065,1065,1100
 1065 I1=NR
      DO 1070 J=2,NZ
      DO 1070 I=1,NR
      I1=I1+1
 1070 MA(I1)=MA(I)
C
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
      I1=1
      Z(I1)=0.0
      DELR=RMI(J)
      GO TO 1120
 1115 DELR=RMI(J)-RMI(J-1)
 1120 VMI=NMI(J)
      DELR=DELR/VMI
      I2=0
 1125 I2=I2+1
      IF (I2-NMI(J)) 1130,1130,1135
 1130 I1M1=I1
      I1=I1+1
      Z(I1)=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 (5,110) (MID(I), I=1,NZONER)
      K=K+NMI(JJ)
 1165 J=J+1
      I=0
      DO 1170 II=1,NZONER
      L=NMR(II)
      DO 1170 KK=1,L
      I=I+1
      IJ=INDEX(NR,J,I)
 1170 MA(IJ)=MID(II)
      IF (J-K) 1165,1175,1175
 1175 CONTINUE
C
C
 1180 WRITE (6,123)
      DO 1185 J=1,NZ
      IJ=INDEX(NR,J,1)
```

```
      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
      DO 1310 IG=1,NG
      DO 1310 K=1,NTYPS
      L=L+1
 1310 C1(L)=0.0
C
      DO 1325 IG=1,NG
      K1=INDEX(NTYPS,IG,1)
      K2=INDEX(NTYPS,IG,4)
      K3=INDEX(NTYPS,IG,NTYPS)
      READ (5,112) (C1(K), K=K1,K2),TEMP,C1(K3)

      WRITE (6,113) (C1(K), K=K1,K2),TEMP,C1(K3)

      IF (NBUCK) 1316,1316,1315
 1315 K1=INDEX(NG,N,IG)
      BUCKG(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
```

```
 1325 CONTINUE
C
      IF (KCGG+1-NTYPS) 1330,1340,1340
 1330 WRITE (6,105)
      IGEND=NG
      NGM1=NG-1
      DO 1335 IG=1,NGM1
      IGEND=IGEND-1
      IF (IGEND-(NTYPS-KCGG)) 1331,1331,1332
 1331 K4=KCGG+IGEND-1
      GO TO 1333
 1332 K4=NTYPS1
 1333 K1=INDEX(NTYPS,IG,KCGG)
      K2=INDEX(NTYPS,IG,K4)
      READ (5,112) (C1(K), K=K1,K2)
 1335 WRITE (6,113) (C1(K), K=K1,K2)
C

 1340 IF (KUP) 1345,1345,1341
 1341 WRITE (6,105)
      IG1=NG+1-KUP
      K3=KCTR+1
      DO 1342 IG=IG1,NG
      K4=KCGG+IG-(NG+1)
      K1=INDEX(NTYPS,IG,K3)
      K2=INDEX(NTYPS,IG,K4)
      READ (5,112) (C1(K), K=K1,K2)
      WRITE (6,113) (C1(K), K=K1,K2)
 1342 CONTINUE
C
 1345 CALL XSTDSN ( N, KUP,   MA, C1, CN2N,  C  )
 1350 WRITE (6,105)
      GO TO 1375
C
C
 1355 DO 1365 N=1,NMAT
      K1=INDEX(NTYNG,N,1)
      K2=INDEX(NTYNG,N,NTYNG)
      READ (5,130) (C(K), K=K1,K2)
      IF (KSISO) 1360,1360,1356
 1356 DO 1357 IJ=1,NIJ
      IF ((MA(IJ)-1)+N) 1357,1358,1357
 1357 CONTINUE
      GO TO 1360
 1358 DO 1359 IG=1,NG
      K3=INDEX(NTYPS,IG,KCTR)
      K3=K3+1
      K3=INDEX(NTYNG,N,K3)
      K4=INDEX(NTYPS,IG,NTYPS)
      K4=INDEX(NTYNG,N,K4)
      DO 1359 K=K3,K4
 1359 C(K)=3.0*C(K)
 1360 IF (N2N) 1365,1365,1361
 1361 K1=INDEX(NG,N,1)
      K2=INDEX(NG,N,NG)
      READ (5,112) (CN2N(K), K=K1,K2)
```

```
 1365 CONTINUE
C
C
 1375 WRITE (6,125)
      DO 1390 N=1,NMAT
      WRITE (6,135) N
      DO 1380 IG=1,NG
      K3=INDEX(NTYPS,IG,1)
      K4=INDEX(NTYPS,IG,NTYPS)
      K1=INDEX(NTYNG,N,K3)
      K2=INDEX(NTYNG,N,K4)
      IF (KBCRED) 1376,1376,1380
 1376 IF (KCTYP) 1377,1377,1380
 1377 READ (5,112) (C(K), K=K1,K2)
 1380 WRITE (6,113) (C(K), K=K1,K2)

      IF (KBCRED) 1385,1385,1390
 1385 IF (N2N) 1390,1390,1386
 1386 K1=INDEX(NG,N,1)
      K2=INDEX(NG,N,NG)
      IF (KCTYP) 1390,1387,1390
 1387 READ (5,112) (CN2N(K), K=K1,K2)
 1390 WRITE (6,105)
C
C     CALCULATING THE OUT SCATTERING (CN2N)
C
      IF (KBCRED) 1400,1400,1440
 1400 IF (N2N) 1401,1401,1425
 1401 ITEMP1=KCTR+1
      DO 1415 IG=1,NG
      DO 1415 M=1,NMAT
      K=INDEX(NG,M,IG)
      CN2N(K)=0.0
      DO 1415 N=ITEMP1,NTYPS
      IF (N-KCGG) 1405,1415,1405
 1405 IGG=IG-KCGG+N
      IF (IGG) 1415,1415,1407
 1407 IF (IGG-NG) 1408,1408,1415
 1408 K1=INDEX(NTYPS,IGG,N)
      K2=INDEX(NTYNG,M,K1)
      CN2N(K)=CN2N(K)+C(K2)
 1415 CONTINUE
C
C     CALCULATING SIG UP, OUT SCATTERING (CUPS)
C
 1425 IF ((KCTR+1)-KCGG) 1426,1450,1450
 1426 ITEMP1=KCTR+1
      ITEMP2=KCGG-1
      DO 1435 IG=1,NG
      DO 1435 M=1,NMAT
      K=INDEX(NG,M,IG)
      CUPS(K)=0.0
      DO 1435 N=ITEMP1,ITEMP2
      IF (KREG) 1427,1427,1430
 1427 IGG=IG-KCGG+N
      IF (IGG) 1435,1435,1428
```

```
 1428 IF (IGG-NG) 1425,1429,1435
 1429 K1=INDEX(NTYPS,IGG,N)
      GO TO 1431
 1430 K1=INDEX(NTYPS,IG,N)
 1431 K2=INDEX(NTYNG,M,K1)
      CUPS(K)=CUPS(K)+C(K2)
 1435 CONTINUE
C
C
 1440 IF (KUPS) 1450,1450,1441
 1441 DO 1445 IG=1,NG
      DO 1445 M=1,NMAT
      K=INDEX(NG,M,IG)
      IF (ABS(CUPS(K))-0.1E-7) 1443,1443,1442
 1442 KUPS=IG
      GO TO 1450
 1443 CONTINUE
 1445 CONTINUE
C
C
 1450 IF (LBUCK) 1455,1455,1451
 1451 IF (KBCRED) 1452,1452,1453
 1452 READ (5,112) (BUCLK(IG), IG=1,NG)
 1453 WRITE (6,131)
      WRITE (6,113) (BUCLK(IG), IG=1,NG)
C
C
 1455 IF (NBUCK) 1465,1465,1456
 1456 IF (KBCRED) 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 (6,113) (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
      IJGST=IJGS+1
      IJGSP=IJGS+NIJ
      IF (KBCRED) 1505,1505,1525
 1505 IF (KFLUXI) 1525,1506,1530
 1506 DO 1507 IJG=IJGST,IJGSP
 1507 XN(IJG)=1.0
      IF (KSISO) 1560,1560,1510
C
 1510 IF (KBCRED) 1511,1511,1513
 1511 DO 1512 IJG=IJGST,IJGSP
 1512 XJR(IJG)=1.0
      GO TO 1515
 1513 CALL BCREAD (XJR(IJGST),XJR(IJGSP))
 1515 IF (NZ-1) 1560,1560,1516
 1516 IF (KBCRED) 1517,1517,1519
 1517 DO 1518 IJG=IJGST,IJGSP
 1518 XJZ(IJG)=1.0
      GO TO 1560
 1519 CALL BCREAD (XJZ(IJGST),XJZ(IJGSP))
      GO TO 1560
C
 1525 CALL BCREAD (XN(IJGST),XN(IJGSP))
      IF (KSISO) 1560,1560,1510
C
 1530 L=0
 1535 READ (5,118) ((NMR(I),RMI(I)), I=1,5)
      I=1
 1540 IF (NMR(I)) 1550,1550,1541
 1541 L1=L+1
      L=L+NMR(I)
      IF (L-NIJ) 1543,1543,1542
 1542 WRITE (6,119)
      KEND=1
      GO TO 1600
 1543 ITEMP1=L1+IJGS
      ITEMP2=L+IJGS
      DO 1544 IJG=ITEMP1,ITEMP2
 1544 XN(IJG)=RMI(I)
      IF (I-5) 1545,1535,1535
 1545 I=I+1
      GO TO 1540
 1550 IF (L-NIJ) 1551,1555,1555
 1551 ITEMP1=ITEMP2+1
      DO 1552 IJG=ITEMP1,IJGSP
 1552 XN(IJG)=0.0
 1555 IF (KSISO) 1560,1560,1510
C
 1560 IF (KBCRED) 1561,1561,1590
 1561 IF (KRBC-1) 1562,1566,1566
 1562 DO 1563 I=1,NDNZNG
 1563 XNOS(I)=0.0
      IF (NZ-1) 1575,1575,1564
 1564 DO 1565 I=1,NDNZNG
 1565 XNOSUP(I)=0.0
```

```
      GO TO 1575
 1566 DO 1570 J=1,NZ
      IJG=INDEX(NZ,IG,J)
      IJG=IJG*NR
      DO 1570 M=1,ND
      LOS=LOS+1
      XNOS(LOS)=XN(IJG)
      IF (NZ-1) 1570,1570,1567
 1567 XNOSUP(LOS)=XN(IJG)
 1570 CONTINUE
C
 1575 IF (KTBC-1) 1576,1578,1578
 1576 DO 1577 I=1,NXNZS
 1577 XNZS(I)=0.0
      GO TO 1590
 1578 IF (NZ-1) 1590,1590,1579
 1579 IJG=(NZ-1)*NR
      IJG=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
C
 1590 IJGS=IJGSP
C
C
C
 1600 RETURN
C
      END
```

```
$IBFTC XSTDSN  DECK
      SUBROUTINE XSTDSN ( N, KUP,   MA, C1, CN2N,  C  )
C
C     SUBROUTINE TDSNXS IS TO OBTAIN TDSN CROSS SECTIONS FROM RP-2
C        FORMAT CROSS SECTIONS.
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCDUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
     1                        OMEGAS,   OMEGPS,
     2              EXTRA1,   EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
     3              KXTRA1,   KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
C
C     THE DIMENSION STATEMENTS
      DIMENSION     MA(1)
      DIMENSION     C1(1),          CN2N(1),        C(1)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 KCTRP1=KCTR+1
      KUP1=KUP+1
      NTYPS1=NTYPS-1
      KSI=0
      IF (KSISO) 1010,1010,1001
 1001 DO 1005 IJ=1,NIJ
      IF (MA(IJ)+N) 1002,1003,1002
 1002 IF ((MA(IJ)-1)+N) 1005,1004,1005
```

```
 1003 KSI=-1
      GO TO 1010
 1004 KSI=1
      GO TO 1010
 1005 CONTINUE
C
C
 1010 DO 1100 IG=1,NG
      L=INDEX(NTYPS,IG,1)
      K1=INDEX(NTYNG,N,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-1
      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=L+NTYPS1
      C(K)=C1(K3)
      N1=N+1
      K3=INDEX(NTYNG,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+KCGG
      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+KCGG
      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
 1043 DO 1045 K3=KCTRP1,NTYPS1
      K4=L+K3-1
 1045 C(K2)=C(K2)-C1(K4)
C
 1050 IF (IG-NG) 1055,1075,1075
 1055 IF (IG+KUP-NG) 1075,1060,1060
 1060 K=K2
      K3=KCTR
      IG1=IG
      L=0
      KUP1=KUP1-1
      L1=0
 1065 L1=L1+1
      IF (L1-KUP1) 1070,1070,1075
 1070 K=K-1
      K3=K3+1
      IG1=IG1+1
      L=L+1
      K4=INDEX(NTYPS,IG1,K3)
      C(K)=C1(K4)
      IF (L-KUP) 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)=C1(K4)
      IF (IG-L) 1100,1100,1090
 1090 IF ((K2-K1)-NTYPS) 1085,1100,1100
C
C
 1100 CONTINUE
C
C
C
      RETURN
C
      END
```

```
$IBFTC SETUP   DECK
      SUBROUTINE SETUP (XKI,  XKE,  MA,  V,  AR,  AZ,  DELZ,  DR,  DZ,
     1   MR,  W,  R,  Z,  RM,  RA,  RAV,  DZI  )
C
C     SUBROUTINE SETUP COMPUTES VARIOUS QUANTITIES NEEDED FOR EVERY
C        ITERATION.
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      NZN,                ITMPRT,
     7   KBCOUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
     1                        OMEGAS,   OMEGPS,
     2                EXTRA1,   EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
     3                KXTRA1,   KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
      COMMON        /CSETUP/  KALCP1,   AVF,      PI,       GLAMP,
     1   EXTRP
C
C     THE DIMENSION STATEMENTS
      DIMENSION     XKI(1),           XKE(1),           MA(1)
      DIMENSION     V(1),             AR(1),            AZ(1),
     1              DELZ(1),          DR(1),            DZ(1),
     2              MR(1),            W(1)
C
      DIMENSION     R(1),             Z(1),             RM(1),
     1              RA(1),            RAV(1),           DZI(1)
C
C     THE FORMAT STATEMENTS
  111 FORMAT (7I16)
  113 FORMAT (7E16.8)
  122 FORMAT (1HL,2X,22HDIRECTIONAL FUNCTIONS    )
  130 FORMAT (1HJ,2HJ=,I3)
```

```
  141 FORMAT (1HL,32HTHE MESH CELL VOLUMES (V(NZ,NR)))
  142 FORMAT (1HL,I6,37H NEGATIVE VOLUMES HAVE BEEN COMPUTED.)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C     COMPUTE INITIAL VALUES.
C
 1000 PI=3.14159265
      GO TO (1005,1010,1015), KGEO
 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 SDGFG1=0.0
C
C
      IF (ITMIST) 1025,1025,1200
 1025 GLAMP=0.0
      EXTRP=0.0
C
C
C     COMPUTE SN CONSTANTS.
C
C
 1050 IF (KBCRED) 1051,1051,1200
 1051 IF (KDREAD) 1055,1055,1052
 1052 DO 1053 M=1,ND
 1053 W(M)=W(M)/AVF
      GO TO 1150
 1055 IF (ND-(NSN+1)) 1125,1125,1060
C
C     CODED FOR CYLINDERS AND TWO DIMENSIONAL SLABS.
C
 1060 NSND2=NSN/2
      AREA=0.0
      DO 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
      MT4=4*M
      M1=M1+1
      DZ2=DZF*(DZ1(M+1)+DZ1(M))
      D1=SQRT(1.0-(DZ2*DZ2))
      DZ(M1)=DZ2
      DR(M1)=-D1
      W(M1)=0.0
 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*AREA))/AVF
      GO TO 1110
C
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)=0.0
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*(DZ1(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(M1)
      W(M)=W(M1)
```

```
      GO TO 1144
C
 1150 M1=ND
 1155 WD=AVF*W(M1)*DR(M1)
      IF (ABS(WD)-0.1E-07) 1160,1160,1156
 1156 IF (WD) 1170,1160,1165
 1160 MR(M1)=MR(M1+1)
 1165 M1=M1-1
      IF (M1) 1200,1200,1155
 1170 M=M1
 1175 M=M+1
      IF (1.0E-04-ABS(DR(M)+DR(M1))) 1175,1180,1180
 1180 MR(M)=M1
      MR(M1)=M
      GO TO 1165
C
C     COMPUTE INITAL FUNCTIONS.
C
 1200 IF (ITMA) 1305,1305,1365
 1305 IF (KALC1-1) 1325,1310,1325
 1310 DO 1315 IG=1,NG
 1315 XKI(IG)=XKI(IG)/CALC1
 1325 XKI(NGP1)=0.0
      DO 1330 IG=1,NG
      XKI(NGP1)=XKI(NGP1)+XKI(IG)
 1330 XKE(IG)=XKI(IG)
      XKE(NGP1)=XKI(NGP1)
C
 1350 DO 1360 I=1,NRP1
 1360 RA(I)=R(I)
C
C     MODIFY RADII.
C
 1365 GO TO (1400,1400,1400,1400,1370,1385), KALCP1
 1370 DO 1375 I=1,NR
      K=MA(I)
      RA(I+1)=RA(I)+(R(I+1)-R(I))*(1.0+EIGEN*RM(K))
      IF (1.0+EIGEN*RM(K)) 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     COMPUTE AREAS AND VOLUMES.
C
 1400 IF (ITMA) 1410,1410,1405
 1405 GO TO (1500,1500,1500,1500,1410,1410), KALCP1
 1410 GO TO (1411,1415,1450), KGEO
 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), KGEO
```

```
 1416 AR(I)=AF
      AZ(I)=RA(IP1)-RA(I)
      GO TO 1420
 1417 AR(I)=AF*RA(I)
      AZ(I)=AF*RAV(I)*(RA(IP1)-RA(I))
 1420 CONTINUE
      GO TO (1421,1422,1450), KGEO
 1421 AR(NRP1)=AF
      GO TO 1423
 1422 AR(NRP1)=AF*RA(NRP1)
 1423 J=1
      DELZ(J)=1.0
      IF (NZ-1) 1430,1430,1425
 1425 JP1=J+1
      DELZ(J)=Z(JP1)-Z(J)
 1430 ITEMP=0
      DO 1440 I=1,NR
      IJ=INDEX(NR,J,I)
      V(IJ)=AZ(I)*DELZ(J)
      IF (V(IJ)) 1435,1440,1440
 1435 ITEMP=ITEMP+1
 1440 CONTINUE
      IF (NZ-J) 1475,1475,1445
 1445 J=JP1
      GO TO 1425
C
 1450 DO 1465 I=1,NRP1
      GO TO (1455,1465,1460), KGEO
 1455 AR(I)=AF
      GO TO 1465
 1460 AR(I)=AF*(RA(I)**2)
 1465 CONTINUE
      ITEMP=0
      DO 1470 I=1,NR
      RAV(I)=0.5*(RA(I+1)+RA(I))
      V(I)=VF*(AR(I+1)*RA(I+1)-AR(I)*RA(I))
      IF (V(I))  1469,1470,1470
 1469 ITEMP=ITEMP+1
 1470 CONTINUE
      DELZ(1)=1.0
C
 1475 IF (ITEMP) 1500,1500,1480
 1480 WRITE (6,142) ITEMP
      KEFN=1480
      KEND=1
C
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,J,1)
      K2=INDEX(NR,J,NR)
 1510 WRITE (6,113) (V(IJ), IJ=K1,K2)
```

```
C
      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) (W(M), M=1,ND)
      WRITE (6,111) (MR(M), M=1,ND)
C
C
C
 1550 IF (KBCRED) 1575,1575,1555
 1555 IF (NFN-1) 1600,1560,1600
 1560 DO 1565 IG=1,NGP1
 1565 XKE(IG)=XKI(IG)
      GO TO 1600
C
C
 1575 NFN=1
C
C
C
 1600 RETURN
C
      END

$IBMAP BCREAD  60,M94,DECK

            7094 RELMOD ASSEMBLY.

                  $IBLDR BCREAD

                  $TEXT  BCREAD
```

BCREAD SUBROUTINE FOR IBSYS

```
                                                    ENTRY   BCREAD

BINARY CARD ID. BCREAD00
00000  1 00000 0 00005  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  0020 00 4 00001  10000
00005  0634 00 4 00006  10011
00006  0634 00 4 00110  10001
00007  0634 00 4 00003  10001
00010  0634 00 1 00002  10001
00011  0634 00 2 00001  10001
00012  0500 00 4 00003  10000          CLA     3,4          PICK UP THE FIRST ARGUMENT
00013  0560 00 4 00004  10000          LDQ     4,4          PICK UP THE SECOND ARGUMENT
00014  0040 00 0 01002  10011          TLQ     *+2          MAKE SURE THE LARGEST
00015  0131 00 0 00000  10000          XCA                  ARGUMENT IS IN THE AC
00016  4600 00 0 00107  10001          STQ     TEMP
00017  0402 00 0 00107  10001          SUB     TEMP
00020  0734 00 1 00000  10000          PAX     0,1          PUT WORD COUNT + 1
00021  1 00001 1 01001  10011          TXI     *+1,1,1      INTO INDEX 1
00022  0534 00 2 00107  10001          LXA     TEMP,2       PICK UP THE FIRST LOAD ADDRESS

BINARY CARD ID. BCREAD01
00023  0634 00 1 00037  10001          SXA     IX1,1
00024  0634 00 2 00040  10001          SXA     IX2,2
00025  0500 60 0 00105  10001          CLA*    IN5
00026  0621 00 0 01002  10011          STA     *+2
00027  0074 00 4 07000  10011          TSX     .CLOSE,4
00030  5 00000 0 00000  10000          MON     **
00031  0500 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  0074 00 4 10000  10011          TSX     .OPEN,4
00036  5 00000 0 00000  10000  MON     MON     **
00037  0774 00 1 00000  10000  IX1     AXT     **,1         HOLDS THE WORD COUNT
00040  0774 00 2 00000  10000  IX2     AXT     **,2         HOLDS THE LOADING ADDRESS
00041  0634 00 2 00047  10001  SXA     SXA     IO,2
00042  7 00026 1 00053  10001          TXL     LASTC,1,22
00043  0074 00 4 11000  10011  READ    TSX     .READ,4
00044  0 00067 0 00000  10100  READ2   PZE     **,,EOB
00045  0 00074 0 00100  10101          PZE     EOF,,ERR

BINARY CARD ID. BCREAD02
00046  4 00002 2 00000  10000          IOCPN   **,,2        SKIP FIRST WORD AND CHECKSUM
00047  0 00026 0 00000  10000  IO      IOCD    **,,22
00050  1 77752 1 01001  10011          TXI     *+1,1,-22
00051  1 00026 2 01001  10011          TXI     *+1,2,22
00052  0020 00 0 00041  10001          TRA     SXA
00053  0500 00 0 00057  10001  LASTC   CLA     DONE
00054  0601 00 0 41002  10011          STO     *-2
00055  4634 00 1 00047  10001          SXD     IO,1         REDUCE THE WORD COUNT
00056  0020 00 0 00043  10001          TRA     READ
00057  0020 00 0 01001  10011  DONE    TRA     *+1
```

```
        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 00 0 00100  10001          TRA     EOF
        00074  000000000000     00010  ERR      CALL    .FXEM.(ERR2)
        00074  0074 00 4 04000  10011
        00075  1 00001 0 01003  10011
        00076  0 00110 0 00076  10100
        00077  0 00000 0 00103  10001
        00100  000000000000     00010  EOF      CALL    EXIT
        00100  0074 00 4 05000  10011
        00101  1 00000 0 01002  10011
        00102  0 00110 0 00077  10100
        00103  0 00000 0 00043  10000  ERR2     PZE     35
        00104  0 00000 0 00044  10000  EOB2     PZE     36
        00105  0 00000 0 12000  10011  IN5      PZE     .UNO5.
        00106  0 00000 0 13000  10011  READ5    PZE     .READ5
        00107  0 00000 0 00000  10000  TEMP     PZE
        00110  000000000000     10000          *LDIR

BINARY CARD ID. BCREAD04
        00111  222351252124     10000
                          00000  01111          END

            CONTROL DICTIONARY

                    $CDICT BCREAD

BINARY CARD ID. BCREAD06
                000112000000                   PREFACE     START=0,LENGTH=74,TYPE=7094,CMPLX=5
                000004000005
                222351252124        BCREAD DECK        LOC=0,LENGTH=74
                000112000000
                222351252124        BCREAD REAL        LOC=0,LENGTH=0
                000000000000
                222351252124        BCREAD REAL        LOC=0,LENGTH=0
                000000000000
                332667254433        .FXEM. VIRTUAL     SECT. 4,CALL
                200000100000
                256731636060        EXIT   VIRTUAL     SECT. 5,CALL
                200000100000
                627062434623        SYSLOC VIRTUAL     SECT. 6
                200000000000
                332343466225        .CLOSE VIRTUAL     SECT. 7
                200000000000
                334647254560        .OPEN  VIRTUAL     SECT. 8
                200000000000
                335125212460        .READ  VIRTUAL     SECT. 9
                200000000000
                336445000533        .UNO5. VIRTUAL     SECT. 10
                200000000000

BINARY CARD ID. BCREAD07
                335125212405        .READ5 VIRTUAL     SECT. 11
                200000000000

                    $DKEND BCREAD
```

```
                SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS   SYMBOL    VALUE    REFERENCES

          BCREAD    00000    66
          DONE      00057    53
          EOB2      00104    72
          EOB       00067    44
          EOF       00100    45,73
          ERR2      00103    77
          ERR       00074    45
          IN5       00105    25
          IO        00047    41,55,63
          IX1       00037    23
          IX2       00040    24
          ..0001    00003    7,10,11
          ..0002    00004
          ..0003    00005    0
          LASTC     00053    42,61
          MON       00036    32
          READ2     00044    33
          READ5     00106    31
          READ      00043    56
  LCTR     BLCTR
  QUAL     UNDS
  LCTR    //
          SHUT      00065    34
          SXA       00041    52,60
          TEMP      00107    16,17,22

REFERENCES TO VIRTUAL SYMBOLS.

