2DBTOR -- A TOROIDAL GEOMETRY NEUTRON DIFFUSION CODE

A Thesis

by

CRAIG ANTHONY HRABAL

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 1990

Major Subject: Nuclear Engineering

2DBTOR -- A TOROIDAL GEOMETRY NEUTRON DIFFUSION CODE

A Thesis

by

CRAIG ANTHONY HRABAL

Approved as to style and content by:

Theodore A. Parish
(Chairman of Committee)

Gerald A. Schlapper
(Member)

R. Dante DeBlassie
(Member)

John W. Poston
(Head of Department)

August 1990

ABSTRACT

2DBTOR -- A Toroidal Geometry Neutron Diffusion Code (August 1990)

Craig Anthony Hrabal, B.S., Texas A&M University;

Chair of Advisory Committee: Dr. Theodore A.Parish

The objective of the research performed here was to produce a scoping code that could be used for fusion reactor blanket design. To this end, the present research initially explored a technique proposed by Pomraning and Stevens, in which the toroidal diffusion problem in toroidal geometry is cast into cylindrical (r-θ) form by a spatially dependent redefinition of the diffusion coefficient, absorption cross-section, and extraneous source function. This idea was explored but was later abandoned in favor of the direct finite differencing of the toroidal diffusion equation.

The direct finite differencing approach was programmed into an existing two-dimensional(x-y, r-z, r-θ, triangular), multi-group neutron diffusion code, 2DB, that had previously been converted to execute on the IBM-AT. Neutronic scoping calculations relevant to fusion reactor design were then performed in a micro-computer environment. The modified code was renamed 2DBTOR.

To verify that 2DBTOR was operating correctly, comparisons were made to both analytical and numerical solutions for several types of problems. Both ANISN and 2DB were used to verify and compare the solutions obtained from 2DBTOR. It was also shown that as the aspect ratio approached infinity (i. e., the major radius became large) the 2DBTOR solution approached the solution for that of 1-D cylindrical geometry. After verifying the solution for a large major radius, the errors associated

with using a non-toroidal scoping code were examined versus using 2DBTOR. To accomplish this, neutron cross-sections for a benchmark problem were input to 2DBTOR and the output was compared to that from ANISN. A method proposed by Price and Chapin, that used volume correction factors to compute the reaction rates in the benchmark blanket , was utilized to provide a means of checking 2DBTOR's results versus those given by a Monte Carlo code. It is also worth noting that 2DBTOR makes possible the calculation of material depletion in the fusion blanket, which is a unique advantage of the new program, 2DBTOR.

In future versions of the 2DBTOR program, it is recommended that the central vacuum should be modelled through an internal boundary condition. A separate void streaming calculation should be used to define the internal boundary condition by specifying the neutron flux to current ratio as a function of position along the vacuum wall. Improved modelling of the central void region will be required if 2DBTOR is to prove to be an attractive program for Tokamak blanket scoping calculations.

ACKNOWLEDGEMENTS

The author would like to take this opportunity to express his gratitude to several people who have made this work possible. First of all, I would like to thank my committee chairman, Dr. Theodore A. Parish, for suggesting this thesis, and for his guidance in its development. Also, I wish to thank Babatunde O. Shofolu for the help given on the computer and Jodine Jansen for her help with the manuscript preparation. Finally, I would like to thank my wife, Karen, for her constant encouragement and moral support.

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
	I.A The Fusion Process	2
	I.B The Need for Scoping Codes to Study the Fusion Process	5
	I.C Previous Studies	6
	I.D Computer Codes Used in This Work	7
	I.E Objectives of This Work	8
II.	THEORY	10
	II.A Introduction	10
	II.B Pomraning and Stevens' Method	22
	II.C Finite Difference Approximation to the Diffusion Equation in Cylindrical Coordinates	24
	II.D Finite Difference Approximation to the Diffusion Equation in Toroidal Coordinates	33
	II.E Analytical Solutions	37
III.	RESULTS	40
	III.A 2DBTOR Verification and Infinite Cylinder Problems	40
	III.B 2DBTOR Verification and Toroidal Problems	43
	III.C Standard Blanket Solutions	46
	III.C.1 Infinite Cylinder	46
	III.C.2 Comparison of Torus and Infinite Cylinder Models	49
IV.	SUMMARY AND CONCLUSIONS	64

TABLE OF CONTENTS (CONT.)

	Page
REFERENCES	66
APPENDIX A. Pomraning and Stevens' Analytical Solution	69
APPENDIX B. Average D_k Solution	72
APPENDIX C. XPROC2DB Listing	75
APPENDIX D. 2DB Errata	79
APPENDIX E. Plasma Source	81
APPENDIX F. Equivalent Source to the Plasma Source at the Inner Edge of the Blanket	83
APPENDIX G. 2DBTOR Manual	84
APPENDIX H. 2DBTOR Listing	147
VITA	237

LIST OF FIGURES

Figure Page

1-1 Reactivity Curves.......... 4

2-1 Toroidal Coordinate Description.......... 20

2-2 Finite Difference Coordinates Used in 2DB and 2DBTOR.......... 25

3-1 Flux versus Poloidal Angle at a Radius of 150 cm for Several Aspect Ratios.......... 44

3-2 Flux versus Poloidal Angle at a Radius of 250 cm for Several Aspect Ratios.......... 45

3-3 Standard Blanket Configuration.......... 47

LIST OF TABLES

Table		Page
2-1	Area and Volume Elements in Cylindrical and Toroidal Coordinates	38
3-1	Standard Blanket Constituents	48
3-2	Radial Averaged Tritium Breeding Ratio's for 2DB and ANISN, Infinite Cylinder Case	50
3-3	Radial Averaged Tritium Breeding Ratio's for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 3)	52
3-4	Radial Averaged Tritium Breeding Ratio's for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 5)	53
3-5	Zone by Zone Averaged T6 for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 3)	54
3-6	Zone by Zone Averaged T7 for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 3)	55
3-7	Zone by Zone Averaged T6 for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 5)	56
3-8	Zone by Zone Averaged T7 for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 5)	57
3-9	T6 for 2DBTOR Versus Volume Corrected T6 for 2DB, Torus Case (Aspect Ratio = 3)	60
3-10	T7 for 2DBTOR Versus Volume Corrected T7 for 2DB, Torus Case (Aspect Ratio = 3)	61

Table		Page
3-11	T6 for 2DBTOR Versus Volume Corrected T6 for 2DB, Torus Case (Aspect Ratio = 5)	62
3-12	T7 for 2DBTOR Versus Volume Corrected T7 for 2DB, Torus Case (Aspect Ratio = 5)	63

I. INTRODUCTION

In order to sustain the current level of world civilization man requires energy. However, at the present pace of energy demand, it is predicted that coal and U-235 resources will be depleted in the next 25 to 100 years.[1] To meet the world's energy demand as the above resources become more scarce, new options will have to be employed. Of the methods available to produce energy, only solar energy, fission breeders, and fusion are thought to be capable of meeting the world's long-term needs. Of these three energy resources, fission breeders produce and use special nuclear materials, which makes them undesirable without stringent safeguards. This leaves solar energy and fusion as the most viable energy resources to meet future energy demands. Both will have to be developed, since a stable society needs alternate sources of energy to call upon in an uncertain future.

Because of progress in plasma confinement, fusion has warranted increased attention in the past decade. What makes fusion of most interest is the fact that one of the fuels required, deuterium, is essentially an inexhaustible resource. Deuterium, which has an average abundance of 0.015% in elemental hydrogen, can be separated relatively easily and cheaply from water.[2] A secondary, but perhaps more important advantage, as regards public perception, is fusion's inherent safety and reduced radioactivity hazard relative to fission. Tritium, one of the more radioactive of the fusion fuels, has a relatively short half-life(12.36y) and decays by emitting low energy β-rays. Unfortunately, activation of structural materials, such as the first wall in a fusion reactor, presents a possible hazard, but this can be controlled by choosing

---

This thesis follows the style of Nuclear Technology.

suitable materials. In regard to safety, with magnetic confinement fusion, a sudden increase in power is likely to be counteracted by altering the conditions that are necessary to position and heat the plasma. In addition to the above advantages, use of exotic fuels which only produce charged particles, might lead to energy conversion efficiencies approaching 100%, since charged particles can possibly be directly converted to electricity.

I.A The Fusion Process

Fusion is essentially the process of two nuclei coming together to form one or more nuclei with an accompanying release of energy. Since very high temperatures (over 10^8 °C) are required to overcome the Coulomb repulsion between the reacting nuclei, a plasma must be produced. The nuclei, which are stripped of electrons at such temperatures, must be confined long enough to fuse. As a consequence of the high temperatures required for fusion, the reactants can not be contained within physical walls, since interactions with the wall material would likely cool the nuclei down below their required temperature for fusion. By virtue of the nuclei being charged particles, however, they can be contained by magnetic fields in various plasma confinement configurations.

Several reactions are considered to be possible for producing power from fusion break-even conditions, but the deuterium-tritium (D-T) reaction has the best chance of being the first to reach the required plasma conditions. The D-T reaction is described below:

$$D + T \rightarrow {}^4He + n + 17.6\ \mathrm{MeV} \qquad (1)$$

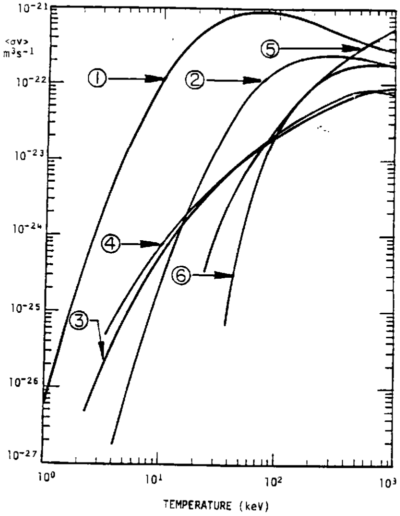
This reaction requires the use of of tritium, which is extremely rare in nature and must be artificially produced. Tritium production is expected to be accomplished by surrounding the plasma with a lithium containing blanket. Tritium is produced from lithium according to the neutron reactions that follow:

$$n + {}^{7}Li \rightarrow T + {}^{4}He + n \tag{2}$$

$$n + {}^{6}Li \rightarrow T + {}^{4}He \tag{3}$$

The reaction with Li-7 has a threshold of 2.7 MeV, while the Li-6 reaction takes place with neutrons down to thermal energies. Although the above reactions require consumption of lithium, the known U.S. reserves of lithium are thought to be able to last 600 years. [3] The D-T reaction discussed above has great potential for fusion due to its large reactivity at relatively low temperatures(<100 keV; 1 keV $\approx 10^9$ °C). The reactivity, $<\sigma v>_{jk}$, is defined to be the average of the fusion cross-section multiplied by the relative speed between the reacting particles j and k averaged over Maxwellian distributions. Curves of the reactivity versus energy for several fuels are shown in Figure 1-1[4]. As can be seen, the D-T reactivity is by far the highest at temperatures below 100 keV. Since the reaction rate per unit volume for two nuclei can be shown to be given by

$$RR_{jk} = a_{jk} n_j n_k <\sigma v>_{jk} \tag{4}$$



Reaction rate parameters for mixtures of Maxwellian distributions at the same temperature. (1) $D + T \rightarrow n + {}^4He$, (2) $D + {}^3He \rightarrow H + {}^4He$, (3) $D + D \rightarrow H + T$, (4) $T + T \rightarrow {}^4He + 2n$, (5) $T + {}^3He \rightarrow$ (various products), (6) $H + {}^{11}B \rightarrow 3({}^4He)$.

Figure 1-1. Reactivity Curves.

where RR_{jk} = reaction rate per unit volume for nuclei j and k,

a_{jk} = 1 for $j \neq k$ and 1/2 for $j=k$,

n_j = atom density of nuclei j,

and n_k = atom density of nuclei k,

it is apparent that the D-T reaction for fixed reactant densities will provide the greatest number of fusions per cm^3 per sec at the lowest temperatures.[5] Since lowering of the temperature needed for fusion makes the required plasma conditions more achievable, it is likely that the first fusion reactions will use D-T fuel.

I.B The Need for Scoping Codes to Study the Fusion Process

In the past decade, as plasma confinement experiments have advanced nearer to break-even, there has been a concurrent interest in the engineering problems associated with power production from nuclear fusion. Since a magnetically confined plasma burning D-T fuel should be a reality within the next decade, calculations of the transport of 14 MeV neutrons through the first wall and blanket of a fusion reactor have become particularly pertinent. Accurate computations of neutron transport are currently feasible; however, these calculations can be labor intensive and expensive. To evaluate the myriad of possibilities for the first wall and blanket, scoping codes to estimate reaction rates and leakage could be advantageous for design purposes.

Engineers who do neutronic computations for fission reactors commonly use neutron transport and diffusion codes to determine the flux profiles within a nuclear reactor. By using a neutron diffusion code, scoping runs can be made to provide an initial estimate for the actual flux profiles. Since diffusion calculations can be performed more easily than the more exact calculations based on transport theory, neutron diffusion codes are used to decrease the computational effort involved in the trial and error process that takes place in designing the cores of nuclear reactors. In the same manner, a need has arisen to perform flux computations for the nuclear design of Tokamak-type fusion reactors which have a toroidal or doughnut shape.

I.C Previous Studies

Several computational models have been applied to estimate the flux profile in toroidally shaped blankets. One approach is to perform diffusion or transport calculations in one-dimensional cylindrical geometry and to ignore the curvature associated with the major radius. In particular, the geometry of the toroidal reactor blanket can be approximated as an infinite right circular cylinder, so that a one-dimensional discrete ordinates code, such as ANISN[6], can be used to solve for the neutron flux in the radial direction. This method is advantageous in that one-dimensional codes are readily available, easy to use, and relatively inexpensive to run. On the other hand, because one-dimensional cylindrical diffusion and transport codes can not model the curvature associated with the major radius present in Tokamaks, such codes are probably not suitable for tori with small aspect ratios.

To take into account the curvature of the major radius as well as the minor radius, two-dimensional cylindrical calculations in r-z geometry have been used to solve for the neutron flux. A typical code for this method is TWOTRAN-II[7], a two-dimensional discrete ordinates transport code which can be run in r-z geometry. Although it is possible to model the curvature of the major radius adequately using such a code, small mesh spacings are required to correctly model curved surfaces in the r-z plane; this can lead to costly and time consuming computations. More recently, computer programs have been developed to perform neutron transport calculations that are more suitable for toroidal geometry. One example of such a code is TRISM[8], a two-dimensional discrete ordinates transport code developed at Los Alamos National Laboratory. Finally, full three-dimensional and Monte Carlo[9] programs have also been applied to blanket problems. While these more advanced methods allow fluxes to be calculated accurately, they can be quite cumbersome and expensive to use.

I.D Computer Codes Used in This Work

2DB, a code written primarily for use in fast reactor calculations, provides two options that are potentially useful for fusion reactor design.[10] First, a fixed source option will enable modeling of fusion blanket problems. Second, material depletion in the blanket of a Tokamak can potentially be calculated by using 2DB's material burn-up option.

ANISN, as stated previously, is a neutron transport code commonly used for fusion blanket analysis in one dimension. It can also be used to provide for a benchmark comparison to the one-dimensional results obtained with 2DB. In addition, ANISN enables one to reduce the number of groups used in the calculations by averaging over the fine groups to produce broad group cross sections. Reducing the number of groups will prove useful in a neutron diffusion code, since run times can be greatly decreased, thus facilitating its use on a micro-computer.

Since ANISN is a transport code, the results for the flux distribution will be more accurate than those obtained based on diffusion theory. In particular, if the validity of diffusion theory in the vacuum region of a Tokamak inordinately affects the flux distribution in the rest of the Tokamak, then results from ANISN will provide a means to check this in 2DB.

The neutron cross section data used in the above codes will be taken from the CLAW-IV[11] library. Most of the materials suggested for fusion applications are contained in this multi-group cross section set. Each material in the CLAW-IV library has four cross section matrices (P_0 - P_3), with 30 neutron groups and 12 photon groups coupled in each. In the diffusion calculations, only the P_0 matrix is used.