          EXIT         5     100
          .CLOSE       7     27,64
          .FXEM.       4     67,74
          .OPEN        8     35
          .READ5      11     106
          .READ        9     43
          .UNO5.      10     105
          SYSLOC       6     5

$IBMAP .READ5  5,DECK

            7094 RELMOD ASSEMBLY.

                  $IBLDR .READ5                                          02/25/66     .READ000

                  $FILE  .READ5  'READ5 ',IN1,READY,INPUT,BLK=28,MULTIREEL,MXBLK,..OLIST    .READ001
```

```
          FILE DICTIONARY.

                  $FDICT .READ5                                                    .REA0002

BINARY CARD ID. .REA0003
                604002000034  READ5  FILE    *READ5                    MXBIN,INPUT,NOHCVN,BLK=28
                000000000000
                512521240560
                606060606060
                606060606060
                     $TEXT  .READ5                                               .REA0004

                                              ENTRY   .READ5

BINARY CARD ID. .REA0005
       00000  0 00000 0 04001  10010  .READ5 PZE     READ5
                                     READ5  FILE    ,IN1,READY,INPUT,BLK=28,MULTIREEL,MXBIN,NOLIST
                           00000  01111          END

          CONTROL DICTIONARY

                  $CDICT .READ5                                                    .REA0006

BINARY CARD ID. .REA0007
                000001000000              PREFACE     START=0,LENGTH=1,TYPE=7094,CMPLX=5
                000004000005
                335125212405       .READ5 DECK        LOC=0,LENGTH=1
                000001000000
                335125212405       .READ5 REAL        LOC=0,LENGTH=0
                000000000000

                     $DKEND .READ5                                  001184       .REA0008

          SYMBOL REFERENCE DATA


REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         .READ5   00000
  FILE   READ5        1   0
  LCTR    BLCTR
  QUAL    UNOS
  LCTR   //
```

```
$IBFTC OUTER   DECK
      SUBROUTINE OUTER (XKI,XKE,      MA,C,        X1,
     1   FG,F,    XNOS,XNOSUP,XNZS,     XN,  XJR,XJZ,          V,AR,AZ,DELZ,
     2   KFING,KASORG,       AG,FNG,SG,ASG,SCG,SDG,SING,XNG,XNB,HNLG,
     3   VNLG,XNLG,       CT,CGG,CGGA,TVLK,SISO,ST,ARSISO,AZSISO,ARST,
     4   AZST,CC,  BNLG,      CUPS,CN2N,              XNM,    XNN,XNA,
     5   XJRN,XJZN,XJRA,XJZA,                 X2,          BUCLK,BUCKG,
     6   MXVARI,ESIJM    )
C
C      SUBROUTINE OUTER IS THE OVERLAY (OR CHAIN) SUBROUTINE OF TDSN
C         THAT PREFORMS THE OUTER (OR MAJOR) ITERATIONS.
C
C
C
C
C
C
C
C      THE COMMON STATEMENTS
       COMMON        /CALL1/   KCHAIN,   KEND,      KEFN
       COMMON        /CALL2/   KBCRED,
     1   KREG,       KALC,     KGEO,     KBBC,      KTBC,      KLBC,
     2   KRBC,                                      NSN,       NG,        NZ,
     3   NR,         NTYPS,    NMAT,     LBUCK,                KCTYP,
     4   KSISO,      KGSF,     KCTR,     KCGG,      KDREAD,    KFLUXI,
     5                         ITMIMX,   ITMIGM,    NZONER,    NZONEZ,
     6   KACCEL,     KALC1,    NFN,      N2N,                  ITMPRT,
     7   KBCDUP,     KACAV,    KGAVE,    KNIP,      KEDIT,
     8                         GLAM,     EIGEN,     EPS,       XNF,
     9   RYF,        TIMAX,    CALC1,                OMEGA,     OMEGAP
       COMMON        /CALL3/
     1                         HZ,       HY,        BF,
     2                                               KCA,       KCF,
     3   KONV,       KFINSH,   KASOR,    KUPS,
     4   NBF,        ND,       NGP1,     NZP1,      NRP1,
     5   NIJ,        NTYNG,    NDNZ,     NDNR,      NDNIJ,     NDNZNG,
     6   NXNZS,      NAR,      NC,       NIJG,      NBUCK,
     7   ITMA,       ITMI,     ITMIST,   ITMIGX,    ITD,
     8                         NEGSOR,               LZSAV,
     9   TIMEX,      TIMEZ,                          SDGFG1,    SCALUP
       COMMON        /CALL4/   EPG,      EPGX,      EPGM,      EPGMP,
     1                         OMEGAS,   OMEGPS,
     2                EXTRA1,  EXTRA2,   EXTRA3,    EXTRA4,    EXTRA5,
     3                KXTRA1,  KXTRA2,   KXTRA3,    KXTRA4,    KXTRA5
       COMMON        /COUTER/  KDR,      KDZ,       KMR,       KW,
     1   KNI,        KNO,                 K180,      KWHITE,    KNOR,
     2   KNORUP,     KNZR,     KNR,      KJRR,      KJZR,
     3               KNZ
       COMMON        /CINNER/  ITMIG,    IG,        IZ,        JUP,
     1   XITGG,      XITGGA,   LG,       NEGPRT,    JZ,        KX
C
C      THE DIMENSION STATEMENTS
       DIMENSION     X1(1),          X2(1)
       DIMENSION     XKI(1),         XKE(1)
       DIMENSION     MA(1),          C(1)
       DIMENSION     FG(1),          F(1)
```

```
      DIMENSION    XNOS(1),        XNOSUP(1),      XNZS(1)
      DIMENSION    XN(1),          XJR(1),         XJZ(1)
      DIMENSION    V(1),           AR(1),          AZ(1),
     1             DELZ(1)
      DIMENSION    KFING(1),       KASORG(1)
      DIMENSION    AG(1),          FNG(1),
     1             SG(1),          ASG(1),         SCG(1),
     2             SDG(1),         SING(1),        XNG(1),
     3             XNB(1),         HNLG(1),        VNLG(1),
     4             XNLG(1),        BNLG(1)
C
      DIMENSION    CT(1),          CGG(1),         CGGA(1),
     1             TVLK(1),        SISO(1),        ST(1),
     2             ARSISO(1),      AZSISO(1),      ARST(1),
     3             AZST(1),        DC(1)
      DIMENSION    XNM(1)
      DIMENSION    XNN(1),         XJRN(1),        XJZN(1)
      DIMENSION    XNA(1),         XJRA(1),        XJZA(1)
      DIMENSION                                    BUCLK(1),
     1             BUCKG(1),       MXVARI(1),      ESIJM(1)
      DIMENSION    CUPS(1),        CN2N(1)
C
C     THE FORMAT STATEMENTS
  100 FORMAT (1H1)
  104 FORMAT (1HL,21HPROGRAM STOP AT KEFN=,I8)
  120 FORMAT (126H ITMA ITMI ITMIGX ITD     TIME REQ          EPG
     1EPGM       KASOR KONV KFINSH     NEUTRON BAL          GLAM           E
     2IGEN      )
  121 FORMAT (I4,I6,I4,I7,3E14.6,I4,I5,I6,E18.8,2F16.8)
  122 FORMAT (20X,4HUPS2,8X,9HXNG(NGP1),4X,10HSING(NGP1),5X,
     1   9HSDG(NGP1),5X,8HAG(NGP1),5X,10HHNLG(NGP1),4X,10HVNLG(NGP1),
     2   4X,10HXNLG(NGP1))
  123 FORMAT (15X,8E14.6)
  124 FORMAT (1H )
  125 FORMAT (5X,5HGROUP,5X,5HITMIG,10X,10HFLUX ERROR,7X,11HERROR RATIO,
     1   5X,12HFORMER RATIO,8X,6HFACTOR,12X,5HOMEGA)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 ITMI=ITMIST                                                        ,1
      IF (KBCRED) 1005,1005,1020                                         ,2
 1005 KONV=0                                                             ,3
      IF (NFN-1) 1025,1010,1025                                          ,4
 1010 CALL FISION ( MA, V, C,  FG, XKE, XN, XJR, XJZ,  F, XNOS, XNOSUP,
     1   XNZS  )                                                        ,5
      IF (NEGSOR) 1025,1025,1015                                         ,6
 1015 KEND=1                                                             ,7
      WRITE (6,104) KEFN                                                 ,8    ,9    ,10
      KEFN=1015                                                          ,11
      GO TO 2570                                                         ,12
 1020 KBCRED=0
C                                                                        ,13
 1025 IF (ITMPRT) 1050,1050,1030                                         ,14
```

```
 1030 WRITE (6,100)
      WRITE (6,120)
      WRITE (6,122)
      WRITE (6,124)
      IF (ITMPRT-2) 1050,1035,1035
 1035 WRITE (6,125)
      WRITE (6,124)
C
C
C     THE BEGINING OF THE MAJOR ITERATION LOOP.  BEGIN THE LOOP IN
C        GROUP KGST.
C
C
 1050 ITMA =ITMA+1
      ITD=0
C
 1060 IG=KGST
      ITMIGX=0
      KFINSH=0
      KASOR=0
      EPGMP=EPGM
      EPGM=0.0
      UPS1=0.0
      UPS2=0.0
      IF (KONV-2) 1061,1070,1061
 1061 IF (ITMA-10) 1065,1065,1070
 1065 TEMP=ITMA
      GO TO 1075
 1070 TEMP=10.0
 1075 EPG=EPG/TEMP
C
C
C     ITERATIONS IN A NEW GROUP IG BEGIN HERE.  BEGIN AT THE TOP OF THE
C        REACTOR STARTING A Z DOWN AND UP PASS.
C
C
 1100 JUP=0
      IZ=NZ
      ITMIG=0
      KFING(IG)=2
      SG(IG)=0.0
      IJG=INDEX(NIJ,IG,0)
      DO 1105 IJ=1,NIJ
      IJG=IJG+1
 1105 XNN(IJ)=XN(IJG)
      IF (KS(SO) 1150,1150,1110
 1110 IJG=INDEX(NIJ,IG,0)
      DO 1115 IJ=1,NIJ
      IJG=IJG+1
 1115 XJRN(IJ)=XJR(IJG)
      IF (NZ-1) 1150,1150,1120
 1120 IJG=INDEX(NIJ,IG,0)
      DO 1125 IJ=1,NIJ
      IJG=IJG+1
 1125 XJZN(IJ)=XJZ(IJG)
C
```

```
C      TO COMPUTE THE ISOTROPIC FIXED SOURCE SISO WHICH INCLUDES THE
C         FISSION SOURCE AND THE ANISOTROPIC FIXED SOURCES ARSISO AND
C         AZSISO.  ALL THE FIXED SOURCES INCLUDE UP AND DOWN SCATTERING
C         BUT NO WITHIN GROUP SCATTERING.
C
 1150 CALL FIXED ( XKE, MA, C, F, V,    XN, XJR, XJZ,
     1    SG, SISO, ARSISO, AZSISO  )
C
C
C      ITERATIONS FOR A NEW Z DOWN AND UP PASS IN A GIVEN GROUP IG
C         BEGIN HERE.
C
C
 1200 CALL INNER (                                          XKE(1),
     1    MA(1),C(1),     X1(KDR),X1(KDZ),X1(KMR),X1(KW),       F(1),
     2    XNOS(1),XNOSUP(1),XNZS(1),                          V(1),AR(1),
     3    AZ(1),DELZ(1),KFING(1),KASORG(1),                SG(1),ASG(1),
     4                  XNG(1),HNLG(1),VNLG(1),XNLG(1),
     5    CT(1),CGG(1),CGGA(1),TVLK(1),SISO(1),ST(1),ARSISO(1),AZSISO(1),
     6    ARST(1),AZST(1),DC(1),  X1(KNI),X1(KNO),X2(KNOR),X2(KNORUP),
     7    XNM(1),X2(KNZ),X2(KNZR),          XNN(1),XNA(1),  X2(KNR),
     8           XJRN(1),XJZN(1),XJRA(1),XJZA(1),X2(KJRR),X2(KJZR),
     9   X2(K180),X2(KWHITE),  BUCLK(1),BUCKG(1),    MXVARI(1),ESIJM(1)  )
C
C
C
 1300 KFINSH=KFINSH+IABS(KFING(IG))
      IF (KSISO) 1315,1315,1305
 1305 IF (KASORG(IG)) 1315,1315,1310
 1310 KASOR=KASOR+1
 1315 ITMI=ITMI+ITMIG
      IF (ITMIG-ITMIGX) 1325,1325,1320
 1320 ITMIGX=ITMIG
 1325 IF (EPGX-EPGM) 1400,1400,1330
 1330 EPGM=EPGX
C
C
C      TO COMPUTE THE NEUTRON BALANCE PER GROUP.
C
C
 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)
      XNLG(IG)=XNLG(IG)+BNLG(IG)
 1415 IF (LBUCK) 1425,1425,1420
 1420 TEMP=BUCLK(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=INDEX(NG,K,IG)
 1435 BNLG(IG)=BNLG(IG)+BUCKG(K1)*V(IJ)*XNN(IJ)
      XNLG(IG)=XNLG(IG)+BNLG(IG)
C
```

```
 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(NTYPS,IG,KCGG)
      K1A=INDEX(NTYPS,IG,KCA)
      DO 1535 IJ=1,NIJ
      K=IABS(MA(IJ))
      XNVIJ=V(IJ)*XNN(IJ)
      K2G=INDEX(NTYNG,K,K1G)
      K2A=INDEX(NTYNG,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(NG,K,IG)
 1535 SDG(IG)=SDG(IG)+CN2N(K1)*XNVIJ
C
 1550 XNB(IG)=(SG(IG)-XNLG(IG)-AG(IG)-SDG(IG))/XNF
C
C
C
C     OVER-RELAXATION OF THE OUTER ITERATIONS.
C
C
 2000 IF (IABS(KACCEL)-2) 2150,2005,2150
 2005 IF (ITMA-1) 2150,2150,2010
 2010 IF (KUPS) 2020,2020,2015
 2015 IF (ABS(SCALUP-1.0)-50.0*EPS) 2020,2020,2150
 2020 IF (EPGMP-10.0*EPS) 2025,2025,2150
 2025 IJG=INDEX(NIJ,IG,0)
      DO 2030 IJ=1,NIJ
      IJG=IJG+1
      XNA(IJ)=XN(IJG)
 2030 XNN(IJ)=OMEGAP*XNN(IJ)-OMEGPS*XNA(IJ)
C
      DO 2050 J=1,NZ
      IJ=INDEX(NR,J,NR)
      TEMP=ORSCAL( IJ, OMEGAP, OMEGPS,  XNA, XNN )
      DO 2050 M=1,ND
      K=INDEX(ND,J,M)
      L=INDEX(NDNZ,IG,K)
      XNOS(L)=XNOS(L)*TEMP
      IF (NZ-1) 2050,2050,2046
 2046 XNOSUP(L)=XNOSUP(L)*TEMP
 2050 CONTINUE
      IF (LZSAV) 2080,2080,2055
 2055 DO 2075 I=1,NR
      IJ=INDEX(NR,NZ,I)
      TEMP=ORSCAL( IJ, OMEGAP, OMEGPS,  XNA, XNN )
      DO 2075 M=1,ND
      K=INDEX(ND,I,M)
      L=INDEX(NDNR,IG,K)
 2075 XNZS(L)=XNZS(L)*TEMP
 2080 IF (KSISO) 2150,2150,2085
 2085 DO 2090 IJ=1,NIJ
```

```
      TEMP=ORSCAL( IJ, OMEGAP, OMEGPS,  XNA, XNN )
      XJRN(IJ)=XJRN(IJ)*TEMP
      IF (NZ-1) 2090,2090,2089
 2089 XJZN(IJ)=XJZN(IJ)*TEMP
 2090 CONTINUE
C
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=1,NIJ
      IJG=IJG+1
      M=IABS(MA(IJ))
      K=INDEX(NG,M,IG)
      TEMP=V(IJ)*CUPS(K)*(XN(IJG)-XNN(IJ))
      UPS1=UPS1+TEMP
 2165 UPS2=UPS2+ABS(TEMP)
C
C     FINISHED WITH GROUP.
C
 1800 IJG=INDEX(NIJ,IG,0)
      DO 1805 IJ=1,NIJ
      IJG=IJG+1
 1805 XN(IJG)=XNN(IJ)
      IF (KSISO) 1900,1900,1810
 1810 IJG=INDEX(NIJ,IG,0)
      DO 1815 IJ=1,NIJ
      IJG=IJG+1
 1815 XJR(IJG)=XJRN(IJ)
      IF (NZ-1) 1900,1900,1820
 1820 IJG=INDEX(NIJ,IG,0)
      DO 1825 IJ=1,NIJ
      IJG=IJG+1
 1825 XJZ(IJG)=XJZN(IJ)
C
C
C     GROUP INDEXING.
C
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
C
C
C     SUMS OF GROUP QUANTITIES ARE COMPUTED AND STORED IN THE NG+1
C        POSITION.
C
C
 2200 SING(NGP1)=0.0
      SDG(NGP1)=0.0
      SCG(NGP1)=0.0
      XNLG(NGP1)=0.0
      HNLG(NGP1)=0.0
      IF (NZ-1) 2202,2202,2201
 2201 VNLG(NGP1)=0.0
 2202 IF (NBF-1) 2203,2203,2204
 2203 IF (NBUCK) 2205,2205,2204
 2204 BNLG(NGP1)=0.0
 2205 AG(NGP1)=0.0
      XNB(NGP1)=0.0
      XNG(NGP1)=0.0
      DO 2215 IG=1,NG
      XNG(NGP1)=XNG(NGP1)+XNG(IG)
      SING(NGP1)=SING(NGP1)+SING(IG)
      SDG(NGP1)=SDG(NGP1)+SDG(IG)
      SCG(NGP1)=SCG(NGP1)+SCG(IG)
      XNLG(NGP1)=XNLG(NGP1)+XNLG(IG)
      HNLG(NGP1)=HNLG(NGP1)+HNLG(IG)
      IF (NZ-1) 2207,2207,2206
 2206 VNLG(NGP1)=VNLG(NGP1)+VNLG(IG)
 2207 IF (NBF-1) 2208,2208,2209
 2208 IF (NBUCK) 2210,2210,2209
 2209 BNLG(NGP1)=BNLG(NGP1)+BNLG(IG)
 2210 AG(NGP1)=AG(NGP1)+AG(IG)
 2215 XNB(NGP1)=XNB(NGP1)+XNB(IG)
C
C     CALCULATE UP SCATTERING SCALING.
C
 2225 IF (KUPS) 2300,2300,2226
 2226 SCALUP=FG(NGP1)/(FG(NGP1)+UPS1)
      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 2285 IG=KUPS,ITEMP
      IJG=INDEX(NIJ,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(ND,J,M)
      L=INDEX(NDNZ,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,IG,K)
 2255 XNZS(L)=SCALUP*XNZS(L)
 2260 IF (KSISO) 2285,2285,2265
 2265 IJG=INDEX(NIJ,IG,0)
      DO 2270 IJ=1,NIJ
      IJG=IJG+1
 2270 XJR(IJG)=SCALUP*XJR(IJG)
      IF (NZ-1) 2285,2285,2271
 2271 IJG=INDEX(NIJ,IG,0)
      DO 2275 IJ=1,NIJ
      IJG=IJG+1
 2275 XJZ(IJG)=SCALUP*XJZ(IJG)
 2285 CONTINUE
C
C     NEW LAMBDA QUANTITIES COMPUTED.
C
 2300 GLAM1=GLAM
      IF (ABS(SDG(NGP1))-0.1E-6) 2305,2305,2310
 2305 SDGFG1=1.0
      SCAT=1.0
      GO TO 2350
 2310 TEMP=SDG(NGP1)/FG(NGP1)
      SCAT=SDGFG1/TEMP
      SDGFG1=TEMP
C
C
C     TO COMPUTE THE NEW FISSION SOURCE.
C
C
 2350 EPG1=EPG
      CALL FISION ( 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
      KONV=3
      GO TO 2500
C
C
C     COMPUTE CONVERGENCE NUMBERS.
C
C
 2375 DIFONE=1.0-GLAM
      TEST1=ABS(DIFONE)
      TEST2=RYF*ABS(1.0-SCAT)
      TEST3=ABS(GLAM1-GLAM)
      IF (KREG) 2377,2377,2376
 2376 TEST4=0.0
```

```
      GO TO 2379
 2377 IF (N2N) 2378,2378,2376
 2378 TEST4=ABS((SING(NGP1)-SDG(NGP1))/XNF)
 2379 CALL TIMEL(TIMEY)
      TIMEZ=ABS((TIMEY-TIMEX))/3600.0
      IF (KALC-1) 2385,2380,2385
 2380 EIGEN=XKI(NGP1)/XKE(NGP1)
      GO TO 2400
 2385 IF (GLAM1) 2390,2395,2390
 2390 EXTRP1=EXTRP
      EXTRP=(EIGENP-EIGEN)/(GLAMP-GLAM)
      TEST5=ABS((EXTRP1/EXTRP)-1.0)
 2395 GO TO 2500
C
C
C     TEST CONVERGENCE NUMBERS.
C
C
 2400 IF (KFINSH) 2401,2401,2415
 2401 IF (TEST1-EPS) 2405,2405,2415
 2405 IF (TEST2-EPS) 2410,2410,2415
 2410 IF (TEST3-EPS) 2411,2411,2415
 2411 IF (TEST4-EPS) 2412,2412,2415
 2412 IF (KUPS) 2445,2445,2413
 2413 IF ((UPS2/XNG(NGP1))-EPS) 2445,2445,2415
 2415 IF (ITMI-(ITMIMX+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-10) 2447,2475,2475
 2447 KONV=2
      GO TO 2455
C
 2450 KONV=0
 2455 IF (ITMPRT) 1050,1050,2460
 2460 WRITE (6,121) ITMA,ITMI,ITMIGX,ITD,TIMEZ,EPG1,EPGM,KASOR,KONV,
     1   KFINSH,XNB(NGP1),GLAM,EIGEN
      IF (NZ-1) 2466,2466,2467
 2466 TEMP=0.0
      GO TO 2470
 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 1050
C
 2475 KONV=2
C
C
C
 2500 TIMEX=EPG1
C
 2550 IF (KEND) 2575,2575,2560
 2560 KEFN=2550
 2570 WRITE (6,104) KEFN
C
C
C
 2575 KCHAIN=3
C
C
      RETURN
C
      END
```

```
$IBFTC ORSCAL  DECK
      FUNCTION ORSCAL ( IJ, SCALE1, SCALE2,  XNA, XNN  )
C
C
C     REACTOR ANALYSIS SECTION     CLAYTON BARBER
C
C     FEBRUARY 11, 1966            VERSION TN
C
C
C     THE DIMENSION STATEMENTS
      DIMENSION    XNA(1),         XNN(1)
C
C
C
 1000 TEMP=XNN(IJ)+SCALE2*XNA(IJ)
      IF (TEMP) 1005,1010,1005
 1005 ORSCAL=SCALE1*XNN(IJ)/TEMP
      RETURN
 1010 ORSCAL=1.0
      RETURN
C
C
C
      END
```

```
$IBFTC FISION  DECK
      SUBROUTINE FISION ( MA,  V,  C,  FG,  XKE,  XN,  XJR,  XJZ,  F,
     1   XNOS,  XNOSUP,  XNZS  )
C
C      SUBROUTINE FISION FOR SUBROUTINES SETUP AND OUTER OF TDSN
C
C
C
C
C
C
C
C
C      TO COMPUTE THE FISSION NEUTRON SOURCE DENSITY F(IJ), FISSION
C         NEUTRON SOURCE GROUP SUMS FG(IG), AND THE NEW LAMBDA.  ALSO TO
C         COMPUTE THE EFFECTIVE NEUTRON FISSION SPECTRUM CHI AND THE
C         FISSION NEUTRON SOURCE GROUP SUMS USING THE NEW LAMBDA.  TO
C         NORMALIZE THE FISSION SOURCES AND THE FLUXES TO A GIVEN
C         NORMALIZATION FACTOR XNF SUCH THAT THE TOTAL FISSION SOURCE
C         EQUALS XNF.  AND FINALLY TO COMPUTE THE INNER ITERATION EPSILON
C         EPG.
C
C
C      THE COMMON STATEMENTS
       COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
       COMMON        /CALL2/   KBCRED,
      1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
      2   KRBC,                          NSN,      NG,       NZ,
      3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
      4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
      5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
      6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
      7   KBCDUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
      8                        GLAM,     EIGEN,    EPS,      XNF,
      9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
       COMMON        /CALL3/
      1                        HZ,       HY,       BF,
      2                                            KCA,      KCF,
      3   KONV,      KFINSH,   KASOR,    KUPS,
      4   NBF,       ND,       NGP1,     NZP1,     NRP1,
      5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
      6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
      7   ITMA,      ITM1,     ITMIST,   ITMIGX,   ITD,
      8                        NEGSOR,             LZSAV,
      9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
       COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
      1                        OMEGAS,   OMEGPS,
      2                EXTRA1,  EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
      3                KXTRA1,  KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
C
C      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)
C
```

```
C     THE FORMAT STATEMENTS
  100 FORMAT (25H TOTAL SOURCE FG(NGP1) OF, E16.8, 24H IS ZERO OR NEGATI
     1VE.          )
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 FTP=FG(NGP1)
      NEGSOR=0
C
C     TO COMPUTE THE FISSION NEUTRON SOURCE DENSITY AND ITS VOLUME
C        INTEGRAL BY GROUPS.
C
 1010 DO 1015 IJ=1,NIJ
 1015 F(IJ)=0.0
      DO 1040 IG=1,NG
      IJG=INDEX(NIJ,IG,0)
      IF (KREG) 1020,1020,1030
 1020 DO 1025 IJ=1,NIJ
      IJG=IJG+1
      K=IABS(MA(IJ))
      K1=INDEX(NTYPS,IG,KCF)
      K2=INDEX(NTYNG,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
C
      FG(NGP1)=0.0
      IF (KREG) 1060,1060,1045
 1045 DO 1055 IG=1,NG
      FG(IG)=0.0
      K1=INDEX(NTYPS,IG,KCF)
      DO 1050 IJ=1,NIJ
      K=IABS(MA(IJ))
      K2=INDEX(NTYNG,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 (KALC) 1100,1100,1075
 1075 IF (FG(NGP1)) 1080,1080,1100
 1080 WRITE (6,100) FG(NGP1)
      NEGSOR=1
C
C     TO CALCULATE THE NEW LAMBDA.
C
```

```
 1100 IF (NFN-1) 1105,1150,1105
 1105 GLAM=FG(NGP1)/FTP
C
C     TO NORMALIZE THE FISSION SOURCES AND THE FLUXES TO A GIVEN INPUT
C        NORMALIZATION FACTOR XNF SUCH THAT THE TOTAL FISSION NEUTRON
C        SOURCE WILL EQUAL XNF FOR THE PROBLEM SETUP AND TO NORMALIZE
C        FOR THE GROWTH FACTOR LAMBDA (GLAM) AFTER EACH POWER ITERATION.
C
 1125 IF (KALC-1) 1150,1130,1150
 1130 TEMP=1.0/GLAM
      DO 1135 IG=1,NGP1
      XKE(IG)=TEMP*XKE(IG)
 1135 FG(IG)=TEMP*FG(IG)
      IF (KREG) 1175,1175,1140
 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
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 (KSISO) 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
C     TO COMPUTE THE EPSILON EPG FOR THE INNER ITERATIONS.
C
C
 1350 TEMP=NG+3
      TEMP=40.0*EPS/TEMP
      EPG=TEMP*FG(NGP1)
C
C
C
 1400 RETURN
C
      END
```

```
$IBFTC FIXED   DECK
      SUBROUTINE FIXED ( XKE, MA, C, F, V,   XN, XJR, XJZ,
     1   SG, SISO, ARSISO, AZSISO  )
C
C      SUBROUTINE FIXED COMPUTES THAT PORTION OF THE SOURCE THAT REMAINS
C         FIXED WITHIN THE GROUP (INNER) ITERATIONS.
C
C
C
C
C
C
C
C      THE COMMON STATEMENTS
       COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
       COMMON        /CALL2/   KBCRED,
      1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
      2   KRBC,                          NSN,      NG,       NZ,
      3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
      4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
      5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
      6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
      7   KBCOUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
      8                        GLAM,     EIGEN,    EPS,      XNF,
      9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
       COMMON        /CALL3/
      1                        HZ,       HY,       BF,
      2                                            KCA,      KCF,
      3   KONV,      KFINSH,   KASOR,    KUPS,
      4   NBF,       ND,       NGP1,     NZP1,     NRP1,
      5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
      6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
      7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
      8                        NEGSOR,             LZSAV,
      9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
       COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
      1                        OMEGAS,   OMEGPS,
      2                EXTRA1,   EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
      3                KXTRA1,   KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
       COMMON        /CINNER/  ITMIG,    IG,       IZ,       JUP,
      1   XITGG,     XITGGA,   LG,       NEGPRT,   JZ,       KX
C
C      THE DIMENSION STATEMENTS
       DIMENSION     XKE(1)
       DIMENSION     MA(1),          C(1)
       DIMENSION     F(1)
       DIMENSION     V(1)
       DIMENSION     XN(1),          XJR(1),         XJZ(1)
       DIMENSION     SG(1)
       DIMENSION     SISO(1),        ARSISO(1),      AZSISO(1)
C
C      THE INTEGER FUNCTIONS
       INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
```

```
 1000 IF (KREG) 1140,1140,1130
 1130 K1=INDEX(NTYPS,IG,KCF)
      DO 1135 IJ=1,NIJ
      K=IABS(MA(IJ))
      K2=INDEX(NTYNG,K,K1)
 1135 SISO(IJ)=C(K2)*F(IJ)
      GO TO 1150
 1140 DO 1145 IJ=1,NIJ
 1145 SISO(IJ)=XKE(IG)*F(IJ)
C
 1150 IF (KSISO) 1155,1155,1151
 1151 DO 1152 IJ=1,NIJ
 1152 ARSISO(IJ)=0.0
      IF (NZ-1) 1155,1155,1153
 1153 DO 1154 IJ=1,NIJ
 1154 AZSISO(IJ)=0.0
C
C     UP SCATTERING
C
 1155 KSCUP=KCGG-(KCTR+1)
      N=IG
      ITY=KCGG
      IF (KSCUP) 1250,1250,1160
 1160 IF (KREG) 1161,1161,1163
 1161 IF ((IG+1)-KUPS) 1250,11161,11161
11161 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 ITY=ITY-1
      IF (KREG) 1171,1171,1172
 1171 K1=INDEX(NTYPS,IG,ITY)
      GO TO 1175
 1172 K1=INDEX(NTYPS,N,ITY)
 1175 IJG=INDEX(NIJ,N,0)
      DO 1200 IJ=1,NIJ
      IJG=IJG+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(NTYNG,K,K1)
      IF (C(K2)) 1185,1190,1185
 1185 SISO(IJ)=SISO(IJ)+C(K2)*XN(IJG)
 1190 IF (KSISO) 1200,1200,1191
 1191 IF (KA-K) 1200,1200,1192
 1192 K2=INDEX(NTYNG,KA,K1)
      IF (C(K2)) 1195,1200,1195
 1195 ARSISO(IJ)=ARSISO(IJ)+C(K2)*XJR(IJG)
      IF (NZ-1) 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-KUPS) 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-1) 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 K1=INDEX(NTYPS,IG,ITY)
      GO TO 1280
 1270 N=N-1
      IF (NTYPS-ITY) 1260,1275,1275
 1275 K1=INDEX(NTYPS,N,ITY)
 1280 IJG=INDEX(NIJ,N,0)
      DO 1325 IJ=1,NIJ
      IJG=IJG+1
      K=IABS(MA(IJ))
      KA=K
      IF (KSISO) 1290,1290,1286
 1286 IF (MA(IJ)) 1287,1290,1290
 1287 KA=K+1
 1290 K2=INDEX(NTYNG,K,K1)
      IF (C(K2)) 1295,1300,1295
 1295 SISO(IJ)=SISO(IJ)+C(K2)*XN(IJG)
 1300 IF (KSISO) 1325,1325,1305
 1305 IF (KA-K) 1325,1325,1310
 1310 K2=INDEX(NTYNG,KA,K1)
      IF (C(K2)) 1315,1325,1315
 1315 ARSISO(IJ)=ARSISO(IJ)+C(K2)*XJR(IJG)
      IF (NZ-1) 1325,1325,1320
 1320 AZSISO(IJ)=AZSISO(IJ)+C(K2)*XJZ(IJG)
 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,NIJ
      SISO(IJ)=SISO(IJ)*V(IJ)
 1355 SG(IG)=SG(IG)+SISO(IJ)
      IF (KSISO) 1400,1400,1360
 1360 DO 1365 IJ=1,NIJ
 1365 ARSISO(IJ)=ARSISO(IJ)*V(IJ)
      IF (NZ-1) 1400,1400,1370
 1370 DO 1375 IJ=1,NIJ
 1375 AZSISO(IJ)=AZSISO(IJ)*V(IJ)
C
C
C
 1400 RETURN
C
      END
```

```
$IBFTC INNER   DECK
      SUBROUTINE INNER (          XKE, MA, C, DR, DZ, MR, W,        F,
     1   XNOS, XNOSUP, XNZS,
     2   V,AR,AZ,DELZ,      KFING,KASORG,        SG,ASG,         XNG,HNLG,
     3   VNLG,XNLG,       CT,CGG,CGGA,TVLK,SISO,ST,ARSISO,AZSISO,ARST,
     4   AZST,DC,    XNI,XNO,XNOR,XNORUP,XNM,XNZ,XNZR,          XNN,XNA,
     5   XNR,XJRN,XJZN,XJRA,XJZA,XJRR,XJZR,     XNM180,WHITE,
     6   BUCLK,BUCKG,      MXVARI,ESIJM       )
C
C      SUBROUTINE INNER PREFORMS THE INNER (OR MINOR) ITERATIONS IN A
C         GIVEN GROUP IG.
C
C
C
C
C
C
C
C
C      THE COMMON STATEMENTS
       COMMON         /CALL1/   KCHAIN,    KEND,      KEFN
       COMMON         /CALL2/   KBCRED,
     1   KREG,        KALC,     KGEO,      KBBC,      KTBC,      KLBC,
     2   KRBC,                                NSN,       NG,        NZ,
     3   NR,          NTYPS,    NMAT,      LBUCK,                KCTYP,
     4   KSISO,       KGST,     KCTR,      KCGG,      KDREAD,    KFLUXI,
     5                          ITMIMX,    ITMIGM,    NZONER,    NZONEZ,
     6   KACCEL,      KALC1,    NFN,       N2N,                  ITMPRT,
     7   KBCDUP,      KACAV,    KGAVE,     KNIP,      KEDIT,
     8                          GLAM,      EIGEN,     EPS,       XNF,
     9   RYF,         TIMAX,    CALC1,                OMEGA,     OMEGAP
       COMMON         /CALL3/
     1                          HZ,        HY,        BF,
     2                                                KCA,       KCF,
     3   KONV,        KFINSH,   KASOR,     KUPS,
     4   NBF,         ND,       NGP1,      NZP1,      NRP1,
     5   NIJ,         NTYNG,    NDNZ,      NDNR,      NDNIJ,     NDNZNG,
     6   NXNZS,       NAR,      NC,        NIJG,      NBUCK,
     7   ITMA,        ITMI,     ITMIST,    ITMIGX,    ITD,
     8                          NEGSOR,               LZSAV,
     9   TIMEX,       TIMEZ,                          SDGFG1,    SCALUP
       COMMON         /CALL4/   EPG,       EPGX,      EPGM,      EPGMP,
     1                          OMEGAS,    OMEGPS,
     2                EXTRA1,   EXTRA2,    EXTRA3,    EXTRA4,    EXTRA5,
     3                KXTRA1,   KXTRA2,    KXTRA3,    KXTRA4,    KXTRA5
       COMMON         /CINNER/  ITMIG,     IG,        IZ,        JUP,
     1   XITGG,       XITGGA,   LG,        NEGPRT,    JZ,        KX
C
C      THE DIMENSION STATEMENTS
       DIMENSION      XKE(1)
       DIMENSION      MA(1),          C(1)
       DIMENSION      DR(1),          DZ(1),          MR(1),
     1                W(1)
       DIMENSION      F(1)
       DIMENSION      XNOS(1),        XNOSUP(1),      XNZS(1)
       DIMENSION      V(1),           AR(1),          AZ(1),
     1                DELZ(1)
       DIMENSION      KFING(1),       KASORG(1)
       DIMENSION      SG(1),          ASG(1),
     1                                XNG(1),         HNLG(1),
     2                VNLG(1),        XNLG(1)
C
```

```
      DIMENSION    CT(1),          CGG(1),         CGGA(1),
     1             TVLK(1),        SISO(1),        ST(1),
     2             ARSISO(1),      AZSISO(1),      ARST(1),
     3             AZST(1),        DC(1)
      DIMENSION    XNI(1),         XNO(1),         XNOR(1),
     1             XNORUP(1),      XNM(1),         XNZ(1),
     2             XNZR(1)
      DIMENSION    XNN(1),         XNA(1),         XNR(1),
     1             XJRN(1),        XJZN(1),        XJRA(1),
     2             XJZA(1),        XJRR(1),        XJZR(1)
      DIMENSION    XNM180(1),      WHITE(1),       BUCLK(1),
     1             BUCKG(1),       MXVARI(1),      ESIJM(1)
C
C     THE FORMAT STATEMENTS
  150 FORMAT (1X, 6HSCALE=,F10.6,8H FOR IG=,I3,11H AND ITMIG=,I3,
     1   19H AND WAS SET TO 1.0 )
  200 FORMAT (2I10,5X,5E17.7)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
C
C     ITERATIONS FOR A NEW Z DOWN AND UP PASS IN A GIVEN GROUP IG
C        BEGIN HERE.
C
C
  990 NEGPRT=0
C
 1000 ITMIG=ITMIG+1
      ITD=ITD+1
C
C     TO INITALIZE THE NEW FLUX AND CURRENTS TO ZERO FOR THE DOWN AND
C        UP PASS.
C
 1025 DO 1050 IJ=1,NIJ
      XNA(IJ)=XNN(IJ)
 1050 XNN(IJ)=0.0
      IF (KSISO) 1100,1100,1060
 1060 DO 1065 IJ=1,NIJ
      XJRA(IJ)=XJRN(IJ)
 1065 XJRN(IJ)=0.0
      IF (NZ-1) 1100,1100,1070
 1070 DO 1075 IJ=1,NIJ
      XJZA(IJ)=XJZN(IJ)
 1075 XJZN(IJ)=0.0
C
C     ITITALIZATIONS FOR THE TOP BOUNDARY AND THE INITIALIZATION TO
C        ZERO OF SUMS PREFORMED DURING A DOWN AND UP PASS ARE DONE HERE.
C
 1100 ESC=0.0
      ESIJM(IG)=0.0
      ESM=0.0
      EAM=0.0
      XNG(IG)=0.0
      IF (KSISO) 1102,1102,1101
 1101 ASG(IG)=0.0
 1102 XITGG=0.0
      XITGGA=0.
      LG=0
```

```
C
      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.
      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,NR
      DO 1120 M=1,ND
      K=INDEX(ND,I,M)
 1120 XNZ(K)=0.
      GO TO 1200
 1125 DO 1130 I=1,NR
      DO 1130 M=1,ND
      K=INDEX(ND,I,M)
      K1=INDEX(NDNR,IG,K)
 1130 XNZ(K)=XNZS(K1)
C
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,NR
      TEMP=0.0
      TEMP1=0.0
      DO 1160 M=1,ND
      K=INDEX(ND,I,M)
      TEMP=TEMP-W(M)*DZ(M)*XNZ(K)
 1160 TEMP1=TEMP1-W(M)*DZ(M)
      TEMP1=TEMP/TEMP1
      DO 1165 M=1,ND
      K=INDEX(ND,I,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,I,M)
 1190 TLG=TLG+W(M)*DZ(M)*XNZ(K)
 1195 VNLG(IG)=VNLG(IG)+TLG*AZ(I)
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
 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.
C
C
```

```
 1300 KX=0
      GO TO 1310
 1305 KX=1
 1310 JZ=IZ
      CALL SORAXS (              MA,     V,      C,          CT,    CGG,
     1   CGGA,    TVLK,    BUCLK,    BUCKG     )
      IF (LG) 1325,1325,1400
C
C     TO COMPUTE THE TOTAL BASIC SOURCE BY ADDING THE SELF-SCATTERING
C        SOURCE TO THE FIXED SOURCE FOR THE GIVEN Z MESH INTERVAL IZ.
C
 1325 DO 1326 I=1,NR
      IJ=INDEX(NR,IZ,I)
 1326 ST(I)=SISO(IJ)+CGG(I)*XNA(IJ)
      IF (KSISO) 1400,1400,1350
 1350 KASORG(IG)=0
      DO 1385 I=1,NR
      TEMP=3.0*ST(I)
      IJ=INDEX(NR,IZ,I)
      ARST(I)=ARSISO(IJ)+CGGA(I)*XJRA(IJ)
      IF (TEMP-ABS(ARST(I))) 1355,1360,1360
 1355 KASORG(IG)=KASORG(IG)+1
      ARST(I)=SIGN(TEMP,ARST(I))
 1360 IF (NZ-1) 1370,1370,1361
 1361 AZST(I)=AZSISO(IJ)+CGGA(I)*XJZA(IJ)
      IF (TEMP-ABS(AZST(I))) 1365,1370,1370
 1365 KASORG(IG)=KASORG(IG)+1
      AZST(I)=SIGN(TEMP,AZST(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
C
C     TO COMPUTE THE ANGULAR FLUX FOR ND DIRECTIONS (IN EACH HALF OF A
C        Z DOWN AND UP PASS FOR A TOTAL OF 2*ND DIRECTIONS) BY RADIAL
C        MESH INTERVALS AND TO SUM IN ORDER TO OBTAIN THE NEW AVERAGED
C        FLUXES XNN.
C
C
 1400 XLLG=0.0
      CALL FLUX (            AR(1),AZ(1),DELZ(1),DR(1),DZ(1),MR(1),W(1),
     1   CT(1),ST(1),ARST(1),AZST(1),DC(1),      XNI(1),XNO(1),XNOS(1),
     2   XNOSUP(1),XNM(1),XNZ(1),            XNN(1),    XJRN(1),XJZN(1),
     3   XNM180(1),WHITE(1),      XLLG     )
C
C
C     TO COMPUTE THE HORIZONTAL NEUTRON LEAKAGES.
C
C
 1450 RLG=0.0
      DO 1455 M=1,ND
 1455 RLG=RLG+W(M)*DR(M)*XNO(M)
      RLG=RLG*AR(NRP1)*DELZ(IZ)
      HNLGR=HNLGR+RLG
      GO TO (1456,1460,1460), KGEO
 1456 IF (KLBC) 1457,1457,1460
 1457 XLLG=XLLG*AR(1)*DELZ(IZ)
      HNLGL=HNLGL+XLLG
C
```