I.E Objectives of This Work

The objective of the research performed here is to produce a scoping code that can be used for fusion blanket design. Pomraning and Stevens[12] have previously explained a technique in which a toroidal geometry problem can be cast into cylindrical (r-θ) form by a spatially dependent redefinition of the diffusion coefficient, absorption cross-section, and extraneous source function. Use of this method will first be attempted in an existing two-dimensional(x-y, r-z, r-θ, triangular), multi-group neutron diffusion code, 2DB, that already executes on the IBM-AT. This method will also be assessed and compared to direct finite differencing of the toroidal diffusion equation. The method with the most advantages will then be implemented in a new version of 2DB that will be called 2DBTOR.

To verify that 2DBTOR is operating correctly, comparisons will be made to both analytical and numerical solutions for several types of problems. ANISN and 2DB will be used to verify the solutions obtained from 2DBTOR.

The modifications that were made to 2DB, a description of the problems and analytical solutions used to verify 2DBTOR, and the further modifications that were necessary to use 2DBTOR for fusion blanket calculations are presented in the following sections of this thesis.

II. THEORY

II.A. Introduction

The design of tokamak reactors requires an accurate calculation of the flux profile in the toroidally-shaped blanket. As stated previously, an r-θ neutron diffusion code will be used, with the appropriate modifications, to obtain a solution for the flux while taking into account the curvature of the torus. The solution from this modified neutron diffusion code might then be used as a first approximation for later runs in a more accurate transport code, such as TRISM[8].

In order to make the appropriate changes so that a two-dimensional, r-θ, diffusion code can be made to model toroidal geometry, an understanding of the form of the diffusion equation in toroidal geometry is necessary. The diffusion equation in toroidal geometry can be developed from the equation for the time rate of change of the number of neutrons at energies, E, in an arbitrary differential volume, dV. Neutrons of energies between E and E+dE, within dV, can be lost or gained by a variety of processes including: (1) production directly from a source, (2) absorption, (3) leakage and (4) scattering. The time rate of change of the number of neutrons in dV and between E and E+dE can be obtained by integrating the neutron density (n(r,E,t)) over dV, and balancing this with the gains and losses as follows:

$$\frac{\partial}{\partial t}\int_{v}\frac{\Phi(\bar{r},E,t)}{\nu}\,dV = \begin{bmatrix}\text{source neutron}\\ \text{production rate}\\ \text{in V at E}\end{bmatrix} - \begin{bmatrix}\text{absorption}\\ \text{rate in V}\\ \text{at E}\end{bmatrix} - \begin{bmatrix}\text{change due}\\ \text{to leakage}\\ \text{from V at E}\end{bmatrix} - \begin{bmatrix}\text{neutron scattering}\\ \text{rate out of}\\ \text{E in V}\end{bmatrix} + \begin{bmatrix}\text{neutron scattering}\\ \text{rate into}\\ \text{E in V}\end{bmatrix} \tag{5}$$

where $\Phi(r,E,t)$ = flux of neutrons in r at E and t = $n(r,E,t)v$

and ν = the speed of the neutrons at E.

Equation (5) is known as the neutron continuity equation. Since the energy dependence of the neutron cross sections vary, equation (5) is usually solved for discrete energy groups(groups denoted by g in this case); thus equation (5) can be written as

$$\frac{\partial}{\partial t}\int_{v}\frac{\Phi_g(r,t)}{\nu_g}\,dV = \begin{bmatrix}\text{source neutron}\\ \text{production rate}\\ \text{in V for}\\ \text{group, g}\end{bmatrix} - \begin{bmatrix}\text{absorption}\\ \text{rate in V}\\ \text{for group, g}\end{bmatrix} - \begin{bmatrix}\text{change due}\\ \text{to leakage}\\ \text{from V}\\ \text{for group, g}\end{bmatrix} - \begin{bmatrix}\text{neutron scattering}\\ \text{rate out of}\\ \text{group, g in V}\end{bmatrix} + \begin{bmatrix}\text{neutron scattering}\\ \text{rate into}\\ \text{group, g in V}\end{bmatrix} \tag{6}$$

Before substituting expressions for the terms in equation (6), it is necessary to consider the flow of neutrons in a medium. Since the neutron current vector, J(r,E,t) is a vector quantity, then

$J \cdot n$ = net rate of flow across a surface, A (n = outward unit normal to a surface, A). (7)

To determine the total net rate of flow out of a closed surface, S, integration of equation (7) over S yields

$$\int_S \vec{J}(\vec{r},E,t) \cdot \hat{n}\, dA,$$

where n is the unit normal to the surface dA. Using the Divergence Theorem, then

$$\int_S \vec{J}(\vec{r},E,t) \cdot \hat{n}\, dA = \int_V \vec{\nabla} \cdot \vec{J}(\vec{r},E,t) dV . \tag{8}$$

Since equation (6) will be used to solve for the neutron flux, it is necessary to relate J(r,E,t) to the flux. In the diffusion approximation the relation between current and flux is assumed to be as follows:

$$\vec{J}(\vec{r},E,t) \cong -D(\vec{r})\vec{\nabla}\Phi(\vec{r},E,t) \tag{9}$$

where D(r) = the diffusion coefficient.

Equation (9), known as Fick's Law for Diffusion, describes the flow of neutrons as being proportional to the negative of the density(flux) gradient, since particles tend to flow from a region of higher density to a region of lower density . Fick's Law is only valid for:

1.) points away from a vacuum boundary,

2.) points away from sources (by a few mean free paths),

3.) isotropic scattering (e.g., equal probability of scattering in any direction,

and 4.) a slowly varying flux (absorption << scattering).[13]

The above conditions can be somewhat relaxed, depending on the accuracy required for the solution of equation (6).

Using Fick's Law it is now possible to express the balance equation only in terms of the neutron flux. Substituting the right hand side (RHS) of equation (9) into the leakage term found from the Divergence Theorem (equation (8)), then

$$\begin{bmatrix}\text{change due}\\ \text{to leakage}\\ \text{from V for}\\ \text{group, g}\end{bmatrix} = \int_V \vec{\nabla} \cdot \left(-D_g(\vec{r})\vec{\nabla}\Phi_g(\vec{r},t)\right) dV \tag{10}$$

The use of Fick's Law in the neutron balance equation, equation (6), is known as the diffusion approximation, and the resulting equations are known as neutron diffusion equations.

The source term in equation (6) can be expressed by defining a source density, $S_g(r,t)$, for the group, g. If $S_g(r,t)$ is integrated over V, then

$$\begin{bmatrix}\text{source neutron} \\ \text{production rate} \\ \text{in V for} \\ \text{group, g}\end{bmatrix} = \int_V S_g(\vec{r},t)\ dV\ . \tag{11}$$

If source neutrons are allowed to be produced by either fission or extraneous(independent of the neutron flux) sources, then let

$$S_g(\vec{r},t) = S_g^{ext}(\vec{r},t) + S_g^{fis}(\vec{r},t) \tag{12}$$

where $S_g^{ext}(\vec{r},t)$ = extraneous source density rate

and $S_g^{fis}(\vec{r},t)$ = fission source density rate.

The fission source density rate is given by

$$S_g^{fis}(\vec{r},t) = \begin{bmatrix}\text{average number of} \\ \text{fission neutrons emitted} \\ \text{within group, g}\end{bmatrix} \sum_{g'=1}^{G} \left\{ \begin{bmatrix}\text{fission rate in} \\ \text{group, g' per} \\ \text{unit volume}\end{bmatrix} \begin{bmatrix}\text{average number of} \\ \text{neutrons released} \\ \text{into group, g, that} \\ \text{occur from fission} \\ \text{in group, g'}\end{bmatrix} \right\} \tag{13}$$

where G = total number of groups.

It is common to denote the average number of neutrons released into group, g, that occur from fission in group, g', by ν_g and the average number of fission neutrons emitted within group, g, by χ_g.[14] Thus

$$S_g^{fis}(\vec{r},t) = \chi_g \sum_{g'=1}^{G} \nu_{g'} \begin{bmatrix} \text{fission rate} \\ \text{in group, } g' \\ \text{per unit volume} \end{bmatrix} \tag{14}$$

Since

$$\begin{bmatrix} \text{reaction rate} \\ \text{per unit} \\ \text{volume} \end{bmatrix} = \Sigma\, \Phi \tag{15}$$

where Σ = macroscopic cross section,

the expression for the fission source can be written as

$$S_g^{fis}(\vec{r},t) = \chi_g \sum_{g'=1}^{G} \nu_{g'} \Sigma_{f_g}(\vec{r})\, \Phi_{g'}(\vec{r},t)\,. \tag{16}$$

The rest of the terms in the neutron balance equation can be expressed analogously to the above. For the absorption rate term, the absorption rate per unit volume is integrated over V to give

$$\begin{bmatrix} \text{absorption rate} \\ \text{in V for group, g} \end{bmatrix} = \int_V \Sigma_{a_g}(\vec{r})\, \Phi_g(\vec{r},t)\, dV \tag{17}$$

where $\Sigma_{ag}(r)$ = the macroscopic absorption cross-section in group g.

Assuming that neutrons can not scatter to groups of higher energy (upscatter), then the neutron outscattering rate term is given by(decreasing group numbers correspond to increasing neutron energy)

$$\begin{bmatrix}\text{neutron scattering rate}\\ \text{out of group g in V}\end{bmatrix} = \int_V \sum_{g'=g+1}^{G} \Sigma_s(g \to g')\, \Phi_g(\vec{r},t)\, dV \tag{18}$$

and the neutron inscattering rate term is given by

$$\begin{bmatrix}\text{neutron scattering rate}\\ \text{into group g in V}\end{bmatrix} = \int_V \sum_{g'=1}^{g-1} \Sigma_s(g \to g')\, \Phi_{g'}(\vec{r},t)\, dV \tag{19}$$

where $\Sigma_S(g \to g')$ = macroscopic scattering cross-section from g to g'

and $\Sigma_S(g \to g')$ = macroscopic scattering cross-section from g' to g.

Now that all of the terms in the neutron balance equation have been determined, equation (6) can be rewritten. Assuming steady state ($\partial/\partial t$ term = 0) and no upscatter, the neutron balance equation becomes, combining terms,

$$0 = \int_V \Big(S_g^{ext} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_g} \Phi_{g'}) - \Sigma_{a_g} \Phi_g - \sum_{g'=g+1}^{G} (\Sigma_s(g \to g') \Phi_g) \quad (20)$$

$$+ \vec{\nabla} \cdot (D_g \vec{\nabla} \Phi_g) + \sum_{g'=1}^{g-1} (\Sigma_s(g \to g') \Phi_{g'}) \Big) \, dV$$

where the (r,t) has been dropped for clarity. Since the volume, V, was arbitrarily chosen, equation (20) reduces to the form that follows:

$$- \vec{\nabla} \cdot (D_g \vec{\nabla} \Phi_g) + \Sigma_{a_g} \Phi_g + \sum_{g'=g+1}^{G} (\Sigma_s(g \to g') \Phi_g) = \quad (21)$$

$$S_g^{ext} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_g} \Phi_{g'}) + \sum_{g'=1}^{g-1} (\Sigma_s(g \to g') \Phi_{g'})$$

Since the removal of neutrons from group g is caused by both downscattering and absorption, the removal cross section for group g is defined as shown below:

$$\Sigma_g^r \equiv \Sigma_{a_g} + \sum_{g'=g+1}^{G} (\Sigma_s(g \to g'))$$

$$= \Sigma_{a_g} + \Sigma_s(g \to g) + \left[\sum_{g'=g+1}^{G} \Sigma_s(g \to g') \right] - \Sigma_s(g \to g)$$

$$= \Sigma_{tr_g} - \Sigma_s(g \to g')$$

where Σ_{tr_g} = the macroscopic transport cross section $= 1/(3D_g)$.[14]

Thus the removal rate/cm^3 is

$$\Sigma^r_g \Phi_g = \Sigma_{a_g} \Phi_g + \sum_{g'=g+1}^{G} (\Sigma_s(g \to g')\, \Phi_g). \tag{22}$$

Finally, substituting equation (22) into equation (21) gives

$$-\vec{\nabla} \cdot (D_g \vec{\nabla} \Phi_g) + \Sigma^r_g \Phi_g = S_g^{ext} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_{g'}} \Phi_{g'}) + \sum_{g'=1}^{g-1} (\Sigma_s(g \to g')\, \Phi_{g'}) \tag{23}$$

To apply the above equation to practical problems, the ∇ operator must be written in the coordinate system of interest. In general orthogonal curvilinear coordinates

$$\vec{\nabla}\Phi = \frac{1}{h_1}\frac{\partial \Phi}{\partial u_1}\hat{e}_1 + \frac{1}{h_2}\frac{\partial \Phi}{\partial u_2}\hat{e}_2 + \frac{1}{h_3}\frac{\partial \Phi}{\partial u_3}\hat{e}_3$$

and

$$\vec{\nabla} \cdot \vec{A} = \frac{1}{h_1 h_2 h_3}\left[\frac{\partial}{\partial u_1}(h_2 h_3 A_1) + \frac{\partial}{\partial u_2}(h_3 h_1 A_2) + \frac{\partial}{\partial u_3}(h_1 h_2 A_3)\right]$$

where u_1, u_2, and u_3 are the coordinates;

e_1, e_2, and e_3 are the corresponding coordinate vectors;

and h_1, h_2, and h_3 are scale factors that depend on the coordinates.

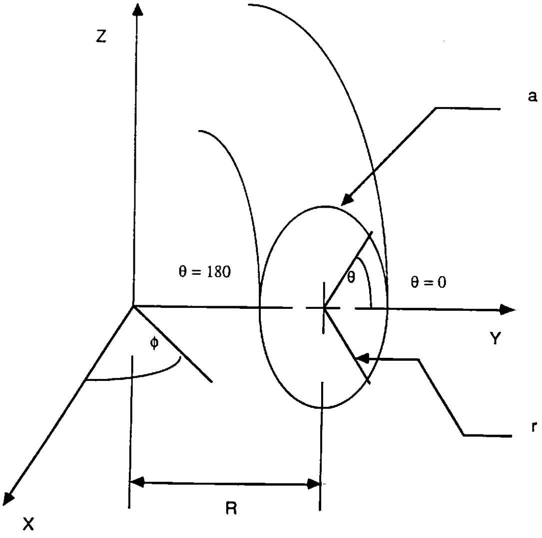
For cylindrical coordinates (r, θ, z)

$h_1 = 1$	$u_1 = r$	$e_1 = r$
$h_2 = r$	$u_2 = \theta$	$e_2 = \theta$
$h_3 = 1$	$u_3 = z$	$e_3 = z.$

For toroidal coordinates (r, θ, ϕ) (see Figure 2-1)

$h_1 = 1$	$u_1 = r$	$e_1 = r$
$h_2 = r$	$u_2 = \theta$	$e_2 = \theta$
$h_3 = R + r\cos\theta$	$u_3 = \phi$	$e_3 = \phi.$[15]

Consider the diffusion equation in cylindrical coordinates. From the above,



R = major radius, a = minor radius, r = radial coordinate,
θ = poloidal angle, and ϕ = toroidal angle.