```
C      TO STORE THE RADIAL BOUNDARY ANGULAR FLUXES.
C
 1460 DO 1465 M=1,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(M)
 1465 CONTINUE
C
C
C      TO CONTROL THE Z DOWN AND UP PASS.
C
C
 1470 IF (NZ-1) 1600,1600,1475
 1475 IF (JUP) 1480,1480,1490
 1480 IF (IZ-1) 1500,1500,1485
 1485 IZ=IZ-1
      IF (LG) 1300,1300,1305
 1490 IF (IZ-NZ) 1495,1540,1540
 1495 IZ=IZ+1
      IF (LG) 1300,1300,1305
C
C
C      TO COMPUTE THE VERICAL BOTTOM BOUNDARY NEUTRON LEAKAGES.
C
C
 1500 BLG=0.0
      IF (KBBC) 1505,1505,1525
 1505 DO 1515 I=1,NR
      TEMP=0.0
      DO 1510 M=1,ND
      K=INDEX(ND,I,M)
 1510 TEMP=TEMP-W(M)*DZ(M)*XNZ(K)
 1515 BLG=BLG+TEMP*AZ(I)
      DO 1520 I=1,NR
      DO 1520 M=1,ND
      K=INDEX(ND,I,M)
 1520 XNZ(K)=0.
C
C      RE-STORE THE OUTWARD DIRECTED ANGULAR FLUXES AT THE BOTTOM
C         BOUNDARY TO BE UPWARD DIRECTED FLUXES AFTER UNDER GOING 180
C         DEGREE ROTATION AND REFLECTION ABOUT THE FIRST DIRECTION
C         MID-POINT.
C
 1525 IF (KBBC-4) 1538,1526,1538
 1526 ITEMP2=MOD(NR,2)
      ITEMP1=NR/2
      IF (ITEMP2) 1528,1528,1527
 1527 ITEMP1=ITEMP1+1
 1528 K=0
      ITEMP=NRP1
      DO 1537 I=1,ITEMP1
      ITEMP=ITEMP-1
      K1=K
      DO 1529 M=1,ND
      K1=K1+1
 1529 XNO(M)=XNZ(K1)
      MM=0
      DO 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)=XNM180(L)
      GO TO 1532
 1531 M1=MR(M)
      K1=INDEX(ND,ITEMP,M1)
      XNZ(K)=XNZ(K1)
 1532 CONTINUE
      IF (I-ITEMP1) 1533,11532,1537
11532 IF (ITEMP2) 1533,1533,1537
 1533 MM=0
      K2=INDEX(ND,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,I)
      XNZ(K2)=XNM180(L)
      GO TO 1536
 1535 M1=MR(M)
      XNZ(K2)=XND(M1)
 1536 CONTINUE
 1537 CONTINUE
C
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
C
C     TO COMPUTE THE VERICAL 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(ND,I,M)
 1545 TEMP=TEMP+W(M)*DZ(M)*XNZ(K)
 1550 TLG=TLG+TEMP*AZ(I)
      VNLG(IG)=VNLG(IG)+TLG
C
C     TO STORE THE TOP BOUNDARY ANGULAR FLUXES.
C
      IF (LZSAV) 1600,1600,1560
 1560 DO 1565 I=1,NR
      DO 1565 M=1,ND
      K=INDEX(ND,I,M)
      L=INDEX(NDNR,IG,K)
 1565 XNZS(L)=XNZ(K)
C
C
 1600 IF (LG) 1610,1610,1650
C
C
```

```
C     TO CHECK FOR THE NECESSITY OF MAKING A NET LEAKAGE
C        CORRECTION PASS AND TO SAVE CERTAIN PARAMETERS WHEN THIS IS
C        REQUIRED.
C
C
 1610 RAHLK=0.0
      HLK=0.0
      IF (KRBC-1) 1613,1611,1611
 1611 TEMP=0.0
      DO 1612 J=1,NZ
      IJ=INDEX(NR,J,NR)
 1612 TEMP=TEMP+XNN(IJ)
      TEMP=ABS(TEMP)
      HLK=ABS(HNLGR)
      IF (TEMP) 11612,1613,11612
11612 RAHLK=HLK/TEMP
 1613 IF (NZ-1) 1618,1618,1614
 1614 IF (KTBC-1) 1618,1615,1615
 1615 TEMP=0.0
      DO 1616 I=1,NR
      IJ=INDEX(NR,NZ,I)
 1616 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 (RAHLK-(EPG/10.0)) 1619,1619,1620
 1619 IF (HLK-(EPG/10.0)) 1700,1700,1620
 1620 IF (ITMIG-1) 1700,1700,11619
11619 IF (EPGX-0.1) 11620,11620,1700
11620 LG=1
      DO 1623 IJ=1,NIJ
      XNR(IJ)=XNN(IJ)
      XNN(IJ)=0.0
      IF (KSISO) 1623,1623,1621
 1621 XJRR(IJ)=XJRN(IJ)
      XJRN(IJ)=0.0
      IF (NZ-1) 1623,1623,1622
 1622 XJZR(IJ)=XJZN(IJ)
      XJZN(IJ)=0.0
 1623 CONTINUE
      DO 1625 J=1,NZ
      DO 1625 M=1,ND
      K=INDEX(ND,J,M)
      L=INDEX(NDNZ,IG,K)
      XNOR(K)=XNOS(L)
      IF (NZ-1) 1625,1625,1624
 1624 XNORUP(K)=XNOSUP(L)
 1625 CONTINUE
      IF (LZSAV) 1631,1631,1626
 1626 DO 1630 I=1,NR
      DO 1630 M=1,ND
      K=INDEX(ND,I,M)
      L=INDEX(NDNR,IG,K)
      XNZR(K)=XNZS(L)
      IF (KTBC) 1627,1627,1628
 1627 XNZ(K)=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.
 1635 HNLG(IG)=HNLGR
      HNLGR=0.0
      HNLGLR=HNLGL
      HNLGL=0.0
      DO 1640 I=1,NR
      ST(I)=0.0
      IF (KSISO) 1640,1640,1636
 1636 ARST(I)=0.0
      IF (NZ-1) 1640,1640,1637
 1637 AZST(I)=0.0
 1640 CONTINUE
      IF (NZ-1) 1400,1400,1641
 1641 DO 1645 M=1,ND
 1645 DZ(M)=-DZ(M)
      VNLGR=VNLG(IG)
      VNLG(IG)=0.0
      BLGR=BLG
      JUP=0
      IZ=NZ
      GO TO 1150
C
C     TO AVERAGE THE SAVED QUANTIES WITH THE CORRECTED VALUES SUCH
C        THAT THE NET CORRECTED PORTION OF THE LEAKAGE WILL BE ZERO.
C
 1650 TEMP1=0.0
      TEMP2=0.0
      IF (KRBC) 1652,1652,1651
 1651 TEMP1=TEMP1+HNLG(IG)
      TEMP2=TEMP2+HNLGR
 1652 IF (KTBC) 1654,1654,1653
 1653 TEMP1=TEMP1+VNLGR
      TEMP2=TEMP2+VNLG(IG)
 1654 IF (TEMP2) 1660,1655,1660
 1655 TEMP2=0.1E-10
 1660 TEMP=-TEMP1/TEMP2
      HNLGR=TEMP*HNLGR+HNLG(IG)
      HNLGL=TEMP*HNLGL+HNLGLR
      IF (NZ-1) 1657,1657,1656
 1656 VNLG(IG)=TEMP*VNLG(IG)+VNLGR
      BLG=TEMP*BLG+BLGR
 1657 DO 1665 IJ=1,NIJ
      XNN(IJ)=TEMP*XNN(IJ)+XNR(IJ)
      IF (KSISO) 1665,1665,1663
 1663 XJRN(IJ)=TEMP*XJRN(IJ)+XJRR(IJ)
      IF (NZ-1) 1665,1665,1664
 1664 XJZN(IJ)=TEMP*XJZN(IJ)+XJZR(IJ)
 1665 CONTINUE
      DO 1670 J=1,NZ
      DO 1670 M=1,ND
      K=INDEX(ND,J,M)
      L=INDEX(NDNZ,IG,K)
      XNOS(L)=TEMP*XNOS(L)+XNOR(K)
      IF (NZ-1) 1670,1670,1668
 1668 XNOSUP(L)=TEMP*XNOSUP(L)+XNORUP(K)
 1670 CONTINUE
      IF (LZSAV) 1700,1700,1675
 1675 DO 1680 I=1,NR
      DO 1680 M=1,ND
```

```
      K=INDEX(ND,I,M)
      L=INDEX(NDNR,IG,K)
 1680 XNZS(L)=TEMP*XNZS(L)+XNZR(K)
C
C
C     TO COMPUTE THE NET LEAKAGES.
C
C
 1700 HNLG(IG)=HNLGR+HNLGL
      XNLG(IG)=HNLG(IG)
      IF (NZ-1) 1720,1720,1705
 1705 VNLG(IG)=VNLG(IG)+BLG
      XNLG(IG)=XNLG(IG)+VNLG(IG)
C
C
C     TO APPLY ACCELERATION BY OVER-RELAXATION USING THE FLEER METHOD
C        OF CHANGING THE OVER-RELAXATION FACTOR FOR THE REST OF A
C        GROUPS ITERATIONS AFTER CERTAIN CONDITIONS ARE MET IF KACCEL
C        IS NEGATIVE.
C
C
 1720 IF (KACCEL) 1723,1721,1722
 1721 ISCALE=0
      GO TO 175
 1722 OMEGAT=0.
      OMEGTP=0.
      IF (ITMIG-1) 1721,1721,1730
 1723 IF (ITMIG-1) 1724,1724,1725
 1724 OMEGAT=0.
      FLXER=1.0
      ISCALE=-1
C
 1725 IF (ISCALE-2) 1726,1748,1748
 1726 FLXERP=FLXER
      FLXER=0.0
      DO 1727 IJ=1,NIJ
 1727 FLXER=FLXER+ABS((XNN(IJ)-XNA(IJ))*V(IJ))
      OMEGTP=OMEGAT
      OMEGAT=FLXER/FLXERP
C
 1730 IF (EPGX-50.0*EPS) 1731,1731,1721
 1731 IF (ISCALE) 1721,1732,1733
 1732 IF (ABS(SCALE-1.0)-10.0*EPS) 1733,1733,1750
 1733 IF (KACCEL) 1735,1750,1734
 1734 FLXER=EPGX
      GO TO 1740
C
 1735 IF (ABS(OMEGAT-OMEGTP)-0.005) 1736,1736,1740
 1736 IF (OMEGAT-OMEGAS) 1740,1740,1737
 1737 FACTOR=(OMEGAT+OMEGAS)/(OMEGA*SQRT(OMEGAT))
      IF (FACTOR-1.0) 1738,1740,1740
 1738 SCALE1=2.0/(1.0+SQRT(1.0-(FACTOR**2)))
      SCALE2=SCALE1-1.0
      ISCALE=2
      GO TO 1745
C
 1740 SCALE1=OMEGA
      SCALE2=OMEGAS
      ISCALE=1
      FACTOR=0.
C
```

```
 1745 IF (ITMPRT-2) 1748,1746,1746
 1746 WRITE (6,200) IG,ITMIG,FLXER,OMEGAT,OMEGTP,FACTOR,SCALE1
C
 1748 DO 1749 IJ=1,NIJ
 1749 XNN(IJ)=SCALE1*XNN(IJ)-SCALE2*XNA(IJ)
C
C
C      STORE CERTAIN VARIABLES AND SCALE QUANTIES FOR A GIVEN GROUP IG .
C
C
 1750 K1G=INDEX(NTYPS,IG,KCGG)
      KX=1
      DO 1775 JZ=1,NZ
      CALL SORAXS (              MA,     V,      C,          CT,   CGG,
     1   CGGA,   TVLK,   BUCLK,   BUCKG
      DO 1775 I=1,NR
      IJ=INDEX(NR,JZ,I)
      XNVIJ=V(IJ)*XNN(IJ)
      XNG(IG)=XNG(IG)+XNVIJ
      K=IABS(MA(IJ))
      K2G=INDEX(NTYNG,K,K1G)
      IF (MA(IJ)) 1751,1752,1752
 1751 KA=K+1
      K2GA=INDEX(NTYNG,KA,K1G)
 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)=IJ
 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
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=ORSCAL( IJ, SCALE1, SCALE2,  XNA, XNN  )
 1803 DO 1810 M=1,ND
      K=INDEX(ND,J,M)
      L=INDEX(NDNZ,IG,K)
      XNOS(L)=XNOS(L)*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=ORSCAL( IJ, SCALE1, SCALE2,  XNA, XNN  )
 1814 DO 1820 M=1,ND
      K=INDEX(ND,I,M)
      L=INDEX(NDNR,IG,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=ORSCAL( IJ, SCALE1, SCALE2,  XNA, XNN  )
 1833 XJRN(IJ)=XJRN(IJ)*TEMP*SCALE
      IF (NZ-1) 1840,1840,1836
 1836 XJZN(IJ)=XJZN(IJ)*TEMP*SCALE
 1840 CONTINUE
C
C     SCALING FLUXES
C
 1845 DO 1846 IJ=1,NIJ
 1846 XNN(IJ)=XNN(IJ)*SCALE
C
C
C     TO CHECK THE CONVERGENCE OF THE INNER ITERATIONS.
C
C
 1850 IF (ESM-EAM) 1855,1855,1860
 1855 EPGX=EAM
      GO TO 1865
 1860 EPGX=ESM
 1865 IF (KFING(IG)-2) 1975,1866,1866
 1866 IF (ESIJM(IG)-EPG/100.0) 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(IG)-BLG)-(EPG/8.0)) 1890,1890,1900
 1890 IF (KRBC) 1925,1925,1895
 1895 IF (ABS(HNLG(IG)-HNLGL)-(EPG/8.0)) 1925,1925,1900
 1900 IF (EPGX-0.05*EPGMP) 1905,1905,1915
 1905 KFING(IG)=-1
      GO TO 1950
 1915 IF (ITMIG-ITMIGM) 1920,1935,1935
```

```
 1920 KFING(IG)=2
      GO TO 1960
 1925 KFING(IG)=0
      GO TO 1950
 1935 KFING(IG)=1
 1950 IF (EPGX-10.0*EPG) 1955,1955,1960
 1955 IF (ITMPRT-3) 1960,1956,1960
 1956 NEGPRT=1
      GO TO 1965
 1960 NEGPRT=0
 1965 JUP=0
      IZ=NZ
      GO TO 1000
C
C
C
 1975 RETURN
C
      END
$IBFTC FLUX     DECK
      SUBROUTINE FLUX (              AR, AZ, DELZ, DR, DZ, MR, W,
     1   CT, ST, ARST, AZST, DC, XNI, XNO, XNOS, XNOSUP, XNM, XNZ,
     2   XNN, XJRN, XJZN,     XNM180,WHITE,     XLLG     )
C
C      SUBROUTINE FLUX FOR SUBROUTINE INNER OF TDSN
C
C
C
C
C
C
C
C      TO COMPUTE THE ANGULAR FLUX FOR ND DIRECTIONS BY RADIAL MESH
C         INTERVALS AND SUM IN ORDER TO OBTAIN THE NEW AVERAGED
C         FLUXES XNN.  THE WEIGHTING FUNCTION W(M) SUMS TO ONE SO THE
C         AVERAGE IS AUTOMATICALLY OBTAINED.  THE CURRENTS ARE ALSO
C         COMPUTED FOR KSISO=1 .
C
C
C      THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGED,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,               KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCDUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
```

```
      COMMON        /CALL4/   EPG,       EPGX,      EPGM,      EPGMP,
     1                        OMEGAS,    OMEGPS,
     2              EXTRA1,   EXTRA2,    EXTRA3,    EXTRA4,    EXTRA5,
     3              KXTRA1,   KXTRA2,    KXTRA3,    KXTRA4,    KXTRA5
      COMMON        /CINNER/  ITMIG,     IG,        IZ,        JUP,
     1   XITGG,     XITGGA,   LG,        NEGPRT,    JZ,        KX
C
C     THE DIMENSION STATEMENTS
      DIMENSION     AR(1),              AZ(1),              DELZ(1),
     1              DR(1),              DZ(1),              MR(1),
     2              W(1)
      DIMENSION     CT(1),              ST(1),              ARST(1),
     1              AZST(1),            DC(1)
      DIMENSION     XNI(1),             XNO(1),             XNOS(1),
     1              XNOSUP(1),          XNM(1),             XNZ(1)
      DIMENSION     XNN(1),             XJRN(1),            XJZN(1)
      DIMENSION     XNM180(1),          WHITE(1)
C
C     THE FORMAT STATEMENTS
  152 FORMAT (16H ANGULAR FLUX AT,I6,9H FIXED TO,E13.5,5H FROM,E13.5,
     1   10H FOR ANGLE,I3,18H AND MESH INTERVAL,I4,9H IN GROUP,I3,
     2   14H AND ITERATION,I3)
  153 FORMAT (15H XNANG WAS ZERO,I4,90H TIMES OVER ALL ANGLES AND R MESH
     1 INTERVALS WITH NON-ZERO CROSS SECTION OF Z MESH INTERVAL,I3,
     2   10H AND GROUP,I3)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C     TO CALCULATE THE BOUNDARY ANGULAR FLUX FOR A WHITE RADIAL
C        OUTER BOUNDARY.
C
 1000 IF (KRBC-3) 1006,1001,1006
 1001 IF (JUP) 1002,1002,1006
 1002 TEMP=0.0
      WHITE(IZ)=0.0
      DO 1005 M=1,ND
      IF (DR(M)) 1005,1005,1003
 1003 K=INDEX(ND,IZ,M)
      L=INDEX(NDNZ,IG,K)
      TEMP=TEMP+W(M)*DR(M)*XNOS(L)
      WHITE(IZ)=WHITE(IZ)+W(M)*DR(M)
      IF (NZ-1) 1005,1005,1004
 1004 TEMP=TEMP+W(M)*DR(M)*XNOSUP(L)
      WHITE(IZ)=WHITE(IZ)+W(M)*DR(M)
 1005 CONTINUE
      WHITE(IZ)=TEMP/WHITE(IZ)
C
C
 1006 NZNANG=0
C
C     TO SET UP FOR 180 DEGREE BOTTOM BOUNDARY REFLECTION.
C
      I180=0
      IF (KBBC-4) 11010,1007,11010
 1007 IF (JUP) 1008,1008,11010
 1008 IF (IZ-1) 11010,1009,11010
 1009 I180=1
      M180=0
C
```

```
11010 DO 1550 M=1,ND
      J180=0
C
C     SET ANGULAR FLUX BOUNDARY CONDITIONS.
C
      IF (DR(M)) 1025,1025,1010
 1010 M1=MR(M)
      IF (I180) 11014,11014,11011
11011 IF (M-ND) 11012,11013,11013
11012 IF (ABS(W(M+1))-0.1E-09) 11013,11013,11014
11013 J180=1
11014 GO TO (1011,1020,1020), KGEO
 1011 IF (KLBC-1) 1015,1020,1020
 1015 XLLG=XLLG-W(M1)*DR(M1)*XNI(M1)
      XNO(M)=0.
      XNI(M)=0.
      GO TO 1100
 1020 XNO(M)=XNI(M1)
      XNI(M)=XNI(M1)
      GO TO 1100
 1025 IF (KRBC-1) 1030,1035,1060
 1030 K=INDEX(ND,IZ,M)
      GO TO 1040
 1035 M1=MR(M)
      K=INDEX(ND,IZ,M1)
 1040 L=INDEX(NDNZ,IG,K)
      IF (JUP) 1045,1045,1050
 1045 XNO(M)=XNOS(L)
      XNI(M)=XNOS(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)
C
C
C
 1100 DO 1500 I1=1,NR
      IF (DR(M)) 1110,1110,1105
 1105 I=I1
      GO TO 1115
 1110 I=NRP1-I1
 1115 IJ=INDEX(NR,IZ,I)
C
C     COMPUTE THE GEOMETRY FUNCTIONS.
C
 1150 IP1=I+1
      MM1=M-1
      DAR=ABS(DR(M))*(AR(IP1)+AR(I))*DELZ(IZ)
      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(MM1)*DR(MM1))*(AR(IP1)-AR(I))*DELZ(IZ)+
     1    W(MM1)*DC(I)
```

```
      DC(I)=DC(I)/W(M)
      DBT=DBT+DC(I)
 1180 DBT=DBT+CT(I)
C
C     CALCULATE THE ANGULAR AVERAGE FLUX XNANG.
C
 1200 IF (DR(M)) 1205,1205,1210
 1205 TEMP=DAR*XNI(M)
      GO TO 1215
 1210 TEMP=DAR*XNO(M)
 1215 IF (NZ-1) 1225,1225,1220
 1220 K1=INDEX(ND,I,M)
      TEMP=TEMP+DAZ*XNZ(K1)
 1225 IF (ABS(W(M))-0.1E-08) 1235,1235,1230
 1230 TEMP=TEMP+DC(I)*XNM(I)
 1235 TEMP=TEMP+ST(I)
      IF (KSISO) 1245,1245,1240
 1240 TEMP=TEMP+DR(M)*ARST(I)
      IF (NZ-1) 1245,1245,1241
 1241 TEMP=TEMP+DZ(M)*AZST(I)
 1245 XNANG=TEMP/DBT
C
C     TO CALCULATE THE ANGULAR EXTRAPOLATED FLUXES.
C
 1250 TEMP=XNANG+XNANG
      IF (ABS(W(M))-0.1E-08) 1255,1255,1260
 1255 XNM(I)=XNANG
      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 XNI(M)=TEMP-XNI(M)
      GO TO 1300
 1285 XNO(M)=TEMP-XNO(M)
C
C     TO TEST FOR A NEGATIVE EXTRAPOLATED VALUE FOR AN ANGULAR FLUX.
C
 1300 IF (LG) 1305,1305,1430
 1305 IF (XNANG) 1310,1320,133
 1310 KEFN=1310
      TEMP=XNANG
      XNANG=0.0
      IF (NEGPRT) 1330,1330,1315
 1315 WRITE (6,152) KEFN,XNANG,TEMP,M,IJ,IG,ITMIG
      GO TO 1330
 1320 IF (ABS(CT(I))-0.1E-9) 1330,1330,1325
 1325 NZNANG=NZNANG+1
 1330 IF (NZ-1) 1350,1350,1335
 1335 IF (XNZ(K1)) 1340,1350,1350
 1340 KEFN=1340
      TEMP=XNZ(K1)
      XNZ(K1)=0.0
      IF (NEGPRT) 1350,1350,1345
 1345 WRITE (6,152) KEFN,XNZ(K1),TEMP,M,IJ,IG,ITMIG
 1350 IF (XNM(I)) 1355,1365,1365
 1355 KEFN=1355
      TEMP=XNM(I)
      XNM(I)=0.
      IF (NEGPRT) 1365,1365,1360
 1360 WRITE (6,152) KEFN,XNM(I),TEMP,M,IJ,IG,ITMIG
```

```
 1365 IF (DR(M)) 1370,1370,1385
 1370 IF (XNI(M)) 1375,1425,1425
 1375 KEFN=1375
      TEMP=XNI(M)
      XNI(M)=0.
      IF (NEGPRT) 1425,1425,1380
 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
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
C
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-II CROSS SECTIONS.
C
 1450 IF (KSISO) 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
C
 1500 CONTINUE
C
C     STORE SOME ANGULAR FLUXES FOR 180 DEGREE BOTTOM BOUNDARY
C        REFLECTION.
C
      IF (J180) 1550,1550,1505
 1505 DO 1510 I=1,NR
      M180=M180+1
 1510 XNM180(M180)=XNM(I)
C
C
 1550 CONTINUE
C
      IF (NZNANG) 1600,1600,1575
 1575 WRITE (6,153) NZNANG,IZ,IG
C
C
 1600 RETURN
C
      END
```