Figure 2-1: Toroidal Coordinate Description.

$$D\vec{\nabla}\Phi = D\frac{\partial\Phi}{\partial r}\hat{r} + \frac{D}{r}\frac{\partial\Phi}{\partial\theta}\hat{\theta} + D\frac{\partial\Phi}{\partial z}\hat{z} \tag{24}$$

Also,

$$\vec{\nabla}\cdot(D\vec{\nabla}\Phi) = \frac{1}{r}\left[\frac{\partial}{\partial r}(rD\frac{\partial\Phi}{\partial r}) + \frac{\partial}{\partial\theta}(\frac{D}{r}\frac{\partial\Phi}{\partial\theta}) + \frac{\partial}{\partial z}(rD\frac{\partial\Phi}{\partial z})\right] \tag{25}$$

Analogously for toroidal coordinates,

$$D\vec{\nabla}\Phi = D\frac{\partial\Phi}{\partial r}\hat{r} + \frac{D}{r}\frac{\partial\Phi}{\partial\theta}\hat{\theta} + \frac{D}{R+r\cos\theta}\frac{\partial\Phi}{\partial\phi}\hat{\phi} \tag{26}$$

and

$$\vec{\nabla}\cdot(D\vec{\nabla}\Phi) = \frac{1}{r(R+r\cos\theta)}\left[\frac{\partial}{\partial r}(r(R+r\cos\theta)D\frac{\partial\Phi}{\partial r}) + \frac{\partial}{\partial\theta}((R+r\cos\theta)\frac{D}{r}\frac{\partial\Phi}{\partial\theta}) + \frac{\partial}{\partial\phi}(\frac{rD}{(R+r\cos\theta)}\frac{\partial\Phi}{\partial\phi})\right] \tag{27}$$

The diffusion equation in a cylindrical coordinate system, assuming axial symmetry (i.e., the flux, $\Phi(r, \theta, z)$ depends only on the r and θ coordinates), is given by

$$-\frac{1}{r}\left[\frac{\partial}{\partial r}(rD_g\frac{\partial \Phi_g}{\partial r}) + \frac{\partial}{\partial \theta}(\frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta})\right] + \Sigma_g^r \Phi_g$$

$$= S_g^{ext} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_g} \Phi_{g'}) + \sum_{g'=1}^{g-1} (\Sigma_s(g \to g') \Phi_{g'}) \tag{28}$$

Similarly the diffusion equation in a toroidal coordinate system with axisymmetry (i.e., the flux, $\Phi(r, \theta, \phi)$ depends only on the r and θ coordinates), is given by

$$-\frac{1}{r(R+r\cos\theta)}\left[\frac{\partial}{\partial r}(r(R+r\cos\theta)D_g\frac{\partial \Phi_g}{\partial r}) + \frac{\partial}{\partial \theta}((R+r\cos\theta)\frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta})\right]$$

$$+\Sigma_g^r \Phi_g = S_g^{ext} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_{g'}} \Phi_{g'}) + \sum_{g'=1}^{g-1} (\Sigma_s(g \to g') \Phi_{g'}) \tag{29}$$

II.B. Pomraning and Stevens' Method

If the toroidal form of the diffusion equation (equation (25)) is multiplied by (R + $r\cos\theta$)/R, then

$$-\frac{1}{r}\frac{\partial}{\partial r}(r(1+\frac{r}{R}\cos\theta)D_g\frac{\partial\Phi_g}{\partial r})-\frac{1}{r^2}\frac{\partial}{\partial\theta}((1+\frac{r}{R}\cos\theta)D_g\frac{\partial\Phi_g}{\partial\theta})+(1+\frac{r}{R}\cos\theta)\Sigma_g^r\Phi_g=(1+\frac{r}{R}\cos\theta)\overline{S}_g$$

where S_g = three terms representing sources on the right hand side of equation (29).

If now one defines

$$D_g' = (1 + r/R\cos\theta)\, D_g$$

$$\Sigma^{r'}{}_g = (1 + r/R\cos\theta)\, \Sigma^r{}_g$$

$$S_g' = (1 + r/R\cos\theta)\, S_g$$

Then the above equation becomes

$$-\frac{1}{r}\frac{\partial}{\partial r}(rD'_g\frac{\partial\Phi_g}{\partial r})-\frac{1}{r^2}\frac{\partial}{\partial\theta}(D'_g\frac{\partial\Phi_g}{\partial\theta})+\Sigma_g^{r'}\Phi_g=\overline{S}'_g$$

This equation is just the cylindrical diffusion equation with a modified diffusion coefficient, removal cross section, and source function. Pomraning and Stevens'[12] proposed the possibility of using an existing r-θ neutron diffusion code and redefining the diffusion coefficient, removal cross section, and source function to model the curvature of the torus naturally. This method was the original idea that suggested the topic of this research ; however, this approach was later abandoned in favor of direct finite differencing of the toroidal diffusion equation.

II.C. Finite Difference Approximation to the Diffusion Equation in Cylindrical Coordinates

To develop a finite difference approximation for the cylindrical diffusion equation(with axial symmetry), it is first necessary to integrate equation (28) over a small, arbitrary volume ΔV (see Figure 2-2). Thus,

$$-\int_{\Delta V}\frac{1}{r}\left[\frac{\partial}{\partial r}(rD_g\frac{\partial\Phi_g}{\partial r})+\frac{\partial}{\partial\theta}(\frac{D_g}{r}\frac{\partial\Phi_g}{\partial\theta})\right]dV+\int_{\Delta V}\left[\Sigma_g^r\Phi_g\right]dV \tag{30}$$

$$=\int_{\Delta V}\left[S_g^{ext}+\chi_g\sum_{g'=1}^{G}(\nu_{g'}\Sigma_{f_{g'}}\Phi_{g'})+\sum_{g'=1}^{g-1}(\Sigma_s(g\to g')\,\Phi_{g'})\right]dV$$

where the first term on the LHS of the equation is the leakage term, the second term on

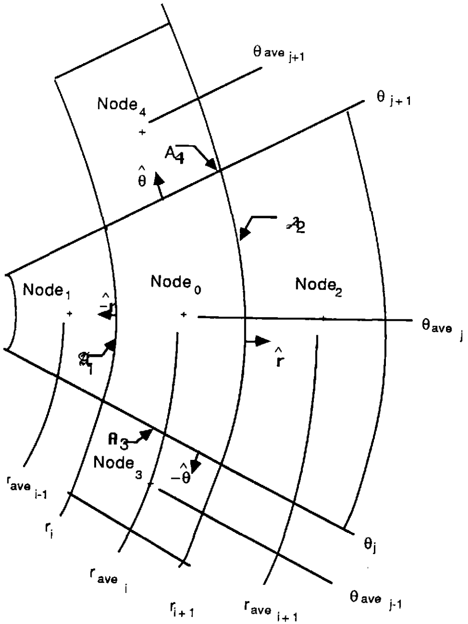


Figure 2-2. Finite Difference Coordinates Used in 2DB and 2DBTOR.

the LHS of the equation is the removal term, and the RHS represents the source terms including fission and scatter. Thus for the removal term,

$$\int_{\Delta V} \Sigma_g^r \Phi_g \, dV = \Sigma_{g_0}^r \Phi_{g_0} \quad (r_{i+1} - r_i)(\theta_{j+1} - \theta_j) \, r_{ave_i} \tag{31}$$

$$= \Sigma_{g_0}^r \Phi_{g_0} V_0$$

where Φ_{g_0} = flux associated with meshpoint o

and $\Sigma_{g_0}^r$ = removal cross section associated with meshpoint o.

Just as above, the source term on the RHS can be shown to have a V_o given by

$$V_o = (r_{i+1} - r_i)(\theta_{j+1} - \theta_j) \, r_{ave_i} . \tag{32}$$

The leakage term is changed to an integral over the surface area of the volume element, thus from the Divergence Theorem

$$-\int_{\Delta V} \vec{\nabla} \cdot D_g \vec{\nabla} \Phi_g \, dV = -\int_A D_g \vec{\nabla} \Phi_g \cdot \hat{n} \, dA \quad .$$

Using equation (24) for the $D_g \nabla \Phi_g$ term (where $d\Phi/dz = 0$)

$$-\int_A D_g \vec{\nabla} \Phi_g \cdot \hat{n} \, dA = -\int_A \left(D_g \frac{\partial \Phi_g}{\partial r} \hat{r} + \frac{D_g}{r} \frac{\partial \Phi_g}{\partial \theta} \vec{\theta} \right) \cdot \hat{n} \, dA \quad .$$

The normal vector, n, is r or θ for those area elements having normals in the positive r or θ, or increasing r or θ, respectively. In the same manner, n is -r or -θ for those elements in the -r or -θ directions. The area element corresponding to a normal in the ±r direction is rdθ, while the area element for a normal in the ±θ direction is dr. Thus for the 4 area elements

$$
\begin{aligned}
& -\int_A \left(D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}\right)\cdot \hat{n}\, dA \\
= & -\int_{A_1} \left(D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}\right)\cdot \hat{n}\, dA_1 \\
& -\int_{A_2} \left(D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}\right)\cdot \hat{n}\, dA_2 \\
& -\int_{A_3} \left(D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}\right)\cdot \hat{n}\, dA_3 \\
& -\int_{A_4} \left(D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}\right)\cdot \hat{n}\, dA_4
\end{aligned}
$$

Substituting for n and the dA's (see Figure 2-2) then the RHS becomes

$$= - \int_{\theta_j}^{\theta_{j+1}} (D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}) \cdot -\hat{r}\, r_i \, d\theta \Big|_{A_1}$$

$$- \int_{\theta_j}^{\theta_{j+1}} (D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}) \cdot \hat{r}\, r_{i+1} \, d\theta \Big|_{A_2}$$

$$- \int_{r_i}^{r_{i+1}} (D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}) \cdot -\hat{\theta} \, dr \Big|_{A_3}$$

$$- \int_{r_i}^{r_{i+1}} (D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta}) \cdot \hat{\theta} \, dr \Big|_{A_4}$$

Since $r \cdot \theta = \theta \cdot r = 0$ and $r \cdot r = \theta \cdot \theta = 1$, then the RHS simplifies to

$$= \int_{\theta_j}^{\theta_{j+1}} D_g \frac{\partial \Phi_g}{\partial r} r_i \, d\theta \Big|_{A_1} - \int_{\theta_j}^{\theta_{j+1}} D_g \frac{\partial \Phi_g}{\partial r} r_{i+1} \, d\theta \Big|_{A_2} + \int_{r_i}^{r_{i+1}} \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta} \, dr \Big|_{A_3} - \int_{r_i}^{r_{i+1}} \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta} \, dr \Big|_{A_4} \quad (33)$$

Since the partial derivatives of the flux will be obtained by differencing the two neighboring flux values, then letting k be the adjacent mesh point to mesh point o gives

$$\left.\frac{d\phi}{dx}\right| \approx \frac{\phi_k - \phi_0}{\Delta x}$$

where x = r or θ depending on the derivative being considered.

Then

$$\left.\int_{\theta_j}^{\theta_{j+1}} D_g \frac{\partial \Phi_g}{\partial r} r_i d\theta \right|_{A_1} = \left.\overline{D}_{g_1} \frac{\partial \Phi_g}{\partial r}\right|_{r_i} r_i(\theta_{j+1} - \theta_j)$$

$$\approx -\overline{D}_{g_1} \frac{\phi_{g_1} - \phi_{g0}}{r_{ave_i} - r_{ave_{i-1}}} r_i(\theta_{j+1} - \theta_j)$$

where Dg_k is defined to be (see Appendix B)

$$\overline{D}_{gk} = \frac{D_{g_o} D_{gk} (\Delta r_o + \Delta r_k)}{(D_{g_o} \Delta r_k + D_{g_k} \Delta r_o)} \quad \text{or} \quad \frac{D_{g_o} D_{gk} (\Delta \theta_o + \Delta \theta_k)}{(D_{g_o} \Delta \theta_k + D_{g_k} \Delta \theta_o)} \quad \text{for } k = 1, 2, 3, \text{ or } 4$$

and Dg_o = average D_g for volume element o,

Dg_k = average D_g for volume element k,

Δr_o = Δr for volume element o,

Δr_k = Δr for volume element k,

$\Delta\theta_o = \Delta\theta$ for volume element o, and

$\Delta\theta_o = \Delta\theta$ for volume element o.

Let

$$A_1 = r_i(\theta_{j+1} - \theta_j)$$
$$L_1 = r_{ave_i} - r_{ave_{i-1}}$$

Then

$$\int_{\theta_j}^{\theta_{j+1}} D_g \frac{\partial \Phi_g}{\partial r} r_i \, d\theta \approx -\overline{D}_{g1} \frac{\phi_{g1} - \phi_{go}}{L_1} A_1 \tag{34}$$

For r_i close to r_{i+1}, $\ln(r_{i+1}/r_i) \approx (r_{i+1} - r_i)$. Thus,

$$\ln\left(\frac{r_{i+1}}{r_i}\right) \approx r_{i+1} - r_i \quad ,$$

which when used in the third term in equation (33) gives

$$\int_{r_i}^{r_{i+1}} \frac{D_g}{r}\frac{\partial\Phi_g}{\partial\theta}\,dr = \overline{D}_{g3}\frac{\partial\Phi_g}{\partial\theta}\bigg|_{\theta_j} \ln r \bigg|_{r_i}^{r_{i+1}}$$

$$\approx -\overline{D}_{g3}\frac{\phi_{g3} - \phi_{g0}}{\theta_{ave_j} - \theta_{ave_{j-1}}}\,(r_{i+1} - r_i) \quad .$$

Let

$$A_3 = (r_{i+1} - r_i)$$
$$L_3 = \theta_{ave_j} - \theta_{ave_{j-1}} \quad ,$$

then

$$\int_{r_i}^{r_{i+1}} \frac{D_g}{r}\frac{\partial\Phi_g}{\partial\theta}\,dr \approx -\overline{D}_{g3}\frac{\phi_{g3} - \phi_{g0}}{L_3}A_3 \; . \tag{35}$$

The above process can be applied to the remaining terms of equation (33) to give

$$-\int_{\theta_j}^{\theta_{j+1}} D_g \frac{\partial \Phi_g}{\partial r} r_{i+1}\, d\theta \approx \overline{D}_{g_2} \frac{\phi_{g_2} - \phi_{g_0}}{r_{ave_{i+1}} - r_{ave_i}} r_{i+1} (\theta_{j+1} - \theta_j) \tag{36}$$

$$\approx -\overline{D}_{g_2} \frac{\phi_{g_2} - \phi_{g_0}}{L_2} A_2$$

and

$$-\int_{r_i}^{r_{i+1}} \frac{D_g \partial \Phi_g}{r\ \partial \theta} dr \approx \overline{D}_{g_4} \frac{\phi_{g_4} - \phi_{g_0}}{\theta_{ave_{j+1}} - \theta_{ave_j}} (r_{i+1} - r_i) \tag{37}$$

$$\approx -\overline{D}_{g_4} \frac{\phi_{g_4} - \phi_{g_0}}{L_4} A_4$$

where $L_2 = r_{ave\,i+1} - r_{ave\,i}$,

$A_2 = r_{i+1} (\theta_{j+1} - \theta_j)$,

$L_4 = \theta_{ave\,j+1} - \theta_{ave\,j}$,

and $A_4 = r_{i+1} - r_i$.

Combining all of the above terms back into equation (30)

$$- \sum_{k=1}^{4} \left[\bar{D}_{g_k} \left(\frac{\phi_{g_k} - \phi_{g_0}}{L_k} \right) A_k \right] + \Sigma^{r}_{g_0} \Phi_{g_0} V_0 \tag{38}$$

$$= \left[S^{ext}_{g_0} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_{g'}} \Phi_{g'_0}) + \sum_{g'=1}^{g-1} (\Sigma_{s_0}(g \rightarrow g') \, \Phi_{g'_0}) \right] V_0 \, .$$

II.D. Finite Difference Approximation to the Diffusion Equation in Toroidal Coordinates

In a similar manner as was used in the previous section, the diffusion equation in a toroidal coordinate system can be written in finite difference form. Using the Divergence Theorem to transpose the leakage term in the toroidal diffusion equation (equation (29)) from a volume to a surface integral and substituting for the $D_g \nabla \Phi_g$ term from equation (26), then

$$-\int\!\!\int_A \left[D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta} + \frac{D_g}{(R + r\cos\theta)}\frac{\partial \Phi_g}{\partial \phi}\hat{\phi} \right] \cdot \hat{n}\, dA + \int\!\!\int_V \left[\Sigma^r_g \Phi_g \right] dV \tag{39}$$

$$= \int\!\!\int_V \left[S_g^{ext} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_{g'}} \Phi_{g'}) + \sum_{g'=1}^{g-1} (\Sigma_s(g \rightarrow g')\, \Phi_{g'}) \right] dV$$

The leakage term in toroidal coordinates becomes (assuming axisymmetry)

$$-\int_A 2\pi (R + r\cos\theta) \left(D_g \frac{\partial \Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial \Phi_g}{\partial \theta}\hat{\theta} \right) \cdot \hat{n}\, dA \; .$$

where the $(R + r\cos\theta)$ term appears from multiplying by $(R + r\cos\theta)$ before taking $d\Phi/d\phi = 0$. As in the cylindrical form of the diffusion equation, n and the dA's are the same, so the only difference in evaluating the toroidal form of the diffusion equation is the $(R + r\cos\theta)$ terms. Thus from the above, the leakage term becomes

$$2\pi\int_A (R+r\cos\theta)\left(D_g\frac{\partial\Phi_g}{\partial r}\hat{r} + \frac{D_g}{r}\frac{\partial\Phi_g}{\partial\theta}\hat{\theta}\right)\cdot\hat{n}\,dA =$$

$$2\pi\int_{\theta_j}^{\theta_{j+1}} D_g(R+r\cos\theta)\frac{\partial\Phi_g}{\partial r}r_i\,d\theta\Bigg|_{A_1} - 2\pi\int_{\theta_j}^{\theta_{j+1}} D_g(R+r\cos\theta)\frac{\partial\Phi_g}{\partial r}r_{i+1}\,d\theta\Bigg|_{A_2}$$

$$+ 2\pi\int_{r_i}^{r_{i+1}} \frac{D_g}{r}(R+r\cos\theta)\frac{\partial\Phi_g}{\partial\theta}dr\Bigg|_{A_3} - 2\pi\int_{r_i}^{r_{i+1}} \frac{D_g}{r}(R+r\cos\theta)\frac{\partial\Phi_g}{\partial\theta}dr\Bigg|_{A_4}$$

$$\approx -2\pi R\,\overline{D}_{g1}\frac{\phi_{g1}-\phi_{g0}}{L_1}\left[r_i\left[(\theta_{j+1}-\theta_j)+\frac{r_i}{R}(\sin\theta_{j+1}-\sin\theta_{j+1})\right]\right]$$

$$-2\pi R\,\overline{D}_{g2}\frac{\phi_{g2}-\phi_{g0}}{L_2}\left[r_{i+1}\left[(\theta_{j+1}-\theta_j)+\frac{r_{i+1}}{R}(\sin\theta_{j+1}-\sin\theta_{j+1})\right]\right]$$

$$-2\pi R\,\overline{D}_{g3}\frac{\phi_{g3}-\phi_{g0}}{L_3}\left[(r_{i+1}-r_i)\left[\frac{\cos\theta_{ave_{j+1}}}{R}+1\right]\right]$$

$$-2\pi R\,\overline{D}_{g4}\frac{\phi_{g4}-\phi_{g0}}{L_4}\left[(r_{i+1}-r_i)\left[\frac{\cos\theta_{ave_j}}{R}+1\right]\right]$$

For the removal term

$$\int_V \Sigma_g^r \Phi_g \, dV = \Sigma_{g_0}^r \Phi_{g_0} \int_{r_i}^{r_{i+1}} \int_{\theta_j}^{\theta_{j+1}} 2\pi r (R + r\cos\theta) \, dr \, d\theta$$

$$= \Sigma_{g_0}^r \Phi_{g_0} 2\pi R \left[\left(\frac{r_{i+1}^2 - r_i^2}{2} \right) (\theta_{j+1} - \theta_j) + \frac{r_{i+1}^3 - r_i^3}{3R} (\sin\theta_{j+1} - \sin\theta_j) \right]$$

$$= \Sigma_{g_0}^r \Phi_{g_0} 2\pi R \left[(r_{i+1} - r_i) \, r_{ave_i} (\theta_{j+1} - \theta_j) + \frac{r_{i+1}^3 - r_i^3}{3R} (\sin\theta_{j+1} - \sin\theta_j) \right]$$

$$= \Sigma_{g_0}^r \Phi_{g_0} 2\pi R \, V_o$$

As in the cylindrical coordinate system diffusion equation, V_O for the source term is the same as the above term for V_O.

Let

$$A_1 = \left[r_i \left[(\theta_{j+1} - \theta_j) + \frac{r_i}{R} (\sin\theta_{j+1} - \sin\theta_{j+1}) \right] \right]$$

$$A_2 = \left[r_{i+1} \left[(\theta_{j+1} - \theta_j) + \frac{r_{i+1}}{R} (\sin\theta_{j+1} - \sin\theta_{j+1}) \right] \right]$$

$$A_3 = \left[(r_{i+1} - r_i) \left[\frac{\cos\theta_{ave_{j+1}}}{R} + 1 \right] \right]$$

$$A_4 = \left[(r_{i+1} - r_i) \left[\frac{\cos\theta_{ave_j}}{R} + 1 \right] \right]$$

Using the above, the toroidal diffusion equation in finite difference form after using equation (39) and dividing through by $2\pi R$ becomes

$$
-\sum_{k=1}^{4}\left[\overline{D}_{g_k}\left(\frac{\phi_{g_k}-\phi_{g_0}}{L_k}\right)A_k\right]+\Sigma^{r}_{g_0}\Phi_{g_0}V_0
$$

$$
=\left[S^{ext}_{g_0}+\chi_g\sum_{g'=1}^{G}(\nu_{g'}\Sigma_{f_{g'_0}}\Phi_{g'_0})+\sum_{g'=1}^{g-1}(\Sigma_{s_0}(g\rightarrow g')\,\Phi_{g'_0})\right]V_0 .
$$

Thus, the finite difference approximation to the diffusion equation in any coordinate system results when appropriate area and volume elements are used. The differences between the cylindrical and toroidal coordinate system area and volume elements are listed in Table 2-1. By changing the area and volume elements to those for toroidal coordinates, a standard neutron diffusion code can be generalized to solve the multi-group diffusion equations in toroidal coordinates and in practice this is easier than the redefinition of cross sections proposed by Pomraning and Stevens.

II.E. Analytical Solutions

In order to insure that the above methodology was implemented correctly, analytical solutions for the flux profiles of an elementary case were obtained. Pomraning and Stevens'[12] presented the analytical solution for the diffusion of neutrons due to a line source in a homogeneous cylindrical medium of radius, a (see Appendix A). This cylinder did not have a central void. Approximate solutions were then derived for the

Table 2-1. Area and Volume Elements in Cylindrical and Toroidal Coordinates.

	Cylindrical	Toroidal
A_1	$r_i(\theta_{j+1} - \theta_j)$	$[r_i[(\theta_{j+1} - \theta_j) + \frac{r_i}{R}(\sin\theta_{j+1} - \sin\theta_{j+1})]]$
A_2	$r_{i+1}(\theta_{j+1} - \theta_j)$	$[r_{i+1}[(\theta_{j+1} - \theta_j) + \frac{r_{i+1}}{R}(\sin\theta_{j+1} - \sin\theta_{j+1})]]$
A_3	$(r_{i+1} - r_i)$	$[(r_{i+1} - r_i)[\frac{\cos\theta_{ave_{j+1}}}{R} + 1]]$
A_4	$r_{i+1} - r_i$	$[(r_{i+1} - r_i)[\frac{\cos\theta_{ave_j}}{R} + 1]]$
V_o	$(r_{i+1} - r_i)(\theta_{j+1} - \theta_j) r_{ave_i}$	$[(r_{i+1} - r_i) r_{ave_i}(\theta_{j+1} - \theta_j) + \frac{r_{i+1}^3 - r_i^3}{3R}(\sin\theta_{j+1} - \sin\theta_j)]$

flux profile within a torus of major radius, R, and minor radius, a. The analytical result for the cylindrical problem is

$$\Phi(r) = \frac{1}{2\pi}\left[K_o\left(\frac{k\,r}{a}\right) - \frac{K_o(k)}{I_o(k)} I_o\left(\frac{k\,r}{a}\right)\right] \tag{40}$$

and the approximate result for the toroidal problem is

$$\Phi(r,\theta) \approx \frac{1}{2\pi}\left[K_o\left(\frac{k\,r}{a}\right) - \frac{K_o(k)}{I_o(k)} I_o\left(\frac{k\,r}{a}\right)\right]\left(1 - \frac{r\,R\cos\theta}{2}\right) + O\left(\left(\frac{a}{R}\right)^2\right) \tag{41}$$ [12]

where $k = (\Sigma_a/D)^{1/2}$; K_o and I_o are the zero order modified Bessel functions of the first and second kind, respectively; and $O((a/R)^2)$ denotes an error term of order $(a/R)^2$. For the above solutions, the only source is a line source of unit strength at $r=0$[12].

III. RESULTS

III.A 2DBTOR Verification and Infinite Cylinder Problems

In order to develop and verify 2DBTOR, modifications had to be made to 2DB, as described previously in the theory section. In addition, 2DB was tested to insure that it was operating correctly. This was done by comparing 2DB's results to both analytical and numerical flux solutions for several types of problems. ANISN was used to obtain numerical results to verify one-dimensional results from 2DB; Pomraning and Stevens' solution for a solid, homogeneous torus with a line source in the center provided an analytical verification in the limit as the aspect ratio of the torus became large (e.g., the torus approximated an infinite cylinder).[12]

The first part of the investigation was to compute, using 2DB, the flux in an infinite, homogeneous cylinder due to a line source. The radius of the cylinder was 300cm. Graphite was chosen as the material for the cylinder, since graphite has good neutron scattering properties ($\Sigma_s >> \Sigma_a$) and thus should prove to give good results for the flux profile when diffusion theory is used. A source density given by

$$S_v = \frac{D}{\pi r_v^2} \qquad (42)^{16}$$

where S_v = source density (n/cm^3),

$D = 1/3\Sigma_{tr}$ = diffusion coefficient,

Σ_{tr} = macroscopic transport cross section,

and r_V = radius of the area for the source,

was input into 2DB for a small radius (r_V = 6cm) which would approximate the line source of equation (40) from the theory (see Appendix A[16]). ANISN was used to collapse the 30 group graphite cross sections to one group using S_2 angular quadrature with the P_0 matrix from the CLAW-IV neutron cross section library. After collapsing to one group, 2DBXPROC, a cross section processor to manipulate the cross sections from ANISN's format into 2DB's format (see Appendix C for a listing of 2DBXPROC), was used so that 2DB could employ the one group cross sections for graphite. Solutions for the flux away from the source for both the analytical calculation and the 2DB run gave results that were in excellent agreement, after errors for both the volume and area elements were corrected (see Appendix D). The one group cross section set for graphite was then input to ANISN with the same conditions as above. Solutions of the flux from ANISN were also in excellent agreement with 2DB.

After studying the infinite, homogeneous cylinder, computations were made to determine the flux due to a uniformly distributed source of 14 Mev neutrons (see Appendix E) in an infinite homogeneous cylinder with a central void. The source radius was 150 cm and the inner and outer radii of the medium were 200cm and 300cm, respectively. ANISN was again used to collapse the 30 group cross section set for graphite to one group. In order to model the central void region, fictitious cross sections were first input into 2DB such that the absorption cross section was zero and the scattering cross section was equal to the transport cross section of graphite. This method proved to give somewhat erroneous results, however, since the value for the diffusion coefficient in the vacuum region was arbitrary. To alleviate this problem, a sufficiently thin source equivalent to the plasma source was placed at the inner edge of the annulus (see Appendix F) and the problem was re-run. This quasi-albedo boundary condition overcame the need for a fictitious scattering cross section in the void. Re-runs of the above problem gave 2DB and ANISN results which were in good agreement.

To complete the test of 2DB, some of the above problems were re-run with more than one energy group. This served to verify that 2DB's use of the downscattering cross sections was being performed correctly. Each problem was solved as before, except that 9 groups were used instead of one. Again the cross sections were obtained by collapsing the 30 group P_0 matrix using ANISN. Each solution of the flux was correct for each problem done previously for graphite. Since fusion problems would involve (n,2n) and (n,n') reactions, it was decided to further test 2DB by using Nb-93, which has a significant (n,2n) reaction at high energy for the problems previously studied. The results obtained using Nb-93 in 2DB for the two cases above (with and without central voids) were discovered to be quite different from those of ANISN. An investigation of 2DB's source code was therefore performed to identify the reason for the differences. It was then discovered that 2DB erroneously computed the downscattering (see Appendix D). After correcting 2DB to compute the proper downscatter contribution, the Nb-93 problem was re-run with a 9-group cross section set, and the results were in good agreement with those obtained using ANISN.

III.B 2DBTOR Verification and Toroidal Problems

After making appropriate changes, as described earlier in this thesis (i.e, change the volume and area elements), to change 2DB to compute the flux for toroidal geometry (see Appendices G and H), a new code was produced which was called 2DBTOR. Using the same conditions stated previously for an infinite homogeneous graphite cylinder with one-group cross sections and without a central void, 2DBTOR was run for aspect ratios of 3 and 5 and compared to Pomraning and Stevens' solutions[12] for the same aspect ratio. The solutions for the flux were in excellent agreement. These solutions for both aspect ratios were then compared to the infinite cylinder solutions at radii of 150cm and 250cm from $\theta = 0$ to 2π. The results are shown in Figures 3-1 and 3-2 for 150cm and 250cm, respectively. As can be seen from Figures 3-1 and 3-2, the flux has a minimum at the outside part of the torus ($\theta = 0$) and increased up to the inner part ($\theta = \pi$), where the flux was a maximum. This was an expected result since the area of the inner portion of the torus is smaller than the outer portion. In addition, as the aspect ratio increases the solution approaches that of the infinite cylinder.

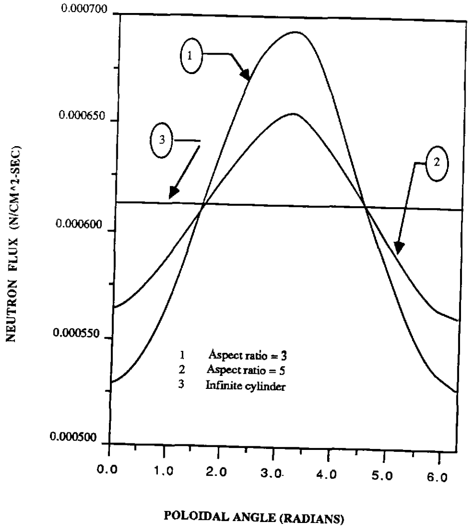


Figure 3-1. Flux Versus Poloidal Angle at a Radius of 150 cm for Several Aspect Ratios.

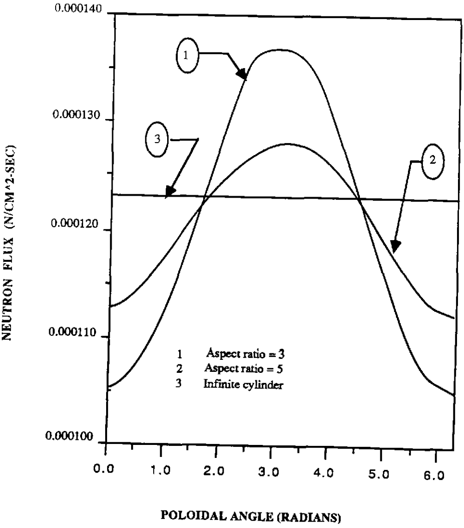


Figure 3-2. Flux Versus Poloidal Angle at a Radius of 250 cm for Several Aspect Ratios.

III.C Standard Blanket Solutions

III.C.1 Infinite Cylinder

The blanket used for this analysis is shown in Figure 3-3, and it is the so-called 'standard' blanket that was formulated as a benchmark for neutronic calculations.[17] The materials used and their atom densities are given in Table 3-1.[17] From Figure 3-3 it can be seen that the blanket consists of 10 zones, or regions, which contain one of 3 mixtures(except for the first 2 zones which are the vacuum and plasma). These are denoted A, B, or C. Each mixture has one or more materials including Li-6, Li-7, C-12, and Nb-93. When lithium is present in a mixture, the medium is medium is assumed to be homogeneous with volume fractions of 94% lithium and 6% niobium. The lithium serves as both a coolant and tritium producer, while the niobium provides the structural function. The plasma has a radius of 150cm and occupies zone 1. A vacuum region is located from 150cm to 200cm and occupies zone 2. The blanket extends from 200cm to 300cm consisting of the first wall (200-200.5cm), a tritium production region (200.5-203.5cm), the second wall (203.5-204cm), 3 tritium production regions of 20cm thickness each (204-264cm), a carbon reflector (264-294cm), and a final tritium production region (294-300cm).

After setting up the standard blanket problem to run on ANISN, the 30-group cross sections in CLAW-IV for all the materials were collapsed to 9-group cross sections, with a P_0 Legendre expansion and S_2 quadrature to represent a diffusion theory calculation. 50 mesh intervals were used in this computation (see Figure 3-3). The 9-group cross section set was unable to produce satisfactory results for the tritium

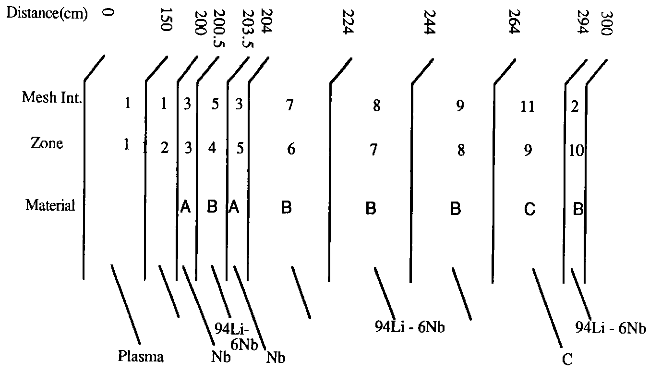


Figure 3-3. Standard Blanket Configuration.