```
$IBFTC SORAXS  DECK
      SUBROUTINE SORAXS (            MA, V, C,   CT, CGG, CGGA,
     1   TVLK, BUCLK, BUCKG    )
C
C      SUBROUTINE SORAXS FOR SUBROUTINE INNER OF TDSN
C
C      TO LOOK UP THE WITHIN GROUP SCATTERING CROSS SECTION AND THE
C         TOTAL COLLISION LOSS CROSS SECTION AND COMPUTE ANY BUCKLING
C         LOSS CROSS SECTIONS TO ADD TO THE TOTAL COLLISION LOSS CROSS
C         SECTION.
C
C
C
C
C
C
C
C      THE COMMON STATEMENTS
       COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
       COMMON        /CALL2/   KBCRED,
      1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
      2   KRBC,                          NSN,      NG,       NZ,
      3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
      4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
      5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
      6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
      7   KBCOUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
      8                        GLAM,     EIGEN,    EPS,      XNF,
      9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
       COMMON        /CALL3/
      1                        HZ,       HY,       BF,
      2                                            KCA,      KCF,
      3   KONV,      KFINSH,   KASOR,    KUPS,
      4   NBF,       ND,       NGP1,     NZP1,     NRP1,
      5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
      6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
      7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
      8                        NEGSOR,             LZSAV,
      9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
       COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
      1                        OMEGAS,   OMEGPS,
      2                EXTRA1,   EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
      3                KXTRA1,   KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
       COMMON        /CINNER/  ITMIG,    IG,       IZ,       JUP,
      1   XITGG,     XITGGA,   LG,       NEGPRT,   JZ,       KX
C
C      THE DIMENSION STATEMENTS
       DIMENSION     MA(1),          V(1),          C(1)
       DIMENSION     CT(1),          CGG(1),        CGGA(1)
       DIMENSION     TVLK(1),        BUCLK(1),      BUCKG(1)
C
C      THE INTEGER FUNCTIONS
       INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
```

```
 1000 DO 1175 I=1,NR
      IJ=INDEX(NR,JZ,I)
      K=IABS(MA(IJ))
      KA=K
      IF (KSISO) 1015,1015,1005
 1005 IF (MA(IJ)) 1010,1015,1015
 1010 KA=K+1
C
C     TO COMPUTE THE TOTAL CROSS SECTION AND ANY TRANSVERSE LEAKAGE
C        CROSS SECTION FOR A GIVEN GROUP IG AND A GIVEN Z MESH
C        INTERVAL IZ.
C
 1015 K1=INDEX(NTYPS,IG,KCTR)
      CL=0.0
      K2=INDEX(NTYNG,K,K1)
      CE=C(K2)
      IF (NBF-1) 1050,1050,1020
 1020 K2=INDEX(NTYNG,KA,K1)
      CTR=C(K2)
      TEMP=ABS(CTR)
      CL=((BF/(TEMP*HZ+1.42089216))**2)*CTR
      IF (HY-0.1E-7) 1030,1030,1025
 1025 CL=CL+((BF/(TEMP*HY+1.42089216))**2)*CTR
 1030 TVLK(IJ)=CL*V(IJ)
 1050 CE=CE+CL
      IF (LBUCK) 1052,1052,1051
 1051 CE=CE+BUCLK(IG)
 1052 IF (NBUCK) 1054,1054,1053
 1053 K2=INDEX(NG,K,IG)
      CE=CE+BUCKG(K2)
 1054 CT(I)=CE*V(IJ)
      IF (KX) 1100,1100,1175
C
C     TO STORE THE WITHIN GROUP SCATTERING CROSS SECTIONS BY VOLUME
C        MESH INTERVALS.
C
 1100 K1=INDEX(NTYPS,IG,KCGG)
      K2=INDEX(NTYNG,K,K1)
      CGG(I)=C(K2)*V(IJ)
      XITGG=XITGG+CGG(I)
      IF (KSISO) 1175,1175,1125
 1125 IF (KA-K) 1130,1130,1135
 1130 CGGA(I)=0.0
      GO TO 1175
 1135 K2=INDEX(NTYNG,KA,K1)
      CGGA(I)=C(K2)*V(IJ)
      XITGGA=XITGGA+CGGA(I)
C
C
 1175 CONTINUE
C
C
      RETURN
C
      END
```

```
$IBFTC DUMPBC  DECK
      SUBROUTINE DUMPBC
C
C     TO GIVE A BCDUMP OF THE PROBLEM.
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        X
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCOUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITM1,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
     1                        OMEGAS,   OMEGPS,
     2                EXTRA1,  EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
     3                KXTRA1,  KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
      COMMON        /COUTER/  KDR,      KDZ,      KMR,      KW,
     1   KNI,       KNO,                KI80,     KWHITE,   KNOR,
     2   KNORUP,    KNZR,     KNR,      KJRR,     KJZR,
     3              KNZ
      COMMON        /CINPT1/  LKI,      LKE,      LFG,      LKF,
     1   LKAS,      LAG,      LFNG,     LSG,      LASG,     LBNLG,
     2   LSCG,      LSDG,     LSING,    LXNG,     LJRG,     LJZG,
     3   LXNB,      LHNLG,    LVNLG,    LXNLG,    LBUC,     LMXVAR,
     4   LAR,       LAZ,      LR,       LRM,      LRA,      LRAV,
     5   LCT,       LCGG,     LCGGA,    LSISO,    LST,      LARSO,
     6   LAZSO,     LARST,    LAZST,    LCUPS,    LESIJM,   LDC,
     7   LNM,       LTVLK,    LDELZ,    LZ,       LDR,      LDZ,
     8   LMR,       LW,       LNI,      LNO,      LN2N
      COMMON        /CINPT2/  LMA,      LV,       LF,       LNN,
     1   LNA,       LJRN,     LJZN,     LJRA,     LJZA,     LJRR,
     2   LJZR,      L180,     LNOR,     LNR,      LNORUP,   LNZR,
     3   LNZ,       LNOSUP,   LNZS,     LNOS,     LN,
     4   LJR,       LJZ,                LC,       LNBUC,
```

```
     5   LWHITE,   LBCDUP,                          LMID,     LNMI,
     6   LNMR,     LRMI,     LDZ1,     LC1,
     7   LSTG,                                                LNORM,
     8                       LNMRA,    LNMZA,    LIDM,     LIDMAP,
     9   LIDGP
      COMMON       /CINPT3/                         LAST1,    LAST2,
     1   LAST3,    LAST4,    LAST
C
C      THE DIMENSION STATEMENTS
       DIMENSION    X(17500)
C
C      THE FORMAT STATEMENTS
  130 FORMAT (1H$80HEND OF BCDUMP   END OF BCDUMP    END OF BCDUMP    EN
     1D OF BCDUMP    END OF BCDUMP     )
C
C      THE INTEGER FUNCTIONS
       INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
  980 KDUP=1
       IF (KBCDUP+1) 990,1030,1000
  990 KDUP=3
       KBCDUP=-KBCDUP
       GO TO 1020
C
 1000 IF (KBCDUP-1) 1100,1005,1015
 1005 IF (KONV-1) 1010,1015,1010
 1010 IF (KONV+1) 1100,1015,1100
 1015 KDUP=2
 1020 CALL BCDUMP (KREG,LAST)
       CALL BCDUMP (X(1),X(LBCDUP))
C
 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,NIJ)
       CALL BCDUMP (X(K1),X(K2))
       GO TO (1040,1035,1035), KDUP
 1035 IF (KSISO) 1040,1040,1036
 1036 K1=ITEMP2+INDEX(NIJ,IG,1)
       K2=ITEMP2+INDEX(NIJ,IG,NIJ)
       CALL BCDUMP (X(K1),X(K2))
       IF (NZ-1) 1040,1040,1037
 1037 K1=ITEMP3+INDEX(NIJ,IG,1)
       K2=ITEMP3+INDEX(NIJ,IG,NIJ)
       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
C
 1100 RETURN
C
      END

$IBMAP BCDUMP   100,M94,DECK

            7094 RELMOD ASSEMBLY.

                  $IBLDR BCDUMP

                  $TEXT  BCDUMP
```

BCDUMP ROUTINE FOR IBSYS

```
                                                ENTRY   BCDUMP

BINARY CARD ID. BCDU0002
      00000  1 00000 0 00005  10001  BCDUMP SAVE    1,2,4
      00001  0774 00 2 00000  10000
      00002  0774 00 1 00000  10000
      00003  0774 00 4 00000  10000
      00004  0020 00 4 00001  10000
      00005  0634 00 4 00000  10011
      00006  0634 00 4 00204  10001
      00007  0634 00 4 00003  10001
      00010  0634 00 1 00002  10001
      00011  0634 00 2 00001  10001
      00012  0500 00 4 00003  10000          CLA     3,4             PICK UP FIRST ARGUMENT
      00013  0560 00 4 00004  10000          LDQ     4,4             PICK UP SECOND ARGUMENT
      00014  0040 00 0 01002  10011          TLQ     *+2
      00015  0131 00 0 00000  10000          XCA
      00016  4600 00 0 00144  10001          STQ     WD1             WD1 HAS THE FIRST ADDRESS
      00017  0534 00 1 00144  10001          LXA     WD1,1           FIRST LOCATION IN INDEX 1
      00020  0402 00 0 00144  10001          SUB     WD1
      00021  0734 00 2 00000  10000          PAX     0,2             THE NO. OF WORDS OUTPUTED IN INDEX 2
      00022  1 00001 2 01001  10011          TXI     *+1,2,1         TRUE WORD COUNT

BINARY CARD ID. BCDU0003
      00023  0634 00 1 00041  10001          SXA     IX1,1
      00024  0634 00 2 00042  10001          SXA     IX2,2
      00025  0500 60 0 00203  10001          CLA*    OUT
      00026  0621 00 0 00121  10001          STA     RITE+1
      00027  0621 00 0 00040  10001          STA     MON
      00030  0734 00 1 00000  10000          PAX     0,1
      00031  1 00001 1 01001  10011          TXI     *+1,1,1
      00032  0634 00 1 01001  10011          SXA     *+1,1
      00033  0441 00 0 00000  10000          LDI     **
      00034  4056 00  040000  10000          LNT     040000
      00035  0020 00 0 01002  10011          TRA     *+2
      00036  0020 00 0 01003  10011          TRA     *+3
      00037  0074 00 4 07000  10011          TSX     .OPEN,4
      00040  5 00000 0 00000  10000  MON     MON     **
      00041  0774 00 1 00000  10000  IX1     AXT     **,1
      00042  0774 00 2 00000  10000  IX2     AXT     **,2
      00043  7 00026 2 00131  10001  TEST    TXL     LASTC,2,22      ONLY 22 WORDS OR LESS LEFT
      00044  2 00026 2 01001  10011          TIX     *+1,2,22
      00045  0634 00 2 00042  10001          SXA     IX2,2

BINARY CARD ID. BCDU0004
      00046  0774 00 2 00026  10000          AXT     22,2
      00047  1 00500 2 01001  10011  TEST4   TXI     *+1,2,320
      00050  4634 00 2 00144  10001          SXD     WD1,2
      00051  2 00500 2 01001  10011          TIX     *+1,2,320
      00052  0634 00 1 00062  10001          SXA     CLA,1
      00053  0634 00 1 00144  10001          SXA     WD1,1
      00054  1 00026 1 01001  10011  LOOP    TXI     *+1,1,22
      00055  0634 00 1 00041  10001          SXA     IX1,1
      00056  0774 00 4 00027  10000          AXT     23,4
      00057  0600 00 4 00174  10001  CLEAR   STZ     CKSUM+23,4      CLEAR THE BUFFER
```

```
        00060  2 00001 4 41001  10011          TIX     *-1,4,1
        00061  0774 00 4 00000  10000          AXT     0,4
        00062  0500 00 4 00000  10000  CLA     CLA     **,4            FILL THE BUFFER WITH
        00063  0601 00 4 00146  10001          STO     CKSUM+1,4       NEW DATA
        00064  1 77777 4 01001  10011          TXI     *+1,4,-1
        00065  2 00001 2 41003  10011          TIX     *-3,2,1
        00066  0774 00 1 00000  10000  CNUM    AXT     **,1            CONSECUTIVELY
        00067  0500 00 0 00200  10001          CLA     HUNBIT          NUMBER
        00070  0771 00 0 00001  10000          ARS     1               THE

BINARY CARD ID. BCDU0005
        00071  7 00143 1 01002  10011          TXL     *+2,1,99        BCDUMP
        00072  1 77634 1 41002  10011          TXI     *-2,1,-100      CARDS
        00073  0621 00 0 00175  10001          STA     GP+1            FROM
        00074  0500 00 0 00202  10001          CLA     BITT            ZERO
        00075  0771 00 0 00001  10000          ARS     1               TO
        00076  7 00011 1 01002  10011          TXL     *+2,1,9         999
        00077  1 77766 1 41002  10011          TXI     *-2,1,-10
        00100  0601 00 0 00176  10001          STO     WORD3
        00101  0500 00 0 00201  10001          CLA     BITU
        00102  0771 00 0 00001  10000          ARS     1
        00103  7 00000 1 01002  10011          TXL     *+2,1,0
        00104  1 77777 1 41002  10011          TXI     *-2,1,-1
        00105  4602 00 0 00176  10001          ORS     WORD3
        00106  0534 00 1 00066  10001          LXA     CNUM,1
        00107  1 00001 1 01001  10011          TXI     *+1,1,1
        00110  7 01747 1 01002  10011          TXL     *+2,1,999
        00111  0774 00 1 00000  10000          AXT     0,1
        00112  0634 00 1 00066  10001          SXA     CNUM,1
        00113  0774 00 1 00026  10000          AXT     22,1            COMPUTE

BINARY CARD ID. BCDU0006
        00114  4500 00 0 00144  10001          CAL     WD1             THE
        00115  0361 00 1 00174  10001          ACL     GP,1            CHECK SUM
        00116  2 00001 1 41001  10011          TIX     *-1,1,1
        00117  0602 00 0 00145  10001          SLW     CKSUM
        00120  0074 00 4 10000  10011  RITE    TSX     .WRITE,4        WRITE THE BINARY CARD ON THE OUTPUT TAPE
        00121  0 00134 0 00000  10100          PZE     **,,EOF
        00122  0 00034 0 00144  10001          IOCD    WD1,,28
        00123  0020 00 0 00041  10001          TRA     IX1
        00124  0020 00 0 01001  10011  RETURN  TRA     *+1
        00125  0774 00 1 00041  10001          AXT     IX1,1
        00126  0634 00 1 00123  10001          SXA     RETURN-1,1
81      00127  0634 00 0 00066  10001          SXA     CNUM
                      00130                   RETURN  BCDUMP
        00131  0500 00 0 00124  10001  LASTC   CLA     RETURN
        00132  0601 00 0 00123  10001          STO     RETURN-1
        00133  0020 00 0 00047  10001          TRA     TEST4
        00134  000000000000     00010  EOF     CALL    .FXEM.(EOF2)
        00134  0074 00 4 04000  10011
        00135  1 00001 0 01003  10011

BINARY CARD ID. BCDU0007
        00136  0 00204 0 00140  10100
        00137  0 00000 0 00143  10001
        00140  000000000000     00010          CALL    EXIT
```

```
      00140  0074 00 4 05000  10011
      00141  1 00000 0 01002  10011
      00142  0 00204 0 00141  10100
      00143  C 00000 0 00042  10000  EOF2    PZE      34
      00144  0 00000 0 00000  10000  WD1     PZE
      00145  2000000000027    00001  CKSUM   BSS      23
      00174  420041004040     10000  GP      OCT      420041004040
      00175  104020400000     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  000020000000     10000  BITU    OCT      20000000
      00202  200000000000     10000  BITT    OCT      200000000000
      00203  0 00000 0 11000  10011  OUT     PZE      .PCH.
      00204  000000000000     10000          *LDIR
      00205  222324644447     10000

BINARY CARD ID. BCDU0008
                       00000  01111          END

          CONTROL DICTIONARY

                $CDICT BCDUMP                                                    BCDU0009


BINARY CARD ID. BCDU0010
              000206000000                PREFACE     START=0,LENGTH=134,TYPE=7094,CMPLX=5
              000004000005
              222324644447        BCDUMP DECK        LOC=0,LENGTH=134
              000206000000
              222324644447        BCDUMP REAL        LOC=0,LENGTH=0
              000000000000
              222324644447        BCDUMP REAL        LOC=0,LENGTH=0
              000000000000
              332667254433        .FXEM. VIRTUAL     SECT. 4,CALL
              200000100000
              256731636060        EXIT   VIRTUAL     SECT. 5,CALL
              200000100000
              627062434623        SYSLOC VIRTUAL     SECT. 6
              200000000000
              334647254560        .OPEN  VIRTUAL     SECT. 7
              200000000000
              336651316325        .WRITE VIRTUAL     SECT. 8
              200000000000
              334723303360        .PCH.  VIRTUAL     SECT. 9
              200000000000

                $DKEND BCDUMP                                    000751      BCDU0011
```

```
          SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL    VALUE    REFERENCES

         BCDUMP    00000    130
         BITT      00202    74
         BITU      00201    101
         CKSUM     00145    57,63,117
         CLA       00062    52
         CLEAR     00057
         CNUM      00066    106,112,127
         EOF2      00143    137
         EOF       00134    121
         GP        00174    73,115
         HUNBIT    00200    67
         IX1       00041    23,55,123,125
         IX2       00042    24,45
         ..0001    00003    7,10,11
         ..0002    00004
         ..0003    00005    0
         LASTC     00131    43
         LOOP      00054
         MON       00040    27
         OUT       00203    25
         RETURN    00124    126,131,132
         RITE      00120    26
  LCTR    BLCTR
  QUAL    UNQS
  LCTR    //
         TEST4     00047    133
         TEST      00043
         W01       00144    16,17,20,50,53,114,122
         WORD3     00176    100,105

REFERENCES TO VIRTUAL SYMBOLS.

         EXIT          5    140
         .FXEM.        4    134
         .OPEN         7    37
         .PCH.         9    203
         .WRITE        8    120
         SYSLOC        6    5

$IBMAP .PCH.    6,DECK

          7094 RELMOD ASSEMBLY.

                $IBLDR .PCH.                                                    02/25/66     .PCH0000

                $FILE  .PCH.    'PCH    *,PP,READY,OUTPUT,BLK=28,MULTIREEL,BIN,NOLIST      .PCH0001
```

```
          FILE DICTIONARY.

                  $FDICT .PCH.                                                    .PCH0002


BINARY CARD ID. .PCH0003
                205C02000034  PCH     FILE    *PCH                    BIN,OUTPUT,NOHDVN,BLK=28
                000000000000
                472330606060
                606060606060
                6C6C606C6060
                     $TEXT  .PCH.                                               .PCH0004

                                              ENTRY   .PCH.

BINARY CARD ID. .PCH0005
       00000  0 00000 0 04001  10010  .PCH.  PZE     PCH
                                     PCH    FILE    ,PP,READY,OUTPUT,BLK=28,MULTIREEL,BIN,NOLIST
                           00000  01111          END

          CONTROL DICTIONARY

                  $CDICT .PCH.                                                  .PCH0006


BINARY CARD ID. .PCH0007
                000001000000               PREFACE     START=0,LENGTH=1,TYPE=7094,CMPLX=5
                000004000005
                334723303360       .PCH.   DECK        LOC=0,LENGTH=1
                000001000000
                334723303360       .PCH.   REAL        LOC=0,LENGTH=0
                000000000000

                     $DKEND .PCH.                                  001184       .PCH0008

          SYMBOL REFERENCE DATA


REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         .PCH.    00000
  FILE   PCH          1   0
  LCTR    BLCTR
  QUAL    UNCS
  LCTR   //
```

```
$IBFTC OUTPUT  DECK
      SUBROUTINE OUTPUT (MA,                  FG,F,              XN,XJR,XJZ,
     1   V,                KFING,KASORG,      AG,FNG,SG,ASG,SCG,SDG,
     2   SING,XNG,XJRG,XJZG,XNB,HNLG,VNLG,XNLG,BNLG,            STG,
     3                              XNORM,      BUCLK,    MXVARI,ESIJM  )
C
C      SUBROUTINE OUTPUT IS THE OVERLAY (OR CHAIN) SUBROUTINE OF TDSN
C         THAT WRITES OUT THE FINAL VALUES OF THE ITERATED QUANTITIES
C         AND COMPUTES AND WRITES OUT EXTRA OUTPUT BASED ON THE
C         FINAL ITERATED VALUES.
C
C
C
C
C
C
C
C      THE COMMON STATEMENTS
       COMMON        /CALL1/   KCHAIN,    KEND,      KEFN
       COMMON        /CALL2/   KBCRED,
      1   KREG,      KALC,     KGEO,      KBBC,      KTBC,      KLBC,
      2   KRBC,                           NSN,       NG,        NZ,
      3   NR,        NTYPS,    NMAT,      LBUCK,                KCTYP,
      4   KSISO,     KGST,     KCTR,      KCGG,      KDREAD,    KFLUXI,
      5                        ITMIMX,    ITMIGM,    NZONER,    NZONEZ,
      6   KACCEL,    KALC1,    NFN,       N2N,                  ITMPRT,
      7   KBCDUP,    KACAV,    KGAVE,     KNIP,      KEDIT,
      8                        GLAM,      EIGEN,     EPS,       XNF,
      9   RYF,       TIMAX,    CALC1,                OMEGA,     OMEGAP
       COMMON        /CALL3/
      1                        HZ,        HY,        BF,
      2                                               KCA,       KCF,
      3   KONV,      KFINSH,   KASOR,     KUPS,
      4   NBF,       ND,       NGP1,      NZP1,      NRP1,
      5   NIJ,       NTYNG,    NDNZ,      NDNR,      NDNIJ,     NDNZNG,
      6   NXNZS,     NAR,      NC,        NIJG,      NBUCK,
      7   ITMA,      ITMI,     ITMIST,    ITMIGX,    ITD,
      8                        NEGSOR,               LZSAV,
      9   TIMEX,     TIMEZ,                          SDGFG1,    SCALUP
       COMMON        /CALL4/   EPG,       EPGX,      EPGM,      EPGMP,
      1                        OMEGAS,    OMEGPS,
      2              EXTRA1,   EXTRA2,    EXTRA3,    EXTRA4,    EXTRA5,
      3              KXTRA1,   KXTRA2,    KXTRA3,    KXTRA4,    KXTRA5
C
C      THE DIMENSION STATEMENTS
       DIMENSION     MA(1)
       DIMENSION     FG(1),          F(1)
       DIMENSION     XN(1),          XJR(1),          XJZ(1)
       DIMENSION     V(1)
       DIMENSION     KFING(1),       KASORG(1)
       DIMENSION     AG(1),          FNG(1),
      1              SG(1),          ASG(1),          SCG(1),
      2              SDG(1),         SING(1),         XNG(1),
      3              XJRG(1),        XJZG(1),         XNB(1),
      4              HNLG(1),        VNLG(1),         XNLG(1),
      5              BNLG(1)
```