Table 3-1. Standard Blanket Constituents.

Material Code Letter (from Figure 3-3)	Constituents	Atom density (atoms/cm^3 x 10^{-24})
A	Nb-93	0.05556
B	Nb-93	0.003334
	Li-6	0.003234
	Li-7	0.04038
C	C-12	0.0804

breeding ratio (TBR = tritium production rate/ source neutron rate) for both T6 (TBR from Li-6) and T7 (TBR from Li-7). When the 9-group cross section set was input back into ANISN for comparison to the results of the 30-group set, differences in the corresponding zone TBR's were detected which were deemed to be unacceptable. To overcome this problem, it was decided to use the 30-group, uncollapsed, cross section set in 2DB and 2DBTOR for TBR computations. Again as in ANISN, 50 mesh intervals were used in both 2DB and 2DBTOR. The TBR results for 2DB and ANISN were compared in the zones of interest (i.e., zones 4,6,7,8, and 10 where lithium was present). The results of the 2DB and ANISN computations are given in Table 3-2 for the above 5 zones. It is seen from Table 3-2 that there is agreement between the results from the codes, especially in the T6 values. The T7 values are acceptable in the inner zones but have significant error in the outer zones. Likewise, the total TBR in the entire blanket is in closest agreement for T6, and differs by less than 5% for T7. A comparison of the individual group fluxes showed a similarly acceptable agreement between 2DB and ANISN. All of the above implies that 2DB can now be used successfully to perform fusion scoping calculations in one-dimensional geometry.

III.C.2 Comparison of Torus and Infinite Cylinder Models

Since 2DB was found to produce adequate results for the standard blanket compared to the ANISN results, a comparison of the results from both 2DBTOR and

Table 3-2. Radially Averaged Tritium Breeding Ratio's for 2DB and ANISN, Infinite Cylinder Case.

Zone Number	Zone Radii (cm)	T6, reaction rate per source neutron rate from Li-6		T7, reaction rate per source neutron rate from Li-7	
		ANISN	2DB	ANISN	2DB
4	200.5-203.5	0.05043	0.05776	0.08191	0.10119
6	204.0-224.0	0.30399	0.34143	0.29809	0.33643
7	224.0-244.0	0.24598	0.25690	0.12189	0.09031
8	244.0-264.0	0.29518	0.27286	0.05050	0.02108
10	294.0-300.0	0.04923	0.03955	0.00093	0.00009
Total		0.94482	0.96850	0.55331	0.54910

2DB was made next. The 2DBTOR analysis was run in both toroidal (aspect ratios = 3 and 5) and cylindrical geometry. To begin the analysis, zones were chosen as in the previous infinite cylinder computation, so that radially averaged TBR's could be computed. In Tables 3-3 and 3-4, the TBR results for aspect ratios of 3 and 5 versus the infinite cylinder results were compared for T6 and T7, respectively. As can be seen there, the total T6 and T7 were about the same for both the toroidal and infinite cylinder calculations. This suggested the idea that if one were only interested in global reaction rates (e.g., not the outside versus inside reaction rates for the torus), then a one-dimensional neutron transport or diffusion code would be adequate for determining these quantities.

To investigate the local effects introduced by the toroidal geometry, the standard blanket was further sub-divided into ten zones of 36^o each in the angular or poloidal direction. This resulted in 100 zones for the entire torus cross section (10 radially x 10 poloidally). The infinite cylinder T6 and T7 results are presented in Tables 3-5 and 3-6 for an aspect ratio of 3 and in Tables 3-7 and 3-8 for an aspect ratio of 5. Only the 5 zones from 0 to π are presented since the torus is symmetric about its axis (see Figure 2-1). From these tables, it was noted that the TBR was largest at the outer part of the torus ($\theta = 0^o$ to 36^o) and decreased to a minimum value at the inner part of the torus ($\theta = 144^o$ to 180^o). The effect of different aspect ratios was also seen by comparing the corresponding zone rates -- e.g., the 0^o to 36^o zone rate for an aspect ratio of 3 was about 7% larger than the 0^o to 36^o zone rate for an aspect ratio of 5, while the 144^o to 180^o zone rate was about 10% smaller for an aspect ratio of 3. This is a desirable result in a practical sense since the tritium on the outside part of the

Table 3-3. Radially Averaged Tritium Breeding Ratio's for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 3).

Zone Number	Zone Radii (cm)	T6, reaction rate per source neutron rate from Li-6		T7, reaction rate per source neutron rate from Li-7	
		2DBTOR	2DB	2DBTOR	2DB
4	200.5-203.5	0.05767	0.05776	0.10119	0.10119
6	204.0-224.0	0.34095	0.34143	0.33643	0.33643
7	224.0-244.0	0.25667	0.25690	0.09031	0.09031
8	244.0-264.0	0.27238	0.27286	0.02107	0.02108
10	294.0-300.0	0.03950	0.03955	0.00009	0.00009
Total		0.96717	0.96850	0.54909	0.54910

Table 3-4. Radially Averaged Tritium Breeding Ratio's for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 5).

Zone Number	Zone Radii (cm)	T6, reaction rate per source neutron rate from Li-6		T7, reaction rate per source neutron rate from Li-7	
		2DBTOR	2DB	2DBTOR	2DB
4	200.5-203.5	0.05781	0.05776	0.10114	0.10119
6	204.0-224.0	0.34167	0.34143	0.33619	0.33643
7	224.0-244.0	0.25690	0.25690	0.09029	0.09031
8	244.0-264.0	0.27286	0.27286	0.02107	0.02108
10	294.0-300.0	0.03950	0.03955	0.00009	0.00009
Total		0.96874	0.96850	0.54878	0.54910

Table 3-5. Zone by Zone Averaged T6 for 2DBTOR Versus 2DB, Torus Case (Aspect Ratio = 3). Angle Measured in Degrees.

Zone	T6						
Number	2DB	0-36	36-72	72-108	108-144	144-180	0-360
4	0.05776	0.00688	0.00644	0.00503	0.00459	0.00459	0.05504
6	0.34143	0.04091	0.03824	0.03393	0.02960	0.02691	0.33914
7	0.25690	0.03110	0.02898	0.02555	0.02209	0.01994	0.25530
8	0.27286	0.03348	0.03107	0.02714	0.02319	0.02073	0.27123
10	0.03955	0.00495	0.00456	0.00394	0.00331	0.00292	0.03935
Total	0.96850	0.11732	0.10929	0.09559	0.08278	0.07509	0.96006

Table 3-6. Zone by Zone Averaged T7 for 2DBTOR Versus 2DB,

Torus Case (Aspect Ratio = 3). Angle Measured in Degrees.

Zone	T7						
Number	2DB	0-36	36-72	72-108	108-144	144-180	0-360
4	0.10119	0.01221	0.01141	0.01012	0.00884	0.00804	0.11240
6	0.33643	0.04074	0.03805	0.03364	0.02926	0.02655	0.33648
7	0.09031	0.01103	0.01027	0.00904	0.00780	0.00703	0.09034
8	0.02108	0.00260	0.00241	0.00211	0.00181	0.00162	0.02108
10	0.00009	0.00001	0.00001	0.00001	0.00001	0.00001	0.00009
Total	0.54910	0.06659	0.06215	0.05492	0.04772	0.04325	0.54924

Table 3-7. Zone by Zone Averaged T6 for 2DBTOR Versus 2DB,

Torus Case (Aspect Ratio = 5). Angle Measured in Degrees.

Zone	T6						
Number	2DB	0-36	36-72	72-108	108-144	144-180	0-360
4	0.05776	0.00642	0.00617	0.00576	0.00535	0.00510	0.05758
6	0.34143	0.03812	0.03655	0.03402	0.03152	0.03000	0.34043
7	0.25690	0.02888	0.02762	0.02560	0.02360	0.02237	0.25612
8	0.27286	0.03095	0.02950	0.02719	0.02491	0.02347	0.27204
10	0.03955	0.00454	0.00431	0.00395	0.00358	0.00335	0.03946
Total	0.96850	0.11732	0.10929	0.09559	0.08278	0.07509	0.96563

Table 3-8. Zone by Zone Averaged T7 for 2DBTOR Versus 2DB,

Torus Case (Aspect Ratio = 5). Angle Measured in Degrees.

Zone	T7						
Number	2DB	0-36	36-72	72-108	108-144	144-180	0-360
4	0.10119	0.01136	0.01089	0.01012	0.00935	0.00887	0.10117
6	0.33643	0.03786	0.03624	0.03362	0.03100	0.02938	0.33619
7	0.09031	0.01023	0.00977	0.00903	0.00829	0.00783	0.09030
8	0.02108	0.00240	0.00229	0.00211	0.00193	0.00182	0.02108
10	0.00009	0.00001	0.00001	0.00001	0.00001	0.00001	0.00009
Total	0.54910	0.06186	0.05920	0.05489	0.05058	0.04800	0.54882

torus is probably more accessible than that of the inner part. In addition, the TBR computed in the top zone (72° to 108°) was about the same for both aspect ratios. It also indicates that the toroidal TBR depended on volumetric effects, since an area on the outer part of the torus corresponds to a larger volume than an area on the inner part.[18]

D. L. Chapin of the Princeton Plasma Physics Laboratory presented the above idea concerning volumetric effects in a topical report.[18] From Table 2-1, the volume occupied by a specified zone is

$$V_t = V_c\left[1 + \frac{r_{i+1}^3 - r_i^3}{3R(r_{i+1} - r_i)\, r_{ave_i}(\theta_{j+1} - \theta_j)}(\sin\theta_{j+1} - \sin\theta_j)\right] \tag{43}$$

where V_t = volume of the torus,

and $V_c = (r_{i+1} - r_i) r_{avei} (\theta_{j+1} - \theta_j)$ = volume of the cylinder with the same zone as the torus.

Thus, a volume factor , F_V, can be defined as

$$F_v \equiv \frac{\text{volume of the torus}}{\text{volume of the cylinder}} = \left[1 + \frac{r_{i+1}^3 - r_i^3}{3R(r_{i+1} - r_i)\, r_{ave_i}(\theta_{j+1} - \theta_j)}(\sin\theta_{j+1} - \sin\theta_j)\right] \quad (44)^{18}$$

When F_V is multipied by the TBR given by the cylinder, good agreement is expected between the torus TBR values and the volume corrected TBR values of the cylinder according to Chapin.[18] Tables 3-9 through 3-12 show the differences between using volume corrections on the cylinder T6 and T7 for aspect ratios of 3 and 5 and then comparing to the torus's T6 and T7. This provides another check on the validity of 2DBTOR for doing toroidal scoping calculations, since Chapin's results were produced using a Monte Carlo code, which provides very accurate analysis for toroidal geometry. Since the TBR results from Tables 3-9 through 3-12 were in good agreement, it is apparent that 2DBTOR worked quite well for doing toroidal geometry scoping computations. Of course, the real test and usefulness of 2DBTOR will be for those designs which are not poloidally symmetric like the standard blanket used here.

Table 3-9. T6 for 2DBTOR Versus Volume Corrected T6 for 2DB, Torus Case (Aspect Ratio = 3). Angle Measured in Degrees.

Zone	T6						
Number	2DBTOR	0-36	36-72	72-108	108-144	144-180	0-360
4	0.05504	0.00699	0.00653	0.00578	0.00503	0.00456	0.05776
6	0.33914	0.04174	0.03884	0.03414	0.02945	0.02654	0.34143
7	0.25530	0.03194	0.02955	0.02569	0.02183	0.01944	0.25691
8	0.27123	0.03449	0.03174	0.02729	0.02283	0.02008	0.27286
10	0.03935	0.00518	0.00471	0.00396	0.00320	0.00273	0.03955
Total	0.96006	0.12034	0.11137	0.09685	0.08233	0.07336	0.96850

Table 3-10. T7 for 2DBTOR Versus Volume Corrected T7 for 2DB, Torus Case (Aspect Ratio = 3). Angle Measured in Degrees.

Zone	T7						
Number	2DBTOR	0-36	36-72	72-108	108-144	144-180	0-360
4	0.11240	0.01224	0.01143	0.01012	0.00881	0.00799	0.10119
6	0.33648	0.04113	0.03827	0.03364	0.02901	0.02615	0.33643
7	0.09034	0.01123	0.01039	0.00903	0.00767	0.00683	0.09031
8	0.02108	0.00267	0.00245	0.00211	0.00176	0.00155	0.02108
10	0.00009	0.00001	0.00001	0.00001	0.00001	0.00001	0.00009
Total	0.54924	0.06728	0.06256	0.05491	0.04726	0.04254	0.54910

Table 3-11. T6 for 2DBTOR Versus Volume Corrected T6 for 2DB, Torus Case (Aspect Ratio = 5). Angle Measured in Degrees.

Zone	T6						
Number	2DBTOR	0-36	36-72	72-108	108-144	144-180	0-360
4	0.05758	0.00650	0.00623	0.00578	0.00533	0.00505	0.05776
6	0.34043	0.03870	0.03696	0.03414	0.03133	0.02958	0.34143
7	0.25612	0.02944	0.02801	0.02569	0.02337	0.02194	0.25691
8	0.27204	0.03161	0.02996	0.02729	0.02461	0.02296	0.27286
10	0.03946	0.00469	0.00441	0.00396	0.00350	0.00322	0.03955
Total	0.96563	0.11095	0.10556	0.09685	0.08814	0.08275	0.96850

Table 3-12. T7 for 2DBTOR Versus Volume Corrected T7 for 2DB,

Torus Case (Aspect Ratio = 5). Angle Measured in Degrees.

Zone	T7						
Number	2DBTOR	0-36	36-72	72-108	108-144	144-180	0-360
4	0.10117	0.01139	0.01091	0.01012	0.00933	0.00884	0.10119
6	0.33619	0.03814	0.03642	0.03364	0.03089	0.02915	0.33643
7	0.09030	0.01035	0.00985	0.00903	0.00822	0.00771	0.09031
8	0.02108	0.00244	0.00231	0.00211	0.00190	0.00177	0.02108
10	0.00009	0.00001	0.00001	0.00001	0.00001	0.00001	0.00009
Total	0.54882	0.06233	0.05950	0.05491	0.05032	0.04749	0.54910

IV. SUMMARY AND CONCLUSIONS

The objective of the research performed here was to produce a scoping code that could be used for fusion reactor design. To this end, the present research initially explored a technique proposed by Pomraning and Stevens[12], in which a toroidal geometry diffusion problem is cast into cylindrical (r-θ) form by a spatially dependent redefinition of the diffusion coefficient, absorption cross-section, and extraneous source function. This idea suggested the approach of a direct finite differencing of the toroidal diffusion equation. Direct finite differencing proved to be more advantageous for incorporation into a computer program and to allow the curvature of the torus to be accounted for naturally.

The direct finite differencing approach was programmed into an existing two-dimensional(x-y, r-z, r-θ, triangular), multi-group neutron diffusion code, 2DB[10], that had previously been converted to execute on the IBM-AT. Neutronic scoping calculations relevant to fusion reactor design were then performed in a micro-computer environment. This modified code was called 2DBTOR.

To verify that 2DBTOR was operating correctly, comparisons were made to both analytical and numerical solutions for several types of problems. ANISN and 2DB were used to verify and compare the solutions obtained from 2DBTOR. It was also shown that as the aspect ratio approached infinity (e. g., the major radius became large) the 2DBTOR solution approached the solution for that of 1-D cylindrical geometry. After verifying the solution for a large major radius, the errors associated with using a non-toroidal scoping code were examined versus 2DBTOR. Neutron cross-sections for a benchmark problem were input into 2DBTOR and the output was compared to that from ANISN. A method proposed by Price and Chapin[18], that used volume correction factors to compute the reaction rates in the benchmark blanket, was utilized to provide a means of checking 2DBTOR's results versus those given by a Monte Carlo code. It was also noted that 2DBTOR would also make possible the calculation of depletion in the fusion blanket, which was a unique advantage of the new program, 2DBTOR.

In future versions of the 2DBTOR program, it is anticipated that the central vacuum should be modelled through an internal boundary condition. A separate void streaming calculation will be used to define the internal boundary condition by specifying the neutron flux to current ratio as a function of position along the vacuum wall.[19] Improved modelling of the central void region will be required if 2DBTOR is to prove to be an attractive program for blanket scoping calculations.

REFERENCES

1. R. A. Gross, <u>Fusion Energy</u>, (John Wiley & Sons, New York, New York ,1984), p. 2.

2. R. A. Gross, <u>Fusion Energy</u>, (John Wiley & Sons, New York, New York ,1984), p. 16.

3. G. A. Miley, <u>Fusion Energy Conversion</u>, (John Wiley & Sons, New York, New York ,1970), p. 250.

4. T. J. Dolan, <u>Fusion Research - Volume 1</u>, (Pergamon Press, New York, New York ,1982), p. 29.

5. R. A. Gross, <u>Fusion Energy</u>, (John Wiley & Sons, New York, New York ,1984), p. 25.

6. D. K. Parsons, "ANISN/PC Manual," EGG-2500, Idaho National Engineering Laboratory Report, Idaho Falls, Idaho ,1987.

7. K. D. Lathrop and F. W. Brinkley, "TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two Dimensional Transport," LA-4848-MS, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1973.

8. J. W. Davidson et al., "Trism: A Two-Dimensional, Finite-Element, Discrete-Ordinates Transport Code with Deterministic Streaming," LA - 7835 - M, Los Alamos National Laboratory, Los Alamos, New Mexico.

9. Los Alamos Monte Carlo Group, "MCNP: A General Monte Carlo Code for Neutron and Photon Transport," Version 2B, LA-7396-M, Los Alamos National Laboratory,Los Alamos, New Mexico, 1981.

10. W. W. Little, Jr. and R. W. Hardie, "2DB User's Manual," BNWL - 831, Pacific Northwest Laboratory, Richland, Washington, 1968.

11. R. J. Barrett and R. E. MacFarlane, "Coupled Neutron and Photon Cross Sections for Transport Calculations," LA-7808-MS, Informal Report, Los Alamos Scientific Laboratory,Los Alamos, New Mexico ,1979.

12. G. C. Pomraning and C. A. Stevens, "Transport and Diffusion Equations in Toroidal Geometry," Nuclear Science and Engineering: 55,359-367 (1974).

13. J. R. Lamarsh, Introduction to Nuclear Reactor Theory, (Addison Wesley Publishing Company, Reading, Massachusetts, 1972), pp. 129 - 131.

14. J. J. Duderstadt and L. J. Hamilton, Nuclear Reactor Analysis, (John Wiley & Sons, New York, New York ,1976), p. 61.

15. T. A. Parish, Personal notes on the ∇ operator, March 1988.

16. T. A. Parish, Personal notes on the plasma source calculation, May 1988.

17. D. Steiner, "Analyses of a Bench-Mark Calculation of Tritium Breeding in a Fusion Reactor Blanket," ORNL-TM-4177, Oak Ridge National Laboratory Report, Oakridge , Tennessee ,1973.

18. W. G. Price and D. L. Chapin, "Comparative Analysis of a Fusion Reactor Blanket in Cylindrical and Toroidal Geometry Using Monte Carlo," MATT-1102, Princeton Plasma Physics Report, Princeton , New Jersey ,1975.

19. W. G. Price and D. L. Chapin, "A Comparison of the D-T Neutron Wall Load Distributions in Several Tokamak Fusion Reactor Designs," MATT-1060, Princeton Plasma Physics Report, Princeton , New Jersey ,1974.

APPENDIX A

POMRANING AND STEVENS' ANALYTICAL SOLUTION

For an infinite line source at the center of a non-fissioning medium with radius, a, the monoenergetic neutron diffusion equation is given by the following (assuming that both the diffusion cofficient does not vary with position and the solution is only radially dependent):

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi}{\partial r}\right) - k^2\phi = Q \qquad \text{(A-1)}$$

where $k^2 = \frac{\Sigma_a}{D}$,

$Q = \frac{S}{D} = 0$ for $r>0$,

ϕ = the flux,

Σ_a = the macroscopic absorption cross section,

and D = the diffusion coefficient .

Solving equation (A-1) gives the following:

$$\phi(r) = AK_0(kr) + CI_0(kr) \qquad \text{(A-2)}$$

where both A and C are constants. Using the boundary condition at r=a where $\phi(a)=0$ allows C to be given as

$$C = -A\frac{K_o(ka)}{I_o(ka)}$$

which when substituted into equation (A-2) gives the following:

$$\phi(r) = A\left[K_0(kr) - \frac{K_o(ka)}{I_o(ka)} I_o(kr)\right] \quad . \tag{A-3}$$

In order to solve for A, one must use a source condition given by

$$\lim_{r\to 0} 2\pi r J(r) = S \tag{A-4}$$

where $\quad J(r) = -D\,\dfrac{\partial\phi(r)}{\partial r}$.

It can be shown that when equation (A-3) is subtituted into equation (A-4) the following results:

$$\lim_{r\to 0} rJ(r) = \frac{S}{2\pi D} = \frac{Q}{2\pi} \quad . \tag{A-5}$$

Since equation (A-4) can also be written as

$$\lim_{r\to 0} rJ(r) = \lim_{r\to 0} AkrK_1(kr) \tag{A-6}$$

which from Meghreblian and Holmes, Reactor Analysis, page 185, is equal to A, this implies that

$$A = \frac{Q}{2\pi} \quad . \tag{A-7}$$

When equation (A-7) is substituted into equation (A-3) the following results:

$$\phi(r) = \frac{Q}{2\pi}\left[K_0(kr) - \frac{K_o(ka)}{I_o(ka)}\,I_o(kr)\right] \quad . \tag{A-8}$$

In order to employ equation (A-8) for comparison to 2DB, it is necessary to determine a volumetric source that is equivalent to the line source used in deriving the flux solution. Thus, the line source is given by

$$S_l = S_v\,\pi r^2 \tag{A-9}$$

where S_l = the line source (n/cm-sec)

and S_v = the volumetric source (n/cm^3-sec).

For Q=1 and since $Q=S_l/D$, then this implies that $S_l = D$. When $S_l = D$ is substituted into equation (A-9), the resulting equation for S_v is given by the following:

$$S_v = \frac{D}{\pi r^2} \quad . \tag{A-10}$$

This is the Sv value that will be input to 2DB so that comparisons to the analytic solution can be made.

APPENDIX B

AVERAGE D_k SOLUTION

From continuity of current for two adjacent mesh points, k and k+1 (see Figure B-1), it can be shown that

$$\frac{D_k}{\frac{\delta r_k}{2}}\left(\phi^{1/2} - \phi_k\right) = \frac{D_{k+1}}{\frac{\delta r_{k+1}}{2}}\left(\phi_{k+1} - \phi^{1/2}\right) \quad . \tag{B-1}$$

where $\frac{\delta r_k}{2}$ = distance from mesh point k to mesh point 1/2

and $\frac{\delta r_{k+1}}{2}$ = distance from mesh point k+1 to mesh point 1/2 .

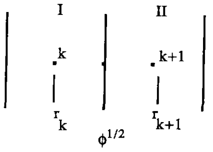


Figure B-1. Mesh points

Rearranging equation (B-1) gives the following:

$$\left[\frac{D_k}{\frac{\delta r_k}{2}} + \frac{D_{k+1}}{\frac{\delta r_{k+1}}{2}}\right]\phi^{1/2} = \frac{D_{k+1}}{\frac{\delta r_{k+1}}{2}}\phi_{k+1} + \frac{D_k}{\frac{\delta r_k}{2}}\phi_k \quad , \tag{B-2}$$

which when solved for $\phi^{1/2}$ yields

$$\phi^{1/2} = \frac{\dfrac{D_{k+1}}{\frac{\delta r_{k+1}}{2}}\phi_{k+1} + \dfrac{D_k}{\frac{\delta r_k}{2}}\phi_k}{\left[\dfrac{D_k}{\frac{\delta r_k}{2}} + \dfrac{D_{k+1}}{\frac{\delta r_{k+1}}{2}}\right]} \quad . \tag{B-3}$$

Substituting the RHS of equation (B-3) for the left-hand term in equation (B-1) gives the following:

$$J_k = \frac{D_{k+1}}{\frac{\delta r_{k+1}}{2}} \frac{D_k}{\frac{\delta r_k}{2}} \frac{1}{\left[\dfrac{D_k}{\frac{\delta r_k}{2}} + \dfrac{D_{k+1}}{\frac{\delta r_{k+1}}{2}}\right]} \left(\phi_{k+1} - \phi_k\right) \quad . \tag{B-4}$$

Further simplifying equation (B-4) gives

$$J_k = \frac{D_{k+1} D_k \left(\delta r_k + \delta r_{k+1}\right)}{\left[D_k\, \delta r_{k+1} + D_{k+1}\, \delta r_k\right]} \left(\frac{\phi_{k+1} - \phi_k}{r_{k+1} - r_k}\right) \tag{B-5}$$

where $\quad r_{k+1} - r_k = \dfrac{\delta r_k}{2} + \dfrac{\delta r_{k+1}}{2}$

Thus, D can be defined to be

$$\overline{D} = \frac{D_{k+1} D_k \left(\delta r_k + \delta r_{k+1}\right)}{\left[D_k \, \delta r_{k+1} + D_{k+1} \, \delta r_k\right]} \quad , \tag{B-6}$$

to yield

$$J_k = \overline{D} \left(\frac{\phi_{k+1} - \phi_k}{r_{k+1} - r_k} \right) \quad . \tag{B-7}$$

APPENDIX C

XPROC2DB LISTING

```
      DIMENSION A(30,46,4,4),B(30,46,4,4)
C      OPEN(UNIT=9,FILE='STDXSC.DAT',STATUS='OLD')
C      OPEN(UNIT=10,FILE='STDXSC2.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=9,FILE='XPROC30.DAT',STATUS='OLD')
      OPEN(UNIT=10,FILE='XPROC30.OUT',STATUS='UNKNOWN')

C  K = NUMBER OF MATERIALS
C  I = GROUP #
C  J = XSC TABLE #
C  NMAT=NUMBER OF MATERIALS
C  NGRUP=# OF GROUPS
C  NTAB=XSEC TABLE LENGTH
C  NSCAT=SCATTERING ORDER
      NSCAT=3
      NMAT=4
      NGRUP=30
      NTAB=46
      DO 30 K=1,NMAT
         DO 25 L=1,NSCAT+1
            DO 20 I=1,NGRUP
               READ(9,10) (A(I,J,K,L),J=1,NTAB)
10             FORMAT(1X/,9(5(E13.5)/),E13.5)
               A(I,14,K,L)=A(I,8,K,L)
```

```
16                    FORMAT(78X,I2)
20          CONTINUE
            READ(9,13)
13          FORMAT(1X)
25        CONTINUE
          READ(9,13)
30    CONTINUE
      DO 50 K=1,NMAT
      L=1
            DO 40 I=1,NGRUP
                        WRITE(10,10) (A(I,J,K,L),J=1,NTAB)
                        WRITE(10,99)
99                      FORMAT(3X,' T',2X)
40          CONTINUE
                        WRITE(10,16)K
                        WRITE(10,13)
50    CONTINUE
      K=4
      L=1
            DO 60 I=1,NGRUP
              A(I,17,K,L)=A(I,16,K,L)
60          CONTINUE
      K=4
```

```
      L=1
          DO 70 I=1,NGRUP
             DO 65 J=1,15
               A(I,J,K,L)=0.0
65           CONTINUE
70        CONTINUE
          DO 170 I=1,NGRUP
             DO 165 J=18,NTAB
               A(I,J,K,L)=0.0
165          CONTINUE
170       CONTINUE

      K=4
      L=1
          DO 260 I=1,NGRUP
                    WRITE(10,10) (A(I,J,K,L),J=1,NTAB)
                    WRITE(10,99)
260       CONTINUE
                    WRITE(10,16)K
                    WRITE(10,13)
      STOP
      END
```

APPENDIX D

2DB ERRATA

I. The area elements for r-θ geometry can be shown to be given by

$$A_r = r_{ave0}\,\Delta\theta_0 \qquad \text{(D-1)}$$

$$A_\theta = \Delta r_0 \qquad \text{(D-2)}$$

where A_r = radial area element and A_θ = axial area element for a volume element, 0. In the original 2DB code, the above area elements included a 2π factor. Since this is incorrect, the 2π terms were deleted (see Appendix H, subroutine INIT). For x-y geometry, the axial area element stays the same as above, but the radial element does not have a r_{ave0} term and thus, is given by

$$A_r = \Delta\theta_0 \; . \qquad \text{(D-3)}$$

Again, the area elements in 2DB included a 2π term. The 2π terms were deleted to cause 2DB to correctly solve the diffusion equation (see Appendix H, subroutine INIT).

II. For the downscattering term in the diffusion equation, 2DB uses the following:

$$XD = \Sigma_{tr} - \Sigma_s - \Sigma_a \tag{D-4}$$

where XD = the sum of the downscattering cross sections.

It can be shown that

$$\Sigma_{tr} = \Sigma_a + \Sigma_s + XD - \Sigma_{n,2n} \ . \tag{D-5}$$

where $\Sigma_{n,2n}$ = (n,2n) cross section.

Substituting equation (D-5) into equation (D-4) gives the following:

$$XD = XD - \Sigma_{n,2n} \ . \tag{D-6}$$

Since XD cannot equal XD - $\Sigma_{n,2n}$, this implies that 2DB has incorrectly calculated the downscattering component for those problems which have a significant amount of (n,2n) scattering reactions taking place. Therefore, 2DB was changed so that the downscattering term was calculated by summing over all the downscattering cross sections for a specified group of neutrons (see Appendix H, subroutine S860).

APPENDIX E

PLASMA SOURCE

The problem to be solved is the determination of the uniformly distributed source of 14.1 MeV neutrons in the plasma (radius = R_{pl}) impinging upon the blanket's inner wall at R_W. It is assumed that 10 MW/m^2 will be the maximum wall load, thus, the current on the inner wall is given by

$$J = \frac{10\text{MW}}{\text{m}^2}\left(\frac{10^6\text{W}}{\text{MW}}\right)\left(\frac{\text{J/s}}{\text{W}}\right)\left(\frac{\text{m}^2}{10^4\text{cm}^2}\right)\left(\frac{\text{MeV}}{1.6 \times 10^{-13}\text{J}}\right)\left(\frac{1}{14.1\ \text{MeV/n}}\right) \qquad \text{(E-1)}$$

$$= 4.43 \times 10^{14}\frac{\text{n}}{\text{cm}^2\text{- sec}} \ .$$

Equating the number of neutrons emitted in the plasma to the number of neutrons impinging upon the inner wall, then for a unit width in the toroidal direction the balance is given by

$$J\, 2\pi R_W \ = \ \text{FRV}\pi R^2{}_{pl} \qquad \text{(E-2)}$$

where FRV = the unifomly distributed source of 14.1 MeV neutrons in the plasma. Rearranging equation (E-2) to solve for FRV gives the following:

$$\text{FRV} = \frac{J\, 2\pi R_w}{R^2_{pl}} \ . \qquad \text{(E-3)}$$

Substitiuting for J from equation (E-1), while letting R_w = 200cm and R_{pl} = 150cm, the uniformly distributed source of 14.1 MeV neutrons in the plasma is equal to 7.88 x 10^{12} n/cm^2-sec.

APPENDIX F
EQUIVALENT SOURCE TO THE PLASMA SOURCE AT THE INNER EDGE OF THE BLANKET

The problem to be solved is the determination of the equivalent source at the inner edge of the blanket (radius = R_w) to the of the uniformly distributed source of 14.1 MeV neutrons in the plasma (radius = R_{pl}). Equating the number of neutrons emitted from the plasma to the number of neutrons emitted from a thin source (width = Δr) at the inner wall (radius = R_w), then for a unit width in the toroidal direction the balance is given by

$$\frac{FRV\ \pi R_{pl}^2}{2\pi R_{pl}} = \frac{S_e \pi \left((R_w + \Delta r)^2 - R_w^2\right)}{2\pi R_w} \tag{F-1}$$

where FRV = the unifomly distributed source of 14.1 MeV neutrons in the plasma and S_e = equivalent source at the inner wall. Rearranging and`simplifying equation (F-1) to solve for S_e gives the following:

$$S_e = \frac{FRV\ R_{pl} R_w}{\left((R_w + \Delta r)^2 - R_w^2\right)} \quad . \tag{F-2}$$

APPENDIX G

2DBTOR MANUAL

2DBTOR MANUAL

Craig Hrabal

Babatunde Shofolu

Theodore Parish

Texas A&M University

College Station, Texas 77840

Acknowledgements

Much of this manual for 2DBTOR was influenced by both the previous 2DB manual (2DB User's Manual written by W. W. Little, Jr. and R. W. Hardie) and the ANISN/PC manual (ANISN/PC Manual written by D. Kent Parsons). Some of the sections of both manuals were incorporated into the 2DBTOR manual, although the style was based for the most part on the ANISN/PC manual.