```
      DIMENSION                    BUCLK(1),      MXVARI(1),
     1          ESIJM(1)
      DIMENSION    STG(1)
      DIMENSION    XNORM(1)
C
C     THE FORMAT STATEMENTS
  100 FORMAT (1H1)
  103 FORMAT (1H )
  111 FORMAT (8I14)
  113 FORMAT (8E14.6)
  120 FORMAT (126H ITMA ITMI ITMIGX ITD    TIME REQ          EPG
     1EPGM       KASOR KONV KFINSH    NEUTRON BAL          GLAM          E
     2IGEN     )
  121 FORMAT (I4,I6,I4,I7,3E14.6,I4,I5,I6,E18.8,2F16.8)
  122 FORMAT (1HK,50HTHE GROUP CONVERGENCE, KFINSH BY GROUP (KFING(IG)))
  123 FORMAT (1HL,28HTOTAL NEUTRON FLUX (XNG(IG)))
  124 FORMAT (1HL,23HFISSION SOURCE (FG(IG)))
  125 FORMAT (1HK,31HIN SCATTERING SOURCE (SING(IG)))
  126 FORMAT (1HK,29HOUT SCATTERING LOSS (SDG(IG)))
  127 FORMAT (1HK,32HSELF SCATTERING SOURCE (SCG(IG)))
  128 FORMAT (1HK,35HNON-SELF SCATTERING SOURCE (SG(IG)))
  129 FORMAT (1HK,22HTOTAL SOURCE (STG(IG)))
  130 FORMAT (1HL,26HFISSION NEUTRONS (FNG(IG)))
  131 FORMAT (1HK,20HABSORPTIONS (AG(IG)))
  132 FORMAT (1HK,29HHORIZONTAL LEAKAGE (HNLG(IG)))
  133 FORMAT (1HK,27HVERTICAL LEAKAGE (VNLG(IG)))
  134 FORMAT (1HK,22HNET LEAKAGE (XNLG(IG)))
  135 FORMAT (1HL,42HNEUTRON BALANCE (XNB(IG))  (SOURCE - LOSS))
  136 FORMAT (1HK,48HTOTAL FIRST DIRECTION NEUTRON CURRENT (XJRG(IG)))
  138 FORMAT (1HK,74HANISOTROPIC NON-SELF SCATTERING SOURCE WITHOUT DIRE
     1CTION COSINES (ASG(IG)))
  140 FORMAT (1HL,2X,40HTHE FISSION RATE PER UNIT VOLUME (F(IJ)))
  141 FORMAT (1HK,49HTOTAL SECOND DIRECTION NEUTRON CURRENT (XJZG(IG)))
  143 FORMAT (1H1,2X,66HTHE NET FIRST DIRECTION CURRENT PER UNIT VOLUME
     1BY GROUP (XJR(IJ)))
  144 FORMAT (1H1,2X,67HTHE NET SECOND DIRECTION CURRENT PER UNIT VOLUME
     1 BY GROUP (XJZ(IJ)))
  145 FORMAT (1H1,2X,58HTHE NON-DIRECTIONAL FLUX PER UNIT VOLUME BY GROU
     1P (XN(IJ)))
  146 FORMAT (1HK,2X,3HIG=,I3)
  147 FORMAT (1HK,82HTHE NUMBER OF MESH INTERVALS AT WHICH A P1 SOURCE C
     1ORRECTION WAS MADE (KASORG(IG)))
  148 FORMAT (1H1,51HNORMALIZATION FACTOR FOR F, XN, XJR, AND XJZ EQUALS
     1,E16.8)
  149 FORMAT (1HL,97HTHE MAXIMUM RATE OF CHANGE IN THE SELF-SCATTERING S
     1OURCE OCCURS AT THE MESH INTERVAL (MXVARI(IG)))
  150 FORMAT (1HK,25HBUCKLING LOSS (BUCLK(IG)))
  151 FORMAT (1HK,24HBUCKLING LOSS (BNLG(IG)))
  152 FORMAT (1HK,68HTHE MAXIMUM RATE OF CHANGE IN THE SELF-SCATTERING S
     1OURCE (ESIJM(IG)))
  165 FORMAT (6X,2HJ=,I3)
  170 FORMAT (1H$, 7F10.6)
  171 FORMAT (1H1, 14HFLUX INTEGRALS)
  172 FORMAT (1H$, 14HFLUX INTEGRALS)
  173 FORMAT (1HL,34HAVERAGE (D*BZS) OVER ENTIRE VOLUME)
  174 FORMAT (1HL,43HAVERAGE PARALLEL (D*BSQ) OVER ENTIRE VOLUME)
```

```
  175 FORMAT (1H$,43HAVERAGE PARALLEL (D*BSQ) OVER ENTIRE VOLUME)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 WRITE (6,100)
      WRITE (6,120)
      WRITE (6,121) ITMA,ITMI,ITMIGX,ITD,TIMEZ,TIMEX,EPGM,KASOR,KONV,
     1   KFINSH,XNB(NGP1),GLAM,EIGEN
      IF (KFINSH) 1010,1015,1010
 1010 WRITE (6,122)
      WRITE (6,111) (KFING(IG), IG=1,NG)
 1015 IF (KASOR) 1030,1030,1020
 1020 WRITE (6,147)
      WRITE (6,111) (KASORG(IG), IG=1,NG)
 1030 WRITE (6,149)
      WRITE (6,111) (MXVARI(IG), IG=1,NG)
      WRITE (6,152)
      WRITE (6,113) (ESIJM(IG), IG=1,NG)
C
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
 1052 DO 1056 IG=1,NG
      IF (KSISO) 1055,1055,1053
 1053 XJRG(IG)=0.0
      IF (NZ-1) 1058,1058,1057
 1057 XJZG(IG)=0.0
 1058 IJG=INDEX(NIJ,IG,0)
      DO 1054 IJ=1,NIJ
      IJG=IJG+1
      XJRG(IG)=XJRG(IG)+V(IJ)*XJR(IJG)
      IF (NZ-1) 1054,1054,1059
 1059 XJZG(IG)=XJZG(IG)+V(IJ)*XJZ(IJG)
 1054 CONTINUE
      ASGSUM=ASGSUM+ASG(IG)
      XJRGSM=XJRGSM+XJRG(IG)
      IF (NZ-1) 1055,1055,1061
 1061 XJZGSM=XJZGSM+XJZG(IG)
 1055 SG(NGP1)=SG(NGP1)+SG(IG)
 1056 FNGSUM=FNGSUM+FNG(IG)
      DO 1060 IG=1,NGP1
 1060 STG(IG)=SG(IG)+SCG(IG)
      IF (LBUCK) 1070,1070,1068
 1068 BUCLKS=0.0
      DO 1069 IG=1,NG
      BUCLK(IG)=BUCLK(IG)*XNG(IG)
 1069 BUCLKS=BUCLKS+BUCLK(IG)
C
```

```
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) (SING(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) (SG(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), ASGSUM

 1090 WRITE (6,130)
      WRITE (6,113) (FNG(IG), IG=1,NG), FNGSUM

      WRITE (6,131)
      WRITE (6,113) (AG(IG), IG=1,NGP1)
      WRITE (6,132)
      WRITE (6,113) (HNLG(IG), IG=1,NGP1)
      IF (NZ-1) 1092,1092,1091
 1091 WRITE (6,133)
      WRITE (6,113) (VNLG(IG), IG=1,NGP1)
 1092 IF (NBF-1) 1093,1093,1094
 1093 IF (NBUCK) 1095,1095,1094
 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
C
 1100 XNMAX=0.0
      DO 1105 IJG=1,NIJG
      IF (XNMAX-XN(IJG)) 1101,1105,1105
 1101 XNMAX=XN(IJG)
```

```
 1105 CONTINUE
      IF (XNMAX) 1106,1107,1106
 1106 YNORM=1.0/XNMAX
      GO TO 1110
 1107 YNORM=1.0
 1110 WRITE (6,148) YNORM
C
C
C
 1125 DO 1130 IJ=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
C
 1150 WRITE (6,145)
      IJG=0
      DO 1160 IG=1,NG
      DO 1155 IJ=1,NIJ
      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
C
      IF (KSISO) 1200,1200,1165
 1165 WRITE (6,143)
      IJG=0
      DO 1170 IG=1,NG
      DO 1166 IJ=1,NIJ
      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
C
      IF (NZ-1) 1200,1200,1171
 1171 WRITE (6,144)
```

```
      IJG=0
      DO 1173 IG=1,NG
      DO 1172 IJ=1,NIJ
      IJG=IJG+1
 1172 XNORM(IJ)=YNORM*XJZ(IJG)
      WRITE (6,146) IG
      DO 1173 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)
 1173 WRITE (6,103)
C
C
C
 1200 IF (KNIP) 1300,1300,1205
 1205 WRITE (6,171)
      WRITE (6,172)
      DO 1280 N=1,NMAT
      DO 1270 IG=1,NG
      STG(IG)=0.0
      IJG=INDEX(NIJ,IG,0)
      DO 1260 IJ=1,NIJ
      IJG=IJG+1
      IF (N-IABS(MA(IJ))) 1225,1245,1225
 1225 IF (MA(IJ)) 1230,1260,1260
 1230 IF (N+(MA(IJ)-1)) 1260,1235,1260
 1235 TEMP=V(IJ)*XJR(IJG)
      IF (NZ-1) 1250,1250,1240
 1240 TEMP=TEMP+V(IJ)*XJZ(IJG)
      GO TO 1250
 1245 TEMP=V(IJ)*XN(IJG)
 1250 STG(IG)=STG(IG)+TEMP
 1260 CONTINUE
 1270 CONTINUE
      WRITE (6,113) (STG(IG), IG=1,NG)
      WRITE (6,170) (STG(IG), IG=1,NG)
 1280 CONTINUE
C
      IF (KNIP-2) 1300,1285,1300
 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) 1305,1400,1305
 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) 1405,1401,1405
 1401 IF (KGAVE) 1410,1410,1405
 1405 KCHAIN=4
      RETURN
C
 1410 IF (KEDIT) 1420,1420,1415
 1415 KCHAIN=5
      RETURN
C
 1420 KCHAIN=1
      RETURN
C
      END
```

```
$IBFTC TABLES  DECK
      SUBROUTINE TABLES (MA,C,  CN2N,   XN,XJR,XJZ,   V,
     1   NMRA,NMZA,IDM,IDMAP,IDGP)
C
C     SUBROUTINE TABLES IS THE OVERLAY (OR CHAIN) SUBROUTINE OF TDSN
C        THAT COMPUTES AND PRINTS OUT THE ACTIVITY AND AVERAGING TABLES.
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCOUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   KYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
     1                        OMEGAS,   OMEGPS,
     2              EXTRA1,   EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
     3              KXTRA1,   KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
      COMMON        /CTABLE/  KTABLE,   NACAVT,   NTYBAR,   KGGBAR
      COMMON        /CSUM/    KSUM,     NACAV1,   NACAV2,   NZM1,
     1   NTYG1,     IG
      COMMON        /CAVER/   ID,       KF,       KC,       NACAV
C
C     THE DIMENSION STATEMENTS
      DIMENSION     MA(1),          C(1)
      DIMENSION     CN2N(1)
      DIMENSION     XN(1),          XJR(1),         XJZ(1)
      DIMENSION     V(1)
      DIMENSION     NMRA(1),        NMZA(1),        IDM(1),
     1              IDMAP(1),       IDGP(1)
      DIMENSION     XTAB(5000)
C
C     THE FORMAT STATEMENTS
  104 FORMAT (1HL,21HPROGRAM STOP AT KEFN=,I8)
  110 FORMAT (7I10)
```

```
  120 FORMAT (1HL,36HTABLES ARRAY SIZE EXCEEDED.  LASTAB=,I8,31H  WITH 5
     1000 STORAGES 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=KCGG-(KCTR+1)
      KDOWN=NTYPS-KCGG
      IF (KGAVE) 1100,1100,1010
 1010 READ (5,110) (IDGP(IG), IG=1,NG)
      KDBAR=KGAVE-1
      IF (KDOWN-KDBAR) 1015,1020,1020
 1015 KDBAR=KDOWN
 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) 1075,1045,1045
 1045 IF (IDGP(IG)-IGB) 1040,1040,1050
 1050 IGB=IGB+1
      GO TO 1030
 1075 NTYBAR=NTYBAR+KUPBAR
      KGGBAR=(KCTR+1)+KUPBAR
C
C
C     COMPUTE THE ACTIVITY TABLES.
C
C
 1100 IF (KACAV) 1120,1502,1105
 1105 IF (KACAV-2) 1110,1501,1110
 1110 WRITE (6,160)
      DO 1115 IJ=1,NIJ
 1115 IDMAP(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
```

```
C
 1150 LKSI=1
      LVSUM=LKSI+NACAVT
      LFX=LVSUM+NACAVT
      LCX=LFX+NACAVT*NGP1
      LCOUT=LCX+NACAVT*NGP1*NTYPS
      LDISAD=LCOUT+1
      LFXBAR=LDISAD+NG
      LCXBAR=LFXBAR+1
      LCOBAR=LCXBAR+1
      LASTAB=LCOBAR
      IF (LASTAB-5000) 1200,1200,1155
 1155 WRITE (6,120) LASTAB
      KEND=1
      KEFN=1155
      GO TO 2050
C
 1200 KTABLE=1
      CALL TABLE (MA,C,CN2N,  XN,XJR,XJZ, V,  IDMAP,IDGP,
     1   XTAB(LKSI),XTAB(LVSUM),  XTAB(LFX),XTAB(LCX),XTAB(LCOUT),
     2   XTAB(LDISAD),  XTAB(LFXBAR),XTAB(LCXBAR),XTAB(LCOBAR)  )
      IF (KEND) 1500,1500,2050
C
C
C     COMPUTE THE AVERAGING TABLES.
C
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 IJ=1,NIJ
 1515 IDMAP(IJ)=MA(IJ)
      NACAV=NMAT
      GO TO 1530
 1520 CALL IDACAV (NR,NZ,NACAV,NZONRA,NZONZA,   NMRA(1),NMZA(1),IDM(1),
     1   IDMAP(1))
C
 1530 NACAV1=NACAV+1
      NACAVT=NACAV1
      IF (KSISO) 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
      LDISAD=LCOUT+NACAVT*NGP1
      LFXBAR=LDISAD+1
      LCXBAR=LFXBAR+NTYBAR*KGAVE
      LCOBAR=LCXBAR+NTYBAR*KGAVE
```

```
      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   XTAB(LKSI),XTAB(LVSUM),  XTAB(LFX),XTAB(LCX),XTAB(LCOUT),
     2   XTAB(LDISAD),  XTAB(LFXBAR),XTAB(LCXBAR),XTAB(LCOBAR)  )
      IF (KEND) 2000,2000,2050
C
C
C
 2000 KCHAIN=5
      RETURN
C
 2050 KCHAIN=1
      WRITE (6,104) KEFN
      RETURN
C
      END
```

```
$IBFTC IDACAV  DECK
      SUBROUTINE IDACAV (NR,NZ,NACAV,NZONRA,NZONZA,   NMRA,NMZA,IDM,
     1   IDMAP)
C
C     SUBROUTINE IDACAV OBTAINS THE IDENIFICATION NUMBER MAP TO USE FOR
C        ACTIVITY TABLES AND AVERAGING TABLES.
C
C
C
C
C
C
C
C     THE DIMENSION STATEMENTS
      DIMENSION    NMRA(1),       NMZA(1),       IDM(1)
      DIMENSION    IDMAP(1)
C
C     THE FORMAT STATEMENTS
  110 FORMAT (7I10)
  115 FORMAT (14I5)
  116 FORMAT (43I3)
  120 FORMAT (1HK,15HMAP FOR TABLES      )
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
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(NR,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)
      DO 1080 J=1,NZ
      I1=INDEX(NR,J,1)
      INR=INDEX(NR,J,NR)
 1080 WRITE (6,116) (IDMAP(IJ), IJ=I1,INR)
C
C
C

 1100 RETURN
C
      END
```

```
$IBFTC TABLE   DECK
      SUBROUTINE TABLE (MA,C,  CN2N,    XN,XJR,XJZ,   V,
     1                  IDMAP,IDGP,     KSI,VSUM,   FX,CX,COUT,   DISAD,
     2   FXBAR,CXBAR,COBAR   )
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCDUP,    KACAV,    KGAVE,    KNIP,     KELIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
     1                        OMEGAS,   OMEGPS,
     2                EXTRA1,   EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
     3                KXTRA1,   KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
      COMMON        /CTABLE/  KTABLE,   NACAVT,   NTYBAR,   KGGBAR
      COMMON        /CSUM/    KSUM,     NACAV1,   NACAV2,   NZM1,
     1   NTYG1,     IG
      COMMON        /CAVER/   ID,       KF,       KC,       NACAV
C
C     THE DIMENSION STATEMENTS
      DIMENSION     MA(1),          C(1)
      DIMENSION     CN2N(1)
      DIMENSION     XN(1),          XJR(1),         XJZ(1)
      DIMENSION     V(1)
      DIMENSION     IDMAP(1),       IDGP(1)
      DIMENSION     KSI(1),         VSUM(1)
      DIMENSION     FX(1),          CX(1),          COUT(1)
      DIMENSION     DISAD(1)
      DIMENSION     FXBAR(1),       CXBAR(1),       COBAR(1)
C
C     THE FORMAT STATEMENTS
  104 FORMAT (1HL,21HPROGRAM STOP AT KEFN=,I8)
  111 FORMAT (7I16)
```

```
  113 FORMAT (7E16.8)
  162 FORMAT (1H$,17HAVERAGING TABLES     )
  163 FORMAT (1H1,20HDISADVANTAGE FACTORS      )
  164 FORMAT (1HL)
  165 FORMAT (1HL,32HZONE IDENTIFICATION NUMBER (ID)=,I3)
  166 FORMAT (1H1,30HGROUP COLLAPSED CROSS SECTIONS)
  167 FORMAT (1HL, 33HI.D. NUMBERS FOR COLLAPSED GROUPS    )
  168 FORMAT (1H$,30HGROUP COLLAPSED CROSS SECTIONS)
  169 FORMAT (1H$,7F10.6)
  170 FORMAT (1H )
  171 FORMAT (1HL,28HTOTAL OF P(0) AND P(T) 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 WIT
     1HIN GROUP SCATTERING CROSS SECTION WHICH HAS BEEN COMPUTED BY NEUT
     2RON BALANCE ,/5X,92HFROM SCATTERING THAT CONTAINS THE EXTRA NEUTRO
     3NS AND THE TOTAL COLLISION LOSS WHICH DOES NOT)
C
C      THE INTEGER FUNCTIONS
       INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 GO TO (1150,1550), KTABLE
C
C
C      COMPUTE THE ACTIVITY TABLES.
C
C
 1150 KF=0
       KC=0
       DO 1165 ID=1,NACAVT
       DO 1160 IG=1,NGP1
       KF=KF+1
       DO 1155 N=1,NTYPS
       KC=KC+1
 1155 CX(KC)=0.0
 1160 FX(KF)=0.0
       KSI(ID)=0
 1165 VSUM(ID)=0.0
       DO 1170 IJ=1,NIJ
 1170 VSUM(NACAV1)=VSUM(NACAV1)+V(IJ)
       IF (KSISO) 1200,1200,1175
 1175 VSUM(NACAV2)=VSUM(NACAV1)
       KSI(NACAV2)=1
C
C
 1200 DO 1250 IG=1,NG
C
 1225 KSUM=1
       CALL SUM (MA,C, CN2N, XN,XJR,XJZ, V,  IDMAP, KSI,VSUM,
      1   FX,CX,COUT  )
       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+NTYPS
      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 KF1=INDEX(NGP1,NACAV1,0)
      GO TO 1425
 1410 KF1=INDEX(NGP1,NACAV2,0)
 1425 DO 1450 IG=1,NG
      KF1=KF1+1
      KF=KF+1
      DISAD(IG)=FX(KF)/VSUM(ID)
      IF (KSI(ID)) 1430,1430,1435
 1430 DISAD(IG)=DISAD(IG)/(FX(KF1)/VSUM(NACAV1))
      GO TO 1450
 1435 DISAD(IG)=DISAD(IG)/(FX(KF1)/VSUM(NACAV2))
 1450 CONTINUE
      KF=KF+1
C
      WRITE (6,113) (DISAD(IG), IG=1,NG)
 1475 WRITE (6,170)
C
      RETURN
C
C
C     COMPUTE THE AVERAGING TABLES.
C
C
 1550 KF=0
      KC=0
      DO 1565 ID=1,NACAVT
      DO 1560 IG=1,NGP1
```

```
      KF=KF+1
      DO 1555 N=1,NTYPS
      KC=KC+1
 1555 CX(KC)=0.C
      COUT(KF)=0.0
 1560 FX(KF)=0.0
 1565 KSI(ID)=0
      IF (KSISO) 1600,1600,1570
 1570 KSI(NACAV2)=1
C
C
 1600 DO 1650 IG=1,NG
C
 1625 KSUM=2
      CALL SUM (MA,C, CN2N, XN,XJR,XJZ, V,  IDMAP, KSI,VSUM,
     1   FX,CX,COUT  )
      IF (KEND) 1650,1650,2050
 1650 CONTINUE
C
C
 1700 KC=0
      DO 1750 ID=1,NACAVT
      IF (IABS(KACAV)-1) 1740,1740,1705
 1705 IF (ID-NACAV1) 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
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 KF=0
      KC=0
      DO 1875 ID=1,NACAVT
      IF (ID-NACAV1) 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 1860 IG=1,KGAVE
```

```
      K1=INDEX(NTYBAR,IG,1)
      K2=INDEX(NTYBAR,IG,NTYBAR)
      WRITE (6,113) (CXBAR(K), K=K1,K2), FXBAR(K1)

      WRITE (6,170)
 1860 WRITE (6,169) (CXBAR(K), K=K1,K2)
C

 1875 CONTINUE
C
C
C
 2000 RETURN
C
C
 2050 WRITE (6,104) KEFN
      KEFN=2050
      RETURN
C
      END
```

```
$IBFTC SUM      DECK
      SUBROUTINE SUM (MA,C,  CN2N,  XN,XJR,XJZ,  V,  IDMAP,  KSI,  VSUM,
     1   FX,CX,COUT   )
C
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NK,        NTYPS,    NMAT,     LBUCK,              KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCOUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   KYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CSUM/    KSUM,     NACAV1,   NACAV2,   NZM1,
     1   NTYG1,     IG
C
C     THE DIMENSION STATEMENTS
      DIMENSION     MA(1),          C(1)
      DIMENSION     CN2N(1)
      DIMENSION     XN(1),          XJR(1),         XJZ(1)
      DIMENSION     V(1)
      DIMENSION     IDMAP(1)
      DIMENSION     KSI(1),         VSUM(1)
      DIMENSION     FX(1),          CX(1),          COUT(1)
C
C     THE FORMAT STATEMENTS
  104 FORMAT (1HL,21HPROGRAM STOP AT KEFN=,I8)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 IJG=INDEX(NIJ,IG,0)
      DO 1200 IJ=1,NIJ
      IJG=IJG+1
      ID=IABS(IDMAP(IJ))
```

```
      IDA=ID
      K=IABS(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)+TEMP
 1020 FX(KF)=FX(KF)+TEMP1
      KF=INDEX(NGP1,ID,NGP1)
      GO TO (1030,1025), KSUM
 1025 COUT(KF)=COUT(KF)+TEMP
 1030 FX(KF)=FX(KF)+TEMP1
      KF=INDEX(NGP1,NACAV1,NGP1)
      GO TO (1040,1035), KSUM
 1035 COUT(KF)=COUT(KF)+TEMP
 1040 FX(KF)=FX(KF)+TEMP1
      IF (IDMAP(IJ)) 1045,1075,1075
 1045 IDA=IDA+1
      IF (MA(IJ)) 1055,1050,1050
 1050 KEND=1
      KEFN=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,IDA,IG)
      FX(KF)=FX(KF)+TEMP2
      KF=INDEX(NGP1,NACAV2,IG)
      FX(KF)=FX(KF)+TEMP2
      KF=INDEX(NGP1,IDA,NGP1)
      FX(KF)=FX(KF)+TEMP2
      KF=INDEX(NGP1,NACAV2,NGP1)
      FX(KF)=FX(KF)+TEMP2
C
 1075 DO 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-KCGG+N
      IF (IGN) 1125,1125,1090
 1090 IF (IGN-NG) 1100,1100,1125
 1100 K1=INDEX(NTYPS,IGN,N)
      K2=INDEX(NTYNG,K,K1)
      KC=INDEX(NTYG1,ID,K1)
      KCS=INDEX(NTYG1,NACAV1,K1)
      TEMP=C(K2)*TEMP1
      CX(KC)=CX(KC)+TEMP
      CX(KCS)=CX(KCS)+TEMP
```