Contents

ACKNOWLEDGEMENTS 86

ABSTRACT 89

1. INTRODUCTION 90

2. THEORETICAL FOUNDATIONS 91

2.1 Discretization of the Diffusion Equation 91

2.1.1 Energy Discretization 91

2.1.2 Spatial Discretization -- The Finite Difference Method 94

2.2 Discussion of Boundary Conditions Used in 2DBTOR 97

2.2.1 Zero Flux Gradient Boundary Condition 97

2.2.2 Zero Flux Boundary Condition 98

2.2.3 Periodic Boundary Condition 99

2.3 Discussion of Triangular Mesh Option 100

2.4 Iterative Solution Methods of 2DBTOR 101

2.4.1 Outer Iterations 102

2.4.2 Inner Iterations 103

3. USER'S GUIDE 105

3.1 2DBTOR Input Data Description (Logical Unit 5) 105

3.2 Supplemental Input Information for 2DBTOR 121

3.2.1 Cross Section Considerations 121

3.2.2 Search Considerations 123

3.2.2.1 Time Absorption (α Calculation) 123

3.2.2.2 Material Concentration (C Calculation) 124

3.2.2.3 Zone Dimensions (δ Calculation).................... 126

3.2.2.4 Buckling (B^2 Calculation).............................. 126

3.2.3 Burnup Model.. 126

3.2.3.1 Remarks on Burnup Equations........................ 128

3.2.4 Source Problems... 128

3.2.5 Remarks on Code Operation.................................... 129

3.3 2DBTOR Format Description... 130

3.3.1 Operators.. 130

3.4 Printed Output Description.. 131

4. PROGRAMMER'S GUIDE TO 2DBTOR.................................. 133

4.1 2DBTOR Files Description.. 133

4.2 2DBTOR Subroutines... 134

4.2.1 Subroutine Calling Sequences.................................. 136

4.3 Selected Definitions of Variables and Arrays Used.................. 138

5. REFERENCES.. 146

2DBTOR MANUAL -- REVISION OF 2DB MANUAL (REVISION 1)

ABSTRACT

1. Program Identification: 2DBTOR is a revised version of 2DB[1].

2. Description of Problem: 2DBTOR is a two-dimensional (X-Y, R-Z, R-θ, triangular, toroidal), multi-group neutron diffusion code for use in fast reactor criticality and burnup analysis. In addition, 2DBTOR may also be used to study fusion blanket problems in toroidal geometry. 2DBTOR solves the multi-group diffusion theory eigenvalue, adjoint, time absorption, fixed source and criticality search (concentration, zone thickness, and buckling) problems.

3. Method of Solution: Multi-group finite difference neutron diffusion equations are solved iteratively in 2DBTOR. The power method, accelerated by a fission source over-relaxation factor calculated in the code, is used for the outer iterations. Inner iterations are accelerated by use of an over-relaxation factor input by the user.

1. INTRODUCTION

A number of significant additions and alterations (e.g., a toroidal geometry option and an activity cross section option) have been made to the 2DB[1] diffusion-burnup code. In addition, some bugs were discovered in the old 2DB version which have been corrected in the current code, 2DBTOR. This manual gives a complete description of the code including all modifications. A description of both the mathematical model and user instructions are given in the body of the report; a sample problem is included in the appendix.

2DBTOR is designed for use in both fast reactor and fusion analysis. Eigenvalues are computed by standard source-iteration techniques. Group rebalancing and successive over-relaxation with line inversion are used to accelerate convergence. Adjoint solutions are obtained by inverting the input data and redefining the source terms.

Variable dimensioning is used to make maximum use of the available fast memory. Since only one energy group is in the fast memory at any given time, the storage requirements are insensitive to the number of energy groups.

Criticality searches can be performed on buckling, time absorption, material concentrations, and region dimensions. Alpha and k_{eff} can be used as parametric eigenvalues. Criticality searches can be performed during burnup to compensate for fuel depletion.

2. THEORETICAL FOUNDATIONS

2.1. Discretization of the Diffusion Equation

2.1.1. Energy Discretization. The diffusion equation can be developed from the equation for the time rate of change of the number of neutrons within dE of energy, E, in an arbitrary differential volume, dV. Neutrons of energies between E and E+dE, within dV, can be lost or gained by a variety of processes including: (1) production directly from a source, (2) absorption, (3) leakage and (4) scattering. The time rate of change of the number of neutrons in dV and between E and E+dE can be obtained by integrating the neutron density (n(r,E,t)) over dV, and balancing this with the gains and losses as follows:

$$\frac{\partial}{\partial t}\int_{V}\frac{\Phi(\vec{r},E,t)}{\nu}\,dV = \begin{bmatrix}\text{source neutron}\\ \text{production rate}\\ \text{in V at E}\end{bmatrix} - \begin{bmatrix}\text{absorption}\\ \text{rate in V}\\ \text{at E}\end{bmatrix} - \begin{bmatrix}\text{change due}\\ \text{to leakage}\\ \text{from V at E}\end{bmatrix} - \begin{bmatrix}\text{neutron scattering}\\ \text{rate out of}\\ \text{E in V}\end{bmatrix} + \begin{bmatrix}\text{neutron scattering}\\ \text{rate into}\\ \text{E in V}\end{bmatrix} \tag{1}$$

where $\Phi(r,E,t)$ = flux of neutrons at r, E and t $= n(r,E,t)\nu$

and ν = the speed of the neutrons at E.

Equation (1) is known as the neutron continuity equation. Since the energy dependence of the neutron cross sections vary, equation (1) is usually solved for discrete energy groups(groups denoted by g in this case); thus equation (1) can be written as

$$\frac{\partial}{\partial t}\int_V \frac{\Phi_g(\vec{r},t)}{\nu_g}\, dV = \begin{bmatrix}\text{source neutron}\\ \text{production rate}\\ \text{in V for}\\ \text{group, g}\end{bmatrix} - \begin{bmatrix}\text{absorption}\\ \text{rate in V}\\ \text{for group, g}\end{bmatrix} - \begin{bmatrix}\text{change due}\\ \text{to leakage}\\ \text{from V}\\ \text{for group, g}\end{bmatrix} - \begin{bmatrix}\text{neutron scattering}\\ \text{rate out of}\\ \text{group, g in V}\end{bmatrix} + \begin{bmatrix}\text{neutron scattering}\\ \text{rate into}\\ \text{group, g in V}\end{bmatrix} \tag{2}$$

Assuming steady state ($\partial/\partial t$ term = 0) and no upscattering, equation (2) becomes upon substituting the corresponding mathematical expressions for the RHS terms,

$$-\vec{\nabla}\cdot(D_g\vec{\nabla}\Phi_g) + \Sigma_{a_g}\Phi_g + \sum_{g'=g+1}^{G}(\Sigma_s(g\rightarrow g')\,\Phi_g) = S_g^{ext} + \chi_g\sum_{g'=1}^{G}(\nu_{g'}\Sigma_{f_{g'}}\Phi_{g'}) + \sum_{g'=1}^{g-1}(\Sigma_s(g\rightarrow g')\,\Phi_{g'}) \tag{3}$$

where the (r,t) arguments have been dropped for clarity. This is the multi-group neutron diffusion equation.

Equation (3) can be further simplified by noting that the removal of neutrons from group g is caused by both downscattering and absorption. The removal cross section for group g is defined as shown below:

$$\begin{aligned}\Sigma_g^r &\equiv \Sigma_{a_g} + \sum_{g'=g+1}^{G} (\Sigma_s(g\to g')) \\ &= \Sigma_{a_g} + \Sigma_s(g\to g) + \left[\sum_{g'=g+1}^{G} \Sigma_s(g\to g')\right] - \Sigma_s(g\to g) \\ &= \Sigma_{tr_g} - \Sigma_s(g\to g')\end{aligned}$$

where Σ_{trg} = the macroscopic transport cross section $= 1/(3D_g)$.[14]

Thus the removal rate/cm^3 is

$$\Sigma_g^r\Phi_g = \Sigma_{a_g}\Phi_g + \sum_{g'=g+1}^{G} (\Sigma_s(g\to g')\,\Phi_g). \tag{4}$$

Substituting equation (4) into equation (3) gives

$$-\vec{\nabla}\cdot(D_g\vec{\nabla}\Phi_g)+\Sigma_g^r\Phi_g = S_g^{ext} + \chi_g \sum_{g'=1}^{G} (\nu_{g'}\Sigma_{f_{g'}}\Phi_{g'}) + \sum_{g'=1}^{g-1}(\Sigma_s(g\to g')\,\Phi_{g'}) \tag{5}$$

the form of the multi-group neutron diffusion equation used in 2DBTOR.

2.1.2. Spatial Discretization - The Finite Difference Method. To develop a finite difference approximation for the diffusion equation(with axial symmetry), it is first necessary to integrate equation (5) over a small, arbitrary volume ΔV (see Figure 2-1) where the mesh points are considered to be in the center of the homogeneous mesh interval. Thus,

$$-\int_{\Delta V} \vec{\nabla} \cdot D_g \vec{\nabla} \Phi_g dV + \int_{\Delta V} \left[\Sigma_g^r \Phi_g \right] dV \tag{6}$$

$$= \int_{\Delta V} \left[S_g^{ext} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_{g'}} \Phi_{g'}) + \sum_{g'=1}^{g-1} (\Sigma_s(g \rightarrow g') \Phi_{g'}) \right] dV$$

where the first term on the LHS of the equation is the leakage term, the second term on the LHS of the equation is the removal term, and the RHS represents the source terms including fission and scatter. Thus for the removal term,

$$\int_{\Delta V} \Sigma_g^r \Phi_g \, dV = \Sigma_{g_0}^r \Phi_{g0} V_o \tag{7}$$

where Φ_{g0} = flux associated with meshpoint o

and $\Sigma_{g_0}^r$ = removal cross section associated with meshpoint o.

The source term on the RHS is done similarly to the above. The leakage term is changed to an integral over the surface area of the volume element, thus from the Divergence Theorem

$$-\int_{\Delta V} \vec{\nabla} \cdot D_g \vec{\nabla} \Phi_g \, dV = -\int_A D_g \vec{\nabla} \Phi_g \cdot \hat{n} \, dA \quad .$$

The flux partial derivatives will be obtained by differencing the two neighboring flux values. Thus, volume integration of equation (5) for mesh point 0 (see Figure 2-1; where r stands for x or r, and θ stands for y, z, or θ, depending on the geometry) leads to the expression

$$-\sum_{k=1}^{4} \left[\overline{D}_{g_k} \left(\frac{\phi_{g_k} - \phi_{g_0}}{L_k} \right) A_k \right] + \Sigma^{r}_{g_0} \Phi_{g_0} V_0 \tag{8}$$

$$= \left[S^{ext}_{g_0} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_{g'_0}} \Phi_{g'_0}) + \sum_{g'=1}^{g-1} (\Sigma_{s_0}(g \rightarrow g') \, \Phi_{g'_0}) \right] V_0 \ .$$

where $\Sigma^{r}_{g_0}$ = removal cross section associated with mesh point 0,

$S^{ext}_{g_0}$ = extraneous source rate associated with mesh point 0,

V_0 = volume associated with mesh point 0.

Φ_{g_k} = flux associated with mesh point 0,

L_k = distance between mesh point k and mesh point 0,

A_k = area of boundary between mesh point k and mesh point 0,

and $\bar{D}_{g_k}$ = effective diffusion coefficient mesh point k and mesh point 0

$$\left(= \frac{D_{g_0} D_{g_k}(\Delta r_o + \Delta r_k)}{(D_{g_0}\Delta r_k + D_{g_k}\Delta r_o)} \quad \text{or} \quad \frac{D_{g_0} D_{g_k}(\Delta\theta_o + \Delta\theta_k)}{(D_{g_0}\Delta\theta_k + D_{g_k}\Delta\theta_o)} \right.$$

$$\left. \text{where} \quad \Delta r_k = \Delta r \text{ for volume element k,} \quad \Delta\theta_k = \Delta\theta \text{ for volume element k} \right)$$

Equation (8) can be recast into a form more convenient for performing flux iterations by rearranging equation (8) to that given below:

$$\phi_0 = \frac{\left[S^{ext}_{g_0} + \chi_g \sum_{g'=1}^{G} (\nu_{g'} \Sigma_{f_{g'_0}} \Phi_{g'_0}) + \sum_{g'=1}^{g-1} (\Sigma_{s_0}(g \rightarrow g') \Phi_{g'_0}) \right] V_0 + \sum_{k=1}^{4} C_k \phi_k}{C_5} \quad , \tag{9}$$

where
$$C_k = \frac{\bar{D}_{g_k} A_k}{L_k} \quad k=1, \dots 4 \tag{10}$$

and
$$C_5 = \Sigma^{r}_{g_0} V_0 + \sum_{k=1}^{4} C_k \quad . \tag{11}$$

2.2. Discussion of Boundary Conditions Used in 2DBTOR

Three boundary conditions are available in 2DBTOR: reflective, extrapolated vacuum, and periodic. The reflective boundary conditions is used on boundaries where $\nabla\Phi = 0$; the extrapolated vacuum boundary condition is used on boundaries where the flux is assumed to be zero at the extrapolated boundary; and the periodic boundary condition is used on boundaries where material conditions are repeating. The above mentioned boundary conditions are described in more detail below.

2.2.1. Zero Flux Gradient Boundary Condition. Consider the left-hand boundary of the one-dimensional reactor shown in Figure 2-2. Imagine that a

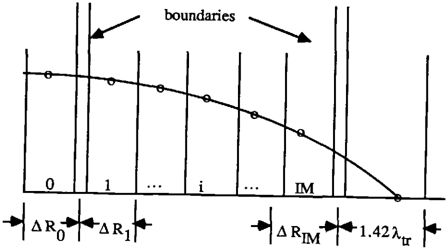


Figure 2-2. Schematic Diagram of 1-D Reactor

pseudo-mesh interval, interval 0, has been added on the left-hand side of the boundary with the same composition and thickness of interval 1. Clearly, then, if $\nabla\Phi = 0$ at the boundary, $\phi_0 = \phi_1$. Therefore, since $(\phi_0-\phi_1)=0$, the coefficient of $(\phi_0-\phi_1)$, C_1(see equations (8) and (10)), is immaterial -- hence C_1 can be set equal to zero. The calculation is therefore performed assuminf that ϕ_0 does not exist and $C_1 = 0$.

2.2.2. Zero Flux Boundary Condition. Again, imagine that a pseudo-mesh interval with the same composition as interval IM has been added to the right hand side of the boundary in Figure 2-2. Now, since $\phi_{IM} \neq 0$ and $\phi_{IM+1} = 0$, the coefficient of $(\phi_{IM}-\phi_{IM+1})$ in equation (8) cannot be disregarded. In fact, from equation (10), it is clear that

$$C_k = \frac{D_k A_k}{.5\,\Delta R_{IM} + .71\lambda_{tr}}$$

where λ_{tr} is assumed to equal $1/\Sigma_{tr}$.

Note , as in the reflective boundary condition case, that there is no contribution of the pseudo-flux in equation (9). For a zero flux gradient, $C_k = 0$; whereas for a zero flux, $\phi_k = 0$.

2.2.3. Periodic Boundary Condition. Periodic boundary conditions are only available for the top and bottom boundaries(e.g., boundaries in the y, z, or θ direction). In this option(see Figure 2-3),

$$\phi_{IM} = \phi_0$$

$$\phi_{IM+1} = \phi_1$$

and

$$C_k(1 \rightarrow IM) = \frac{\overline{D}_k A_k}{.5(\Delta R_1 + \Delta R_{IM})} .$$

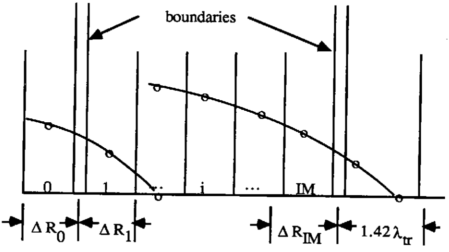


Figure 2-3. Schematic Diagram of 1-D Reactor

It should be stressed that the pseudo-mesh intervals discussed above are not in any way a part of the code. They are mentioned here only for heuristic purposes.

2.3. Discussion of Triangular Mesh Option

Since most fast reactors are composed of hexagonal assemblies, 2DBTOR includes a triangular mesh option. Hexagons are formed by appropriate grouping of six triangular mesh intervals.

In the triangular mesh option, the (i,j) coordinate grid is composed of a rectangular array of triangles. As in the other geometry options, the mesh points are placed in the center of each interval, or triangle. See Figure 2-4 for a simple 3 x 4 example. In contrast to the other geometry options, however, the mesh boundaries must be equally spaced. In fact, the radial (RB_i) and axial (θB_j) mesh boundaries must be computed by the expressions

$$RB_i = (i-1)\frac{FTF}{2\sqrt{3}}, \qquad i=1, \ldots, IM+1 \tag{12}$$

$$\theta B_j = (j-1)\frac{FTF}{2}, \qquad j=1, \ldots, JM+1 \tag{13}$$

where FTF is the flat-to-flat hexagon width.

Only vacuum and reflective boundary conditions are available with the triangular mesh option. The user is cautioned against using reflective left and right boundaries since this implies no surface leakage from each mesh interval on the left and right border.

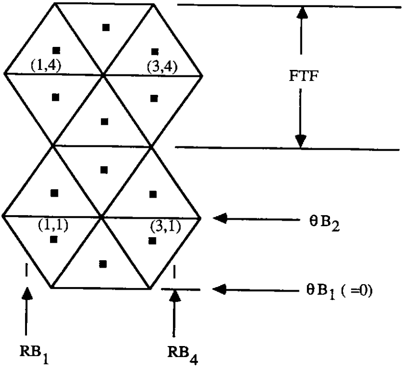


Figure 2-4. Triangular Mesh Example (3 x 4)

2.4. Iterative Solution Methods of 2DBTOR

Within the 2DBTOR code, two distinct levels of iteration may be found for general problems. The top level of iteration (i.e., outer iterations) is for the spatially and energy-group summed fission sources. The second level of iteration (i.e., inner iterations) is for the individual group fluxes that result from a given source.

The following sections describe both the inner and outer iteration procedures used in 2DBTOR and discuss the methods used to accelerate those procedures.

2.4.1. Outer Iterations. Outer iterations in 2DBTOR are based on the power iteration method. That is, at each outer iteration, a total fission source is calculated. Upon convergence, the ratio of the latest fission source to the previous fission source is the eigenvalue. Thus, the eigenvalue is used to renormalize the fission sources between outer iterations, and the ratio between normalized fission source iterates approaches 1.0.

For search problems, the eigenvalue is defined to be the value of the search quantity (e.g., time absorption, zone thickness,etc.) that produces criticality. In these problems, the eigenvalue is used to change the problem at each search step so that the fisdsion source ratio still approaches 1.0 at convergence.

For each outer iteration, the inner iteration procedure starts with group 1 and sweeps through the groups in order of decreasing energy. The downscattering component of the source for the current group is calculated from the latest values of the higher energy fluxes.

Fission source over-relaxation is employed in 2DBTOR to accelerate convergence. The procedure is as follows: After the new fission source rate profile, $F^{v+1}{}_1$, is calculated, a second "new" value, $F^{v+1}{}_2$, is computed by magnifying the difference between the new fission source rate and the old fission source rate. Thus,

$$F_2^{v+1} = F^v + \beta'(F_1^{v+1} - F^v),$$

where β' is the fission source over-relaxation factor. $F^{v+1}{}_2$ is than normalized to give the same total source as $F^{v+1}{}_1$.

2.4.2. Inner Iterations. Inner iterations are computed using successive line over-relaxation (SLOR). That is, the fluxes on each vertical (or horizontal) line are simultaneously computed (by the familiar Crout reduction technique) and then over-relaxed using

$$\phi^{v+1} = \phi^{v} - \beta(\phi^{v+1} - \phi^{v}) ,$$

where β is the over-relaxation factor. In r-q problems or problems involving periodic boundary conditions, direct inversion is performed on vertical lines beginning at the left boundary and proceeding by column to the right boundary. In triangular problems, direct inversion is performed along horizontal lines beginning at the bottom boundary and proceeding by row to the top boundary. In all other situations, direct inversion is performed along the dimension with the most mesh points. One mesh sweep is defined as one inner iteration.

The flux over-relaxation factor, β, is an input parameter. The fission source over-relaxation factor, β', is computed internally from the expression

$$\beta' = 1.0 + 0.6(\beta - 1) .$$

As in the original version of 2DB, the flux in each group is normalized (by balancing the total source rate and loss rate) immediately before each group-flux calculation. Thus, a one-region problem with zero-gradient boundary conditions would be solved in exactly one outer iteration.

It should be mentioned that an alternating direction SLOR scheme (using line inversion for rows and then colums in alternation) is included as an option to enhance convergence for problems involving tight mesh spacing in both directions.

3. USER'S GUIDE

3.1. 2DBTOR Input Data Description (Logical Unit 5)

This section describes the input data for the 2DBTOR code. The input has the following general structure:

Section	Description
A	Title Card
B	Single Integer and Real Numbers
C	Cross Section Data
D	Fixed Source Data (if needed)
E	Miscellaneous Data
F	Burnup Data

Each input section is begun on a new line and ended on a later line by a terminate marker. A "T" anywhere on a nontitle card or on a line by itself constitutes a terminate marker.

Each of the input sections will be described below. Locations of the terminate markers will also be given. The length of each data section is denoted by the number or variable in slashes "//" by the input description. The expression in braces "{}" by the input description is the condition that requires the input to be present. If no condition is given, the input is always needed.

The data format conventions used by 2DBTOR are described fully in Section 3.3. Succinctly descibed, however, 2DBTOR uses free format augmented by an operator notation, which conserves space in the input.

Further details about the various input options for 2DBTOR may be found in Section 3.2, which immediately follows the input data description.

A. Title Card - (4A20 Format)

B. Single Integer and Real Numbers

B.1 Integer control parameters /13/

	Variable Name	Description
1.	A02	0 - regular calculation
		1 - adjoint calculation
2.	I04	Eigenvalue type
		0 - distributed source (D.)
		1 - eigenvalue calculation
		2 - time absorption (α) search calculation (E.7.)
		3 - concentration search (C) calculation

Variable Name		Description
		4 - zone thickness search (δ) calculation (E.11.-14.)
		5 - buckling (B^2) calculation (E.5.)
3.	S02	Parametric eigenvalue type 0 - no effect 1 - eigenvalue (k_{eff}) 2 - α (E.7.)
4.	IGM	Number of energy groups
5.	IHT	Position of σ_{tr}
6.	NXCM	Number of downscattering terms
7.	MCR	Number of cross section sets to be read from cards/tape (+N/-N) (+N go to C.)
8.	G07	Maximum number of inner iterations for each group per outer iteration (suggested value $\geq$ 20)

	Variable Name	Description
9.	D05	Maximum number of outer iterations(suggested value ≥ 50 for general usage but D05 = 1 fixed source calculations without fission)
10.	MAXT	Maximum run time (minutes)
11.	NPRT	Print option 0 - mini 1 - midi 2 - cross sections 3 - fluxes
12.	M07	Flux guess 0 - no effect 1 - flux guess from file FOR14.DAT
13.	NPUN	Flux dump 0 - no effect 1 - flux dump to file FOR16.DAT

Terminate Marker

B.2 Integer control parameters /14/

	Variable Name	Description
1.	IGE	Geometry parameter 0 - X-Y 1 - R-Z 2 - R-θ or toroidal (B.2.2.) 3 - triangular 4 - toroidal
2.	ITOR	Toroidal specifier (0/1 = R-θ/ toroidal)
3.	NACT	Number of activities (E.15.) 0 - no effect >0 - read table positions for N activities
4.	IM	Number of radial fine mesh intervals
5.	JM	Number of axial fine mesh intervals
6.	IZM	Number of zones (or regions)
7.	MT	Total number of material (|MCR| + mixtures formed in

	Variable Name	Description
		mixing table)
8.	M01	Mixing table length
9.	B01	Left boundary condition
		0 - vacuum
		1 - reflection
10.	B02	Right boundary condition
		0 - vacuum
		1 - reflection
11.	B03	Top boundary condition
		0 - vacuum
		1 - reflection
		2 - periodic
12.	B04	Bottom boundary condition
		0 - vacuum
		1 - reflection
		2 - periodic
13.	IZ	Number of radial zones
		0 - no effect
		>0 - only if I04 = 4
14.	JZ	Number of axial zones
		0 - no effect
		>0 - only if I04 = 4

Terminate Marker

B.3 Floating point control parameters /6/

	Variable Name	Description
1.	EV	First eigenvalue guess
2.	EVM	Initial eigenvalue modifier for search problems, zero otherwise
3.	S03	parametric eigenvalue
4.	BUCK	Buckling (cm^{-2}) (E.5)
5.	LAL	Lower limit for λ in search calculations, zero otherwise
6.	LAH	Upper limit for λ in search calculations, zero otherwise

Terminate Marker

B.4 Floating point control parameters /6/

	Variable Name	Description
1.	EPS	Eigenvalue relative convergence criterion (suggested value, EPS = 0.0001)
2.	EPSA	Parametric convergence criterion
3.	G06	Pointwise flux relative convergence criterion, zero otherwise (suggested value, EPSA = 2.0*EPS)
4.	POD	Paramater oscillation damper (suggested value, POD = 1.0)
5.	ORF	Over relaxation factor (suggested value, $1.0 \leq ORF \leq 2.0$)
6.	S01	S01<0, power (MWT) for R-Z geometry, (power/height (MWT/cm) for all but R-Z geometry) S01>0, neutron source rate (n/cm^3) (0.0 for source without fission)

Terminate Marker

B.4 Floating point control parameters /1/ {ITOR = 1}

	Variable Name	Description
1.	BIGR	Major radius (cm)

Terminate Marker

C. Cross Section Data {MCR>0}

	Variable Name	Description
1.	HOLN(MCR)	Name of isotope
2.	ATW(MCR)	Atomic weight of isotope

Terminate Marker

ITL = NXCM + IHT + 1 = Cross section table length

3.	C(ITL,IGM,MT)	Read cross sections for first group /ITL/.

4. *Terminate Marker*

5. Repeat C.3 and C.4 for all groups /IGM-1/

6. Repeat C.1 - C.5 for all materials /MT-1/

D. Fixed Source Data {I04 = 0}

	Variable Name	Description
1.	S2(IM*JM)	Source in first group

2. *Terminate Marker*

3. Repeat D.1 - D.2 for all groups /IGM-1/

E. Miscellaneous Data

	Description
1.	Radial mesh line coodinates defining the IM fine mesh intervals /IM+1/ (should be strictly ascending in order)

Terminate Marker

Description

2. Axial mesh line coordinates defining the JM fine mesh intervals /JM+1/ (should be strictly ascending in order)

Terminate Marker

3. Zone numbers by fine mesh interval /IM*JM/

Terminate Marker

4. Material numbers by zone /IZM/

Terminate Marker

5. Buckling coefficients by zone /IZM/ {I04=5 or BUCK>0}

Terminate Marker

6. Fission spectrum data /IGM/ (the sum of the entries should equal 1.0 for eigenvalue calculations (I04=1))

Terminate Marker

Description
7. Neutron velocities by group /IGM/ {I04=2 or S02=2}

Terminate Marker

8. Mixture material numbers in mixing table /M01/ {M01>0}

Terminate Marker

9. Component material numbers of mixtures in mixing table /M01/ {M01>0}

Terminate Marker

10. Atom densities of component materials in mixing table /M01/ {M01>0}

Terminate Marker

11. Delta option radial zone numbers by fine mesh interval /IM/ {I04=4}

Terminate Marker

12. Delta option radial zone modifiers /IZ/ {I04=4}

Terminate Marker

13. Delta option axial zone numbers by fine mesh interval /JM/ {I04=4}

Terminate Marker

14. Delta option axial zone modifiers /JZ/ {I04=4}

Terminate Marker

15. Cross section table position for activities /NACT/ {NACT>0}

Terminate Marker

16. End of problem identifier (NCON)

    0 - End of problem (only if no burnup)

    >0 - Take time step of DELT and read burnup data for N isotopes (F.)

    <0 - Take time step of DELT (F.)

    (DELT is in Burnup Data)

Terminate Marker

F. Burnup Data {NCON≠0}

F.1 Integer control parameters /1/

	Variable Name	Description
1.	DELT	Time step (days)

Terminate Marker

F.2 Integer control parameters /12/ {NCON>0}

1.	MATN(NCON)	Material sequence number of burnable isotope
2.	NBR(NCON)	0 - No effect 1 - Fertile isotope 2 - Fissile istope
3.	LD(NCON)	0 - No decay source N - Decay source from burnable isotope N

	Variable Name	Description
4.	LCN(NCON,1)	0 - No capture source N - Capture source from burnable isotope N
5.	LCN(NCON,2)	0 - No capture source N - Capture source from burnable isotope N
6.	LFN(NCON,1)	0 - No fission source N - Fission source from burnable isotope N
7.	LFN(NCON,2)	0 - No fission source N - Fission source from burnable isotope N
8.	LFN(NCON,3)	0 - No fission source N - Fission source from burnable isotope N
9.	LFN(NCON,4)	0 - No fission source N - Fission source from burnable isotope N
10.	LFN(NCON,5)	0 - No fission source N - Fission source from burnable isotope N

	Variable Name	Description
11.	LFN(NCON,6)	0 - No fission source N - Fission source from burnable isotope N
12.	LFN(NCON,7)	0 - No fission source N - Fission source from burnable isotope N

Terminate Marker

F.3 Integer control parameters /1/ {NCON>0}

1.	ALAM(ITEMP)	Decay constant for decay of burnable isotope N (days-1) (0.0 for no decay)

Terminate Marker

3.2. Supplemental Input Information for 2DBTOR

3.2.1. Cross Section Considerations. Cross sections input into 2DBTOR are ordered for each group as shown in Table 2.

Table 2. Order of cross sections in 2DBTOR

Cross Section	Group	Position Description
σ activity 1 (optional)	g	
.	.	
.	.	
.	.	
σ activity N (optional)	g	(N = IHT - 4)
σ fission	g	
σ absorption	g	
$\nu\sigma$ fission	g	
σ transport	g	IHT
σ selfscatter	$g \rightarrow g$	IHS
σ downscatter	$g - 1 \rightarrow g$	
.	g	
.	g	
.	g	
σ downscatter	g - NXCM$\rightarrow g$	ITL

If activity cross sections are *not* present, then IHT = 4.

The absorption cross section is used only for editing purposes. If a removal cross section is to be calculated, then 2DBTOR computes this by subtracting the self scatter cross section from the transport cross section.

Material numbers in 2DBTOR start at 1 and go through MT (the user specified total number of materials). Materials entered by cards or tape start at 1 and run through MCR (the user-specified number of materials from cards or tape). Materials formed in the mixing table are numbered from MCR+1 through MT.

The cross section mixing table is controlled by three input arrays: I0, I1, and I2. The length of the cross section mixing table (M01) is specified by the user. It is important to remember to initialize each array to zero before performing a mix in the mixing table since 2DBTOT does not do this internally in the program.

For each row of entries in the mixing table, three operations are possible. First, all of the cross sections in a given material number may be multiplied by a constant. This option is useful in number density variations. Second, a set of cross sections may be multiplied by a constant and added into another material. This option is useful in mixing cross sections. Finally, all of the cross sections of a material may be multiplied by the eigenvalue. This option is useful in concentration searches (I04 = 3).

Table 3 illustrates the three options available from the mixing table:

Table 3. 2DBTOR mixing table options

Options	Material Number (I0)	Component Number (I1)	Concentration or Numeric Constant (I2)
1	M	0	X
1	M	0	X
1	M	0	X

3.2.2. Search Considerations. The 2DBTOR code computes implicit eigenvalue searches on time absorption, material composition, zone thickness, and material buckling. In contrast to a k_{eff} calculation, the fission spectrum is not multiplied by $1/\lambda$ after each outer iteration. Instead, after a converged λ has been obtained ($|\lambda^{\nu