```
      IF (KREG) 1101,1101,1102
 1101 IF (IG-IGN) 1110,1105,1110
 1102 IF (N-KCTR) 1105,1105,1103
 1103 IF (N-KCGG) 1110,1105,1110
 1105 KT=INDEX(NTYPS,NGP1,N)
      KC=INDEX(NTYG1,ID,KT)
      KCS=INDEX(NTYG1,NACAV1,KT)
      CX(KC)=CX(KC)+TEMP
      CX(KCS)=CX(KCS)+TEMP
 1110 IF (ID-IDA) 1115,1125,1125
 1115 K2=INDEX(NTYNG,KA,K1)
      KC=INDEX(NTYG1,IDA,K1)
      KCS=INDEX(NTYG1,NACAV2,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-KCTR) 1120,1120,1118
 1118 IF (N-KCGG) 1125,1120,1125
 1120 KC=INDEX(NTYG1,IDA,KT)
      KCS=INDEX(NTYG1,NACAV2,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 (1175,1200), KSUM
 1165 KSI(IDA)=1
      GO TO (1170,1200), KSUM
 1170 VSUM(IDA)=VSUM(IDA)+V(IJ)
 1175 VSUM(ID)=VSUM(ID)+V(IJ)
C
 1200 CONTINUE
C
C
 1250 RETURN
C
 1275 WRITE (6,104) KEFN
      RETURN
C
      END
```

```
$IBFTC AVERAG  DECK
      SUBROUTINE AVERAG ( KSI,  FX,CX,COUT  )
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,               KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCDUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CAVER/   ID,       KF,       KC,       NACAV
C
C     THE DIMENSION STATEMENTS
      DIMENSION     KSI(1)
      DIMENSION     FX(1),          CX(1),          COUT(1)
C
C     THE FORMAT STATEMENTS
  113 FORMAT (7E16.8)
  169 FORMAT (1H$,7F10.6)
  170 FORMAT (1H )
  173 FORMAT (1HJ,22HCOLLAPSED TO ONE GROUP)
  174 FORMAT (24H OUT SCATTERING BY GROUP)
  175 FORMAT (1H$,24H OUT SCATTERING BY GROUP)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 DO 1125 IG=1,NG
      K1=KC+1
      K2=KC+NTYPS
      DO 1100 N=1,NTYPS
      KC=KC+1
      IF (N-KCTR) 1005,1005,1010
```

```
 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-KCGG) 1050,1075,1050
 1050 CX(KC)=CX(KC)/FX(KF)
      GO TO 1100
 1075 CX(KC)=CX(KC1)-CX(KC1-2)-COUT(KF)
 1100 CONTINUE
C
      IF (IABS(KACAV)-1) 1125,1125,1105
 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
C
 1150 KF=INDEX(NGP1,ID,NGP1)
      K1=KC+1
      K2=KC+NTYPS
      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
 1160 KC=KC+1
      IF (KSI(ID)) 1170,1170,1165
 1165 CX(KC)=CX(KC)/FX(KF)
      GO TO 1175
 1170 CX(KC)=CX(KC1)-CX(KC1-2)
 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)(COUT(KF), KF=KF1,KF2)
      IF (ID-NACAV) 1190,1190,1195
 1190 IF (KACAV) 1195,1200,1200
 1195 WRITE (6,175)

      WRITE (6,169) (COUT(KF), KF=KF1,KF2)
C
C
C
 1200 RETURN
C
      END
```

```
$IBFTC COLAPS  DECK
      SUBROUTINE COLAPS ( KOLAPS,NTYLAP,KGGLAP,   KIG,   KSI,
     1   FS,CS,             FOLAP,COLAP,OUTLAP  )
C
C     SUBROUTINE COLAPS IS TO GIVE COLLAPSED GROUP CROSS SECTIONS.
C        (THIS VERSION MUST BE CALLED BY TABLES.)
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCDUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CAVER/   ID,       KF,       KC,       NACAV
C
C     THE DIMENSION STATEMENTS
      DIMENSION     KIG(1)
      DIMENSION     KSI(1)
      DIMENSION     FS(1),          CS(1)
      DIMENSION     FOLAP(1),       COLAP(1),       OUTLAP(1)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 K=0
      DO 1010 IG=1,KOLAPS
      DO 1005 N=1,NTYLAP
      K=K+1
      COLAP(K)=0.0
 1005 FOLAP(K)=0.0
 1010 OUTLAP(IG)=0.0
C
```

```
 1025 DO 1060 IG=1,NG
      KF=KF+1
      IGK=KIG(IG)
      K=INDEX(NTYLAP,IGK,0)
      DO 1055 N=1,NTYPS
      KC=KC+1
      IF (N-KCTK) 1030,1030,1035
 1030 K=K+1
      FOLAP(K)=FOLAP(K)+FS(KF)
      L=KF
      GO TO 1050
 1035 IGN=IG+KCGG-N
      IF (IGN) 1055,1055,1040
 1040 IF (IGN-NG) 1045,1045,1055
 1045 IGNLAP=KIG(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)*FS(L)
 1055 CONTINUE
 1060 CONTINUE
      KF=KF+1
      KC=KC+NTYPS
C
 1100 KCTR1=KCTR+1
      DO 1115 IG=1,KOLAPS
      DO 1115 N=KCTR1,NTYLAP
      IGN=IG+KGGLAP-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 IG=1,KOLAPS
      DO 1180 N=1,NTYLAP
      IF (N-KCTR) 1155,1155,1160
 1155 IGN=IG
      GO TO 1170
 1160 IGN=IG-N+KGGLAP
      IF (IGN) 1180,1180,1165
 1165 IF (IGN-KOLAPS) 1170,1170,1180
 1170 K=INDEX(NTYLAP,IG,N)
      COLAP(K)=COLAP(K)/FOLAP(K)
      IF (IGN-IG) 1175,1180,1175
 1175 OUTLAP(IGN)=OUTLAP(IGN)+COLAP(K)
 1180 CONTINUE
      IF (KSI(ID)) 1185,1185,1200
 1185 DO 1190 IG=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
```

```
$IBFTC EDIT    DECK
      SUBROUTINE EDIT (MA,C,XN,V)
C
C     SUBROUTINE EDIT COMPUTES ACTIVITIES, FLUX INTEGRALS, AND AVERAGE
C        ACTIVITIES FOR A SPECIFIED CROSS SECTION OVER GIVEN SETS OF
C        GROUPS AND REACTOR REGIONS.
C
C
C
C
C
C
C
C     THE COMMON STATEMENTS
      COMMON        /CALL1/   KCHAIN,   KEND,     KEFN
      COMMON        /CALL2/   KBCRED,
     1   KREG,      KALC,     KGEO,     KBBC,     KTBC,     KLBC,
     2   KRBC,                          NSN,      NG,       NZ,
     3   NR,        NTYPS,    NMAT,     LBUCK,              KCTYP,
     4   KSISO,     KGST,     KCTR,     KCGG,     KDREAD,   KFLUXI,
     5                        ITMIMX,   ITMIGM,   NZONER,   NZONEZ,
     6   KACCEL,    KALC1,    NFN,      N2N,                ITMPRT,
     7   KBCDUP,    KACAV,    KGAVE,    KNIP,     KEDIT,
     8                        GLAM,     EIGEN,    EPS,      XNF,
     9   RYF,       TIMAX,    CALC1,              OMEGA,    OMEGAP
      COMMON        /CALL3/
     1                        HZ,       HY,       BF,
     2                                            KCA,      KCF,
     3   KONV,      KFINSH,   KASOR,    KUPS,
     4   NBF,       ND,       NGP1,     NZP1,     NRP1,
     5   NIJ,       NTYNG,    NDNZ,     NDNR,     NDNIJ,    NDNZNG,
     6   NXNZS,     NAR,      NC,       NIJG,     NBUCK,
     7   ITMA,      ITMI,     ITMIST,   ITMIGX,   ITD,
     8                        NEGSOR,             LZSAV,
     9   TIMEX,     TIMEZ,                        SDGFG1,   SCALUP
      COMMON        /CALL4/   EPG,      EPGX,     EPGM,     EPGMP,
     1                        OMEGAS,   OMEGPS,
     2                EXTRA1,   EXTRA2,   EXTRA3,   EXTRA4,   EXTRA5,
     3                KXTRA1,   KXTRA2,   KXTRA3,   KXTRA4,   KXTRA5
C
C     THE DIMENSION STATEMENTS
      DIMENSION     MA(1),          C(1),           XN(1),
     1              V(1)
      DIMENSION     ACTINT(201),    FINT(201),      PAVE(201)
C
C     THE FORMAT STATEMENTS
  100 FORMAT (1H1)
  101 FORMAT (1HL)
  110 FORMAT (7I10)
  112 FORMAT (7F10.6)
  113 FORMAT (7E16.8)
  200 FORMAT (1HL,9HEDIT SET ,I2)
  201 FORMAT (1HL,10HINTEGRALS ,I3,18H HAVE LIMITS FROM ,I2,4H TO ,I2,
     1   12H ON GROUPS, ,I2,4H TO ,I2,31H ON RADIAL MESH INTERVALS, AND
     2   ,I2,4H TO ,I2,24H ON AXIAL MESH INTERVALS)
  202 FORMAT (1HL,12HSIGMA EQUALS,E16.8,14H  FOR OPTION 1)
```

```
  203 FORMAT (1HL,32HACTIVITY INTEGRALS FOR EDIT SET ,I3)
  204 FORMAT (1HL,28HFLUX INTEGRALS FOR EDIT SET ,I3)
  205 FORMAT (1HL,32HAVERAGE ACTIVITIES FOR EDIT SET ,I3)
C
C     THE INTEGER FUNCTIONS
      INDEX(LENGTH,INDEXO,INDEXL)=LENGTH*(INDEXO-1)+INDEXL
C
C
C
 1000 WRITE (6,100)
      READ (5,110) NISET
      DO 1175 ISET=1,NISET
      WRITE (6,200) ISET
      READ (5,110) INT
      INTP1=INT+1
      ACTINT(INTP1)=0.0
      FINT(INTP1)=0.0
C
      DO 1150 IN=1,INT
      READ (5,110) 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 IG=IG1,IG2
      GO TO (1025,1050,1055,1060), KEDIT
 1025 READ (5,112) SIGMA
      WRITE (6,202) SIGMA
      GO TO 1075
 1050 K1=INDEX(NTYPS,IG,KCA)
      GO TO 1075
 1055 K1=INDEX(NTYPS,IG,KCF)
      GO TO 1075
 1060 K1=INDEX(NTYPS,IG,KCTR)
 1075 DO 1125 J=IZ1,IZ2
      DO 1125 I=IR1,IR2
      IJ=INDEX(NR,J,I)
      IJG=INDEX(NIJ,IG,IJ)
      GO TO (1110,1100,1100,1100), KEDIT
 1100 K=IABS(MA(IJ))
      K2=INDEX(NTYNG,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
C
      ACTINT(INTP1)=ACTINT(INTP1)+ACTINT(IN)
      FINT(INTP1)=FINT(INTP1)+FINT(IN)
 1150 PAVE(IN)=ACTINT(IN)/FINT(IN)
      PAVE(INTP1)=ACTINT(INTP1)/FINT(INTP1)
C
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) (PAVE(IN), IN=1,INTP1)
      WRITE (6,101)
 1175 CONTINUE
C
C
C
      RETURN
C
      END
```

APPENDIX E

SAMPLE PROBLEM

```
CAR_SON TEST PROBLEM        SLAB        C=1.1                                   TITLE
3 GROUPS        S(8)        32 INTERVALS                                        TITLE
1  2-24-66                                                                      TITLE
         0                                                                      CONTROL
         0         1         1         1         0         0         0         CONTROL
         0         3        32         0         0         0         0         CONTROL
         0         0         1         4         7         0         0         CONTROL
         0         7         2         0         0         0         0         CONTROL
  0.000000  0.00000  0.00010  1.00000  1.00000  1.50000  0.00000               CONTROL
  1.200000  1.05000                              -2         1                  CONTROL
         3         2         3         1         1         4                  OUTPUT
    16  1.056700    1                                                         MESH
    16  2.113400    2                                                         MESH
.333333333.333333333.333333333                                               CHIS
   0.50000   0.00000  0.100000  1.000000.333333333.333333333.333333333       X-S 1-1
  0.000000  0.00000                                                           X-S 1-1
   1.00000   0.00000  0.100000  1.000000  0.333333.333333333.333333333       X-S 2-1
.333333333  0.00000                                                           X-S 2-1
   1.50000   0.00000  0.100000  1.000000  0.333333  0.333333.333333333       X-S 3-1
.333333333.333333333                                                         X-S 3-1
   0.50000   0.00000  0.100000  1.000000  0.333333  0.333333  0.533333       X-S 1-2
  0.000000  0.00000                                                           X-S 1-2
   1.00000   0.00000  0.100000  1.000000  0.333333  0.533333  0.533333       X-S 2-2
  0.500000  0.00000                                                           X-S 2-2
   1.50000   0.00000  0.100000  1.000000  0.333333  0.333333  0.533333       X-S 3-2
  0.500000  0.00000                                                           X-S 3-2
         1         1         1                                               CALL ID
         1                                                                   EDIT
         1                                                                   EDIT
         2         3         5        32         1         1                EDIT
```

LISTING OF SAMPLE PROBLEM OUTPUT

OVERLAY ORIGIN CARDS AND ASSIGNED LINK NUMBERS

```
$ORIGIN        TDSN00        IS LINK   1, PARENT LINK IS   0
$ORIGIN        TDSN00        IS LINK   2, PARENT LINK IS   0
$ORIGIN        TDSN21        IS LINK   3, PARENT LINK IS   2
$ORIGIN        TDSN22        IS LINK   4, PARENT LINK IS   3
$ORIGIN        TDSN00        IS LINK   5, PARENT LINK IS   0
$ORIGIN        TDSN00        IS LINK   6, PARENT LINK IS   0
$ORIGIN        TDSN61        IS LINK   7, PARENT LINK IS   6
$ORIGIN        TDSN61        IS LINK   8, PARENT LINK IS   6
$ORIGIN        TDSN62        IS LINK   9, PARENT LINK IS   8
$ORIGIN        TDSN62        IS LINK  10, PARENT LINK IS   8
$ORIGIN        TDSN62        IS LINK  11, PARENT LINK IS   8
$ORIGIN        TDSN00        IS LINK  12, PARENT LINK IS   0
```

```
                                M E M O R Y   M A P

SYSTEM                                                          00000 THRU  02717

FILE BLOCK ORIGIN                                               02720

     NUMBER OF FILES -   4

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

FILE LIST ORIGIN                                                03000

PRE-EXECUTION INITIALIZATION                                    03010

CALL ON OBJECT PROGRAM                                          03041

OBJECT PROGRAM                                                  03046 THRU  34262

     1.  DECK 'TDSN   '  *          03046      LINK NO.   0

         OVERLAY COMMUNICATION      05060                     0
     2.  DECK 'INPUT  '  *          15037                     1
     3.  DECK 'READSV'   *          20601                     1
     4.  DECK 'XSTDSN'   *          25347                     1
     5.  DECK 'SETUP  '  *          26445                     1
     6.  DECK 'BCREAD'   *          30617                     1
     7.  DECK '.READ5'   *          30731                     1
     8.  DECK 'OUTER  '  *          15037                     2
     9.  DECK 'ORSCAL'   *          21314                     2
    10.  DECK 'FISION'   *          21375                     3
    11.  DECK 'FIXED  '  *          22413                     3
    12.  DECK 'INNER  '  *          23627                     4
    13.  DECK 'FLUX   '  *          31327                     4
    14.  DECK 'SORAXS'   *          33547                     4
    15.  DECK 'DUMPBC'   *          15037                     5
    16.  DECK 'BCDUMP'   *          15464                     5
    17.  DECK '.PCH.  '  *          15672                     5
    18.  DECK 'OUTPUT'   *          15673                     5
    19.  DECK 'TABLES'   *          15037                     6
    20.  DECK 'IDACAV'   *          30022                     7
    21.  DECK 'TABLE  '  *          30022                     8
    22.  DECK 'SUM    '  *          31661                     9
    23.  DECK 'AVERAG'   *          31661                    10
    24.  DECK 'COLAPS'   *          31661                    11
    25.  DECK 'EDIT   '  *          15037                    12
    26.  SUBR '.IBSYS'              00000                     0
    27.  SUBR '.LRCOM'              00000                     0
    28.  SUBR '.IOEX '              00702                     0
    29.  SUBR '.JBCOM'              02652                     0
    30.  SUBR '.LXCOM'              05161                     0
    31.  SUBR '.IODEF'              05515                     0
    32.  SUBR '.IOCSF'              05734                     0
```

```
   33.  SUBR '.LOVRY'  *          10227          0
   34.  SUBR '.LXSL  '             10606          0
   35.  SUBR '.FPTRP'              10745          0
   36.  SUBR 'FOUT  '              11413          0
   37.  SUBR 'FCNV  '              11461          0
   38.  SUBR 'FIOS  '              13203          0
   39.  SUBR 'FIOH  '              13366          0
   40.  SUBR 'FSEL  '              14331          0
   41.  SUBR 'FWRD  '              14416          0
   42.  SUBR 'FRDD  '              14445          0
   43.  SUBR 'UNOS  '              14477          0
   44.  SUBR 'UNO6  '              14500          0
   45.  SUBR 'FSCN  '              14503          0
   46.  SUBR 'FSQR  '              14657          0
   47.  SUBR '.IOE. '              14723          0
   48.  SUBR '.IOE56'              14744          0
   49.  SUBR '.RWDOE'              14760          0
   50.  SUBR '.SNTL.'              15000          0
   51.  SUBR '.SQRTN'  *           15004          0
   52.  SUBR 'TIME1 '  *           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.
```

```
CARLSON TEST PROBLEM      SLAB      C=1.1
3 GROUPS      S(8)      32 INTERVALS
2-24-66


     KBCRED= 0

        KREG          KALC          KGEO          KLBC          KRBC          KBBC          KTBC
           0             1             1             1             0             0             0

         NSN            NG            NR            NZ          NMAT         NTYPS           N2N
           8             3            32             0             2             9             0

       KCTYP         KSISO          KGST          KCTR          KCGG         NBUCK         LBUCK
           0             0             1             4             7             0             0

      ITMIMX        ITMIGM        NZONER        NZONEZ        KCREAD        KFLUXI        KALC1
        4000             7             2             0             0             0             0

        GLAM          EIGEN           EPS           XNF           RYF          TIMAX         CALC1
0.              0.            0.99999999E-04 1.00000000E 00 1.00000000E 00 0.15000000E 01 0.

       OMEGA         OMEGAP                                                     KACCEL          KUPS
0.12000000E 01 0.10500000E 01                                                      -2             1

      ITMPRT         KBCDUP         KACAV         KGAVE          KNIP         KEDIT
           3             2             3             1             1             4

        LAST          LAST1         LAST2         LAST3         LAST4
         854           619           854           809           744

      NMI(I)        RMI(I)        MID(I)
          16 0.10567000E 01             1
          16 0.21134000E 01             2

THE FIRST DIRECTION MESH CELL BOUNDARY POSITIONS (R(NR))
 0.             0.66043749E-01 0.13208750E-00 0.19813125E-00 0.26417500E-00 0.33021875E-00 0.39626250E-00
 0.46230625E-00 0.52835000E 00 0.59439375E 00 0.66043749E 00 0.72648124E 00 0.79252499E 00 0.85856874E 00
 0.92461249E 00 0.99065624E 00 0.10567000E 01 0.11227437E 01 0.11887874E 01 0.12548312E 01 0.13208749E 01
 0.13869187E 01 0.14529625E 01 0.15190062E 01 0.15850499E 01 0.16510937E 01 0.17171375E 01 0.17831812E 01
 0.18492249E 01 0.19152687E 01 0.19813124E 01 0.20473561E 01 0.21133999E 01

THE MATERIAL MAP (MA(NZ,NR))
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2

THE FISSION SPECTRUM (XKI(NG))
 0.33333333E-00  0.33333333E-00  0.33333333E-00
```

```
THE CROSS SECTIONS (C(NMAT,NG,NTYPS))
     N=    1
 0.50000000E 00  0.              0.99999999E-01  1.00000000E 00  0.33333333E-00  0.33333333E-00  0.33333333E-00
 0.              0.
 1.00000000E 00  0.              0.99999999E-01  1.00000000E 00  0.              0.33333333E-00  0.33333333E-00
 0.33333333E-00  0.
 0.15000000E 01  0.              0.99999999E-01  1.00000000E 00  0.              0.              0.33333333E-00
 0.33333333E-00  0.33333333E-00

     N=    2
 0.50000000E 00  0.              0.99999999E-01  1.00000000E 00  0.              0.              0.50000000E 00
 0.              0.
 1.00000000E 00  0.              0.99999999E-01  1.00000000E 00  0.              0.50000000E 00  0.50000000E 00
 0.50000000E 00  0.
 0.15000000E 01  0.              0.99999999E-01  1.00000000E 00  0.              0.              0.50000000E 00
 0.50000000E 00  0.


THE OUT SCATTERING REMOVAL CROSS SECTIONS (CN2N(NMAT,NG))
     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 (V(NZ,NR))
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  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  0.66043735E-01  0.66043735E-01


  DIRECTIONAL FUNCTIONS
-1.00000000E 00 -0.88191709E 00 -0.62994078E 00 -0.37796447E-00 -0.12598816E-00  0.12598816E-00  0.37796447E-00
 0.62994078E 00  0.88191709E 00
 0.              0.12500000E-00  0.12500000E-00  0.12500000E-00  0.12500000E-00  0.12500000E-00  0.12500000E-00
 0.12500000E-00  0.12500000E-00
              9               9               8               7               6               5               4
              3               2
```

ITMA	ITMI	ITMIOX	ITO	TIME REQ	EPS	EPSM	KASER KCNV KFINSH	NEUTRON BAL	ALAM	EIGEN
			JOS2	XNG(NGP1)	SING(NGP1)	SDG(NGP1)	AG(NGP1)	HNLG(NGP1)	VNLG(NGP1)	XNLG(NGP1)
GROUP		ITMIG		FLUX ERROR	ERROR RATIO	FORMER RATIO	FACTOR		OMEGA	
		1		5	0.3746975E-02	0.2454362E-00	0.2420339E-00	0.74c9577E 00	0.1202922E 01	
		2		4	0.1974381E-02	0.2054584E-00	0.1895131E-00	0.	0.1200000E 01	
		2		5	0.1338523E-03	0.6779456E-01	0.2054584E-00	0.	0.1200000E 01	
		2		6	0.1130271E-04	0.8444170E-01	0.6779456E-01	0.	0.1200000E 01	
		3		4	0.4328935E-02	0.2284990E-00	0.2196293E-00	0.	0.1200000E 01	
		3		5	0.3504639E-03	0.8095645E-01	0.2284990E-00	0.	0.1200000E 01	
		3		6	0.3034210E-04	0.8657712E-01	0.8095645E-01	0.	0.1200000E 01	
1	19	7	19	0.888869E-01	0.666667E-03	0.106742E-04	0 0 0	-0.25531974E-06	0.86748671	0.86748672
				0.257358E-00	0.874414E 01	0.522601E 01	0.521550E 01 0.	0.101050E 01	0.	0.101050E 01
		1		3	0.1094316E-02	0.2165382E-00	0.2508727E-01	0.	0.1200000E 01	
		1		4	0.8773677E-04	0.8017501E-01	0.2165382E-00	0.	0.1200000E 01	
		1		5	0.7922726E-05	0.9030109E-01	0.8017501E-01	0.	0.1200000E 01	
		2		4	0.2065343E-02	0.2355412E-00	0.2326814E-00	0.7476501E 00	0.1202019E 01	
		3		3	0.5359903E-02	0.2370012E-00	0.2106269E-00	0.	0.1200000E 01	
		3		4	0.4597277E-03	0.8577164E-01	0.2370012E-00	0.	0.1200000E 01	
		3		5	0.4006564E-04	0.8715123E-01	0.8577164E-01	0.	0.1200000E 01	
		3		6	0.3489719E-05	0.8709963E-01	0.8715123E-01	0.	0.1200000E 01	
2	36	6	17	0.133611E-00	0.333333E-03	0.299429E-05	0 0 0	0.14901161E-05	1.08129461	0.93800873
				0.116378E-00	0.884054E 01	0.523144E 01	0.530650E 01 0.	0.924943E 00	0.	0.924943E 00
		1		3	0.1421382E-02	0.2337873E-00	0.3862995E-01	0.	0.1200000E 01	
		1		4	0.1347744E-03	0.9481926E-01	0.2337873E-00	0.	0.1200000E 01	
		1		5	0.1360312E-04	0.1009325E-00	0.9481926E-01	0.	0.1200000E 01	
		2		3	0.2763592E-02	0.2383320E-00	0.2253189E-00	0.	0.1200000E 01	
		2		4	0.2377098E-03	0.8601482E-01	0.2383320E-00	0.	0.1200000E 01	
		2		5	0.2049065E-04	0.8619772E-01	0.8601482E-01	0.	0.1200000E 01	
		3		3	0.1899118E-02	0.2399549E-00	0.2491490E-00	0.	0.1200000E 01	
		3		4	0.1667340E-03	0.8779549E-01	0.2399549E-00	0.	0.1200000E 01	
		3		5	0.1457101E-04	0.8739073E-01	0.8779549E-01	0.	0.1200000E 01	
3	51	5	15	0.173056E-00	0.222222E-03	0.399543E-05	0 0 0	0.20116568E-05	1.04166363	0.97710835
				0.386941E-01	0.963995E 01	0.578075E 01	0.579863E 01 0.	0.982120E 00	0.	0.982120E 00
		1		3	0.2658517E-03	0.2234088E-00	0.2655385E-01	0.	0.1200000E 01	
		1		4	0.2434094E-04	0.9155833E-01	0.2234088E-00	0.	0.1200000E 01	
		1		5	0.2529210E-05	0.1039077E-00	0.9155833E-01	0.	0.1200000E 01	
		2		2	0.4292498E-02	0.2447656E-00	0.1753718E-01	0.	0.1200000E 01	
		2		3	0.3745342E-03	0.8725320E-01	0.2447656E-00	0.	0.1200000E 01	
		2		4	0.3224989E-04	0.8610666E-01	0.8725320E-01	0.	0.1200000E 01	
		2		5	0.2796401E-05	0.8671041E-01	0.8610666E-01	0.	0.1200000E 01	
		3		2	0.3120912E-02	0.2411650E-00	0.1294098E-01	0.	0.1200000E 01	
		3		3	0.2770208E-03	0.8876277E-01	0.2411650E-00	0.	0.1200000E 01	
		3		4	0.2417954E-04	0.8728420E-01	0.8876277E-01	0.	0.1200000E 01	
		3		5	0.2140479E-05	0.8852439E-01	0.8728420E-01	0.	0.1200000E 01	
4	66	5	15	0.212778E-00	0.166667E-03	0.724867E-06	0 0 0	0.26822090E-05	1.01190050	0.98873644
				0.139560E-01	0.984117E 01	0.591809E 01	0.592435E 01 0.	0.993741E 00	0.	0.993741E 00
		1		3	0.8099476E-04	0.2051543E-00	0.2572696E-01	0.	0.1200000E 01	
		1		4	0.7294360E-05	0.9005966E-01	0.2051543E-00	0.	0.1200000E 01	
		2		2	0.1581691E-02	0.2427169E-00	0.6516608E-02	0.	0.1200000E 01	
		2		3	0.1385033E-03	0.8756660E-01	0.2427169E-00	0.	0.1200000E 01	
		2		4	0.1196454E-04	0.8638453E-01	0.8756660E-01	0.	0.1200000E 01	
		3		2	0.1092252E-02	0.2426487E-00	0.4501370E-02	0.	0.1200000E 01	
		3		3	0.9738492E-04	0.8915978E-01	0.2426487E-00	0.	0.1200000E 01	

```
        3          4          0.8506314E-05    0.3734734E-01    0.8915975E-01    0.                0.1200000E 01
5   75  4     12  0.2447226E-00  0.1333333E-03  0.2227677E-05   0    0    0   0.2294778BE-05      1.00408055      0.99277104
          0.4979146E-02  0.9911426E 01  0.5966099E 01  0.5968299E 01  0.          0.9977977E 00  0.          0.9977977E 00

        1          3          0.2787765E-04    0.1967143E-00    0.2589043E-01    0.                0.1200000E 01
        1          4          0.2499440E-05    0.8965748E-01    0.1967143E-00    0.                0.1200000E 01
        2          2          0.9508671E-03    0.2398617E-00    0.2338294E-02    0.                0.1200000E 01
        2          3          0.4923644E-04    0.8778628E-01    0.2398617E-00    0.                0.1200000E 01
        2          4          0.4207149E-05    0.8544788E-01    0.8778628E-01    0.                0.1200000E 01
        3          2          0.4008941E-03    0.2341753E-00    0.1711940E-02    0.                0.1200000E 01
        3          3          0.3587197E-04    0.8947993E-01    0.2341753E-00    0.                0.1200000E 01
        3          4          0.3095576E-05    0.8629511E-01    0.8947993E-01    0.                0.1200000E 01
6   90  4     12  0.2769446E-00  0.1111111E-03  0.7799618E-06   0    0    0   0.2637505BE-05      1.00173062      0.99448916
          0.1798638E-02  0.9933673E 01  0.5983399E 01  0.5984138E 01  0.          0.9992638E 00  0.          0.9992638E 00

        1          2          0.6372699E-04    0.2932620E-01    0.2173039E-02    0.                0.1200000E 01
        1          3          0.6853962E-05    0.1075519E-00    0.2932620E-01    0.                0.1200000E 01
        2          2          0.1823925E-03    0.2461703E-00    0.7409198E-03    0.                0.1200000E 01
        2          3          0.1664917E-04    0.8799249E-01    0.2461703E-00    0.                0.1200000E 01
        2          4          0.1392050E-05    0.8673657E-01    0.8799249E-01    0.                0.1200000E 01
        3          2          0.1203830E-03    0.2600389E-00    0.4629424E-03    0.                0.1200000E 01
        3          3          0.1076981E-04    0.8946286E-01    0.2600389E-00    0.                0.1200000E 01
7   108 4     10  0.3038896E-00  0.9923681E-04  0.2160099E-05   0    0    0   0.2413968BE-05      1.00040486      0.99489182
          0.5329948E-03  0.9947962E 01  0.5991112E 01  0.5991113E 01  0.          0.9999968E 00  0.          0.9999968E 00

        1          2          0.1087799E-04    0.4029397E-01    0.2704621E-03    0.                0.1200000E 01
        1          3          0.9555729E-06    0.3777515E-01    0.4029397E-01    0.                0.1200000E 01
        2          2          0.5679553E-04    0.2572054E-00    0.2206178E-03    0.                0.1200000E 01
        2          3          0.4999372E-05    0.8802405E-01    0.2572054E-00    0.                0.1200000E 01
        3          2          0.3780579E-04    0.2596256E-00    0.1456166E-03    0.                0.1200000E 01
        3          3          0.3359815E-05    0.8867038E-01    0.2596256E-00    0.                0.1200000E 01
8   109 3      9  0.3286111E-00  0.8333333E-04  0.9259815E-06   0    0    0   0.2577900BE-05      1.00003241      0.99492408
          0.1500626E-03  0.9994914E 01  0.5991899E 01  0.5991899E 01  0.          0.9999968E 00  0.          0.9999968E 00

        1          2          0.3874760E-05    0.7165704E-01    0.5407367E-04    0.                0.1200000E 01
        2          2          0.1035375E-04    0.2515859E-00    0.6500267E-04    0.                0.1200000E 01
        3          2          0.9528332E-05    0.2376522E-00    0.4009192E-04    0.                0.1200000E 01
9   115 2      6  0.3452788E-00  0.7407416E-04  0.3031856E-05   0    2    0   0.1952052BE-05      1.00002168      0.99494566
          0.4437736E-04  0.9994931E 01  0.5991996E 01  0.5992016E 01  0.          0.9999981E 00  0.          0.9999981E 00

        1          2          0.1200886E-05    0.2655546E-01    0.4522168E-04    0.                0.1200000E 01
        2          2          0.4409387E-05    0.2399775E-00    0.1037417E-04    0.                0.1200000E 01
        3          2          0.2874147E-05    0.2397763E-00    0.1198668E-04    0.                0.1200000E 01
```

ITMA	ITMI	ITMIGX	ITD	TIME REQ	EPS	EPSM	KASOR	KONV	KFINSH	NEUTRON BAL	GLAM	EIGEN
10	121	2	6	0.361944E-00	0.666667E-04	0.816771E-06	0	2	0	0.245869E-05	1.00001255	0.99495815

THE MAXIMUM RATE OF CHANGE IN THE SELF-SCATTERING SOURCE OCCURS AT THE MESH INTERVAL (MXVARI(IG))
13 1 1

THE MAXIMUM RATE OF CHANGE IN THE SELF-SCATTERING SOURCE (ESIJM(IG))
0.144648E-07 0.2623E2E-07 0.184356E-07

TOTAL NEUTRON FLUX (XNG(IG))
0.254001E 01 0.385076E 01 0.355875E 01 0.994953E 01

FISSION SOURCE (FG(IG))
0.333333E-00 0.333333E-00 0.333333E-00 1.000000E 00

IN SCATTERING SOURCE (SING(IG))
0.141877E 01 0.239309E 01 0.218028E 01 0.599214E 01

OUT SCATTERING LOSS (SDG(IG))
0.157800E 01 0.228647E 01 0.212768E 01 0.599215E 01

SELF SCATTERING SOURCE (SGG(IG))
0.962014E 00 0.156430E 01 0.143107E 01 0.395738E 01

NON-SELF SCATTERING SOURCE (SG(IG))
0.175211E 01 0.272642E 01 0.251362E 01 0.699214E 01

TOTAL SOURCE (STG(IG))
0.271412E 01 0.429072E 01 0.394469E 01 0.109495E 02

FISSION NEUTRONS (FNG(IG))
0.254001E-00 0.385076E-00 0.355875E-00 0.994953E 00

ABSORPTIONS (AG(IG))
0. 0. 0. 0.

HORIZONTAL LEAKAGE (HNLG(IG))
0.174108E-00 0.439952E-00 0.385934E-00 0.999993E 00

NET LEAKAGE (XNLG(IG))
0.174108E-00 0.439952E-00 0.385934E-00 0.999993E 00

NEUTRON BALANCE (XNB(IG)) (SOURCE - LOSS)
0.759959E-06 0.864267E-06 0.834465E-06 0.245869E-05

NORMALIZATION FACTOR FOR F, XN, XJR, AND XJZ EQUALS 0.47181510E-00

THE FISSION RATE PER UNIT VOLUME (F(IJ))

J= 1

0.290005E-00	0.289588E-00	0.288755E-00	0.287507E-00	0.285845E-00	0.283772E-00	0.281289E-00	0.278400E-00
0.275108E-00	0.271417E-00	0.267331E-00	0.262856E-00	0.257995E-00	0.252754E-00	0.247139E-00	0.241156E-00
0.234810E-00	0.228107E-00	0.221052E-00	0.213651E-00	0.205907E-00	0.197823E-00	0.189398E-00	0.180628E-00
0.171502E-00	0.162000E-00	0.152085E-00	0.141696E-00	0.130733E-00	0.119036E-00	0.106349E-00	0.922641E-01

YGD0782 CLAY BARBER
THE NON-DIRECTIONAL FLUX PER UNIT VOLUME BY GROUP (XN(IJ))

IG= 1
J= 1

0.919088E 00 0.917160E 00 0.913294E 00 0.907471E 00 0.899656E 00 0.889801E 00 0.877834E 00 0.863654E 00
0.847106E 00 0.827972E 00 0.805905E 00 0.780365E 00 0.750494E 00 0.714896E 00 0.671273E 00 0.615798E 00
0.552890E 00 0.496315E-00 0.450816E-00 0.412807E-00 0.380012E-00 0.350974E-00 0.324732E-00 0.300634E-00
0.278218E-00 0.257136E-00 0.237103E-00 0.217864E-00 0.199156E-00 0.180879E-00 0.162036E-00 0.142671E-00

IG= 2
J= 1

1.000000E 00 0.998990E 00 0.996981E 00 0.993991E 00 0.990055E 00 0.985218E 00 0.979546E 00 0.973131E 00
0.966100E 00 0.958640E 00 0.951028E 00 0.943684E 00 0.937269E 00 0.932845E 00 0.932142E 00 0.938022E 00
0.946855E 00 0.946141E 00 0.938543E 00 0.921342E 00 0.898516E 00 0.871286E 00 0.840412E 00 0.806367E 00
0.769429E 00 0.729727E 00 0.687248E 00 0.641819E 00 0.593054E 00 0.540249E 00 0.482226E-00 0.417065E-00

IG= 3
J= 1

0.980964E 00 0.979734E 00 0.977280E 00 0.973611E 00 0.968743E 00 0.962699E 00 0.955509E 00 0.947215E 00
0.937872E 00 0.927558E 00 0.916382E 00 0.904509E 00 0.892186E 00 0.879801E 00 0.867980E 00 0.857741E 00
0.848315E 00 0.836613E 00 0.821162E 00 0.802360E 00 0.780541E 00 0.755967E 00 0.728834E 00 0.699276E 00
0.667375E 00 0.633139E 00 0.596499E 00 0.557275E 00 0.515119E 00 0.469431E-00 0.419225E-00 0.362905E-00

FLUX INTEGRALS
0.184796E 01 0.216653E 01 0.208984E 01
0.652058E 00 0.168426E 01 0.146893E 01

YGD0782 CLAY BARBER
ACTIVITY TABLES

ZONE IDENTIFICATION NUMBER (ID)= 1

0.92397866E 00	0.	0.18479577E-00	0.18479577E 01	0.69661377E 00	0.72217661E 00	0.61598589E 00
0.	0.	0.18479577E 01				
0.21665298E 01	0.	0.21665298E-00	0.21665298E 01	0.	0.69661377E 00	0.72217661E 00
0.61598589E 00	0.	0.21665298E 01				
0.31347620E 01	0.	0.20898414E-00	0.20898413E 01	0.	0.	0.69661377E 00
0.72217661E 00	0.61598589E 00	0.20898413E 01				

GROUP TOTAL

0.62252703E 01	0.	0.61043284E 00	0.61043283E 01	0.	0.	0.20347761E 01
0.	0.	0.61043283E 01				

ZONE IDENTIFICATION NUMBER (ID)= 2

0.34602898E-00	0.	0.69205797E-01	0.69205797E 00	0.	0.	0.34602898E-00
0.	0.	0.69205797E 00				
0.16842600E 01	0.	0.16842600E-00	0.16842600E 01	0.	0.73446736E 00	0.84213001E 00
0.34602898E-00	0.	0.16842600E 01				
0.22034021E 01	0.	0.14689346E-00	0.14689347E 01	0.	0.	0.73446736E 00
0.84213001E 00	0.	0.14689347E 01				

GROUP TOTAL

0.42336909E 01	0.	0.38452525E-00	0.38452525E 01	0.	0.	0.19226263E 01
0.	0.	0.38452525E 01				

TOTAL OF P(O) AND P(T) ZONES

0.12700077E 01	0.	0.25400155E-00	0.25400155E 01	0.69661377E 00	0.72217661E 00	0.96201482E 00
0.	0.	0.25400155E 01				
0.38507897E 01	0.	0.38507897E-00	0.38507897E 01	0.	0.14310811E 01	0.15643065E 01
0.96201482E 00	0.	0.38507897E 01				
0.53381638E 01	0.	0.35587759E-00	0.35587759E 01	0.	0.	0.14310811E 01
0.15643065E 01	0.61598589E 00	0.35587759E 01				

GROUP TOTAL

0.10458960E 02	0.	0.99495800E 00	0.99495794E 01	0.	0.	0.39574019E 01
0.	0.	0.99495794E 01				

DISADVANTAGE FACTORS

ZONE IDENTIFICATION NUMBER (ID)= 1

0.14550759E 01	0.11252392E 01	0.11744719E 01

ZONE IDENTIFICATION NUMBER (ID)= 2

0.54492422E 00	0.87476084E 00	0.82552808E 00

AVERAGING TABLES

ZONE IDENTIFICATION NUMBER (ID)= 1

0.50000000E 00 0. 0.99999999E-01 1.00000000E 00 0.33333332E-00 0.33333333E-00 0.33333336E-00
0. 0. 0.18479577E 01

1.00000000E 00 0. 0.99999999E-01 1.00000000E 00 0. 0.33333332E-00 0.33333334E-00
0.33333332E-00 0. 0.21665298E 01

0.15000000E 01 0. 0.09999999E-00 1.00000000E 00 0. 0. 0.33333335E-00
0.33333333E-00 0.33333332E-00 0.20898413E 01

COLLAPSED TO ONE GROUP

0.10198124E 01 0. 0.09999999E-00 1.00000000E 00 0. 0. 1.00000000E 00
0. 0. 0.61043283E 01

OUT SCATTERING BY GROUP

0.66666664E 00 0.66666666E 00 0.66666665E 00

ZONE IDENTIFICATION NUMBER (ID)= 2

0.50000000E 00 0. 0.99999999E-01 1.00000000E 00 0. 0. 0.50000000E 00
0. 0. 0.69205797E 00

1.00000000E 00 0. 0.99999999E-01 1.00000000E 00 0. 0.50000000E 00 0.50000000E 00
0.50000000E 00 0. 0.16842600E 01

0.15000000E 01 0. 0.99999996E-01 1.00000000E 00 0. 0. 0.50000000E 00
0.50000000E 00 0. 0.14689347E 01

COLLAPSED TO ONE GROUP

0.11010176E 01 0. 0.99999996E-01 1.00000000E 00 0. 0. 1.00000000E 00
0. 0. 0.38452525E 01

OUT SCATTERING BY GROUP

0.50000000E 00 0.50000000E 00 0.50000000E 00

TOTAL OF P(O) AND P(T) ZONES

0.50000000E 00 0. 0.09999999E-00 1.00000000E 00 0.19574533E-00 0.18753987E-00 0.37874369E-00
0. 0. 0.25400155E 01

1.00000000E 00 0. 0.99999998E-01 1.00000000E 00 0. 0.40212733E-00 0.40623009E-00
0.37874368E-00 0. 0.38507897E 01

0.14999999E 01 0. 0.99999999E-01 1.00000000E 00 0. 0. 0.40212736E-00
0.40623006E-00 0.24251264E-00 0.35587759E 01

COLLAPSED TO ONE GROUP

0.10511962E 01 0. 0.10000000E-00 1.00000000E 00 0. 0. 1.00000000E 00
0. 0. 0.99495794E 01

OUT SCATTERING BY GROUP

0.62125631E 00 0.59376991E 00 0.59787264E 00

```
      YGD0782 CLAY BARBER
GROUP COLLAPSED CROSS SECTIONS


I.D. NUMBERS FOR COLLAPSED GROUPS
              1               1               1


ZONE IDENTIFICATION NUMBER (ID)=  1
 0.10198124E 01  0.             0.99999999E-01  1.00000000E 00  1.00000000E 00  0.61043269E 01


ZONE IDENTIFICATION NUMBER (ID)=  2
 0.11010176E 01  0.             0.99999998E-01  1.00000000E 00  1.00000000E 00  0.38452526E 01


TOTAL OF P(O) AND P(T) ZONES
 0.10511961E 01  0.             0.99999999E-01  1.00000000E 00  1.00000000E 00  0.99495611E 01



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
 0.60681600E 01  0.60681600E 01

FLUX INTEGRALS FOR EDIT SET   1
 0.60681600E 01  0.60681600E 01

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

REFERENCES

1. Carlson, Bengt G.: Solution of the Transport Equation by S_n Approximations. Rep. No. LA-1891, Los Alamos Scientific Lab., Feb. 1955.

2. Carlson, B. G.; and Bell, G. I.: Solution of the Transport Equation by the S_n Method. Nuclear Data and Reactor Theory. Vol. 16 of Proceedings of the Second United Nations International Conference on the Peaceful Uses of Atomic Energy, United Nations, 1958, pp. 535-549.

3. Carlson, Bengt G.: Numerical Solution of Transient and Steady-State Neutron Transport Problems. Rep. No. LA-2260, Los Alamos Scientific Lab., May 1959.

4. Carlson, Bengt G.: The Numerical Theory of Neutron Transport. Statistical Physics. Vol. 1 of Methods in Computational Physics, Berni Alder, Sidney Fernbach and Manuel Rotenberg, eds., Academic Press, 1963, pp. 1-42.

5. Putnam, G. E.: Topic-A Fortran Program for Calculating Transport of Particles in Cylinders. Rep. No. IDO-16968, Phillips Petroleum Co., Apr. 1964.

6. Blue, E.; and Flatt, H. P.: Convergence of the S_n Method for Thermal Systems. Nucl. Sci. Eng., vol. 7, no. 2, Feb. 1960, pp. 127-132.

7. Lee, Charles E.: The Discrete S_n Approximation to Transport Theory. Rep. No. LA-2595, Los Alamos Scientific Lab., June 1961.

8. Alexander, J. H.; Hinman, G. W.; and Triplett, J. R.: GAPLSN-A Modified DSN Program for the Solution of the One-Dimensional Anisotropic Transport Equation. Rep. No. GA-4972, General Atomics, Mar. 1964.

9. Ackroyd, R. T.; and Pendlebury, E. D.: Survey of Theoretical Calculation Methods. Criticality Control in Chemical and Metallurgical Plant, European Nuclear Energy Agency (Paris), 1961, pp. 47-118.

10. Lathrop, K. D.: DTF-IV-A FORTRAN-IV Program for Solving the Multigroup Transport Equation with Anisotropic Scattering. Rep. No. LA-3373, Los Alamos Scientific Lab., July 1965.

11. Clark, Melville, Jr.; and Hansen, Kent F.: Numerical Methods of Reactor Analysis. Academic Press, 1964, pp. 125-129.

12. Fletcher, J. L.; Jewett, J. P.; and Reilly, E. D., Jr.: FLEER. A Two-Dimensional Triangular Mesh Diffusion Program for the IBM 704. Rep. No. KAPL-2086, Knolls Atomic Power Lab., 1960.

13. Polyak, B. T.: Some Methods of Accelerating the Convergence of Iterative Schemes. Rep. No. JPRS-28179 (TT 65-30083), Joint Publications Research Service, Jan. 1965.

14. Staff of IBM: IBM 7090/7094 Programming Systems, FORTRAN IV Language. Form C 28-6274-1, International Business Machines, May, 1963.

15. Staff of IBM: IBM 7090/7094 IBSYS Operating System, IBJOB Processor Debugging Package. Form C28-6362-1, International Business Machines, June, 1964.

16. Carlson, Bengt G.; Worlton, William J.; Guber, Walter; and Shapiro, Morton: DTF Users Manual. Rep. No. UNC Phys/Math 3321, Vols. I and II, United Nuclear Corp., Nov. 1963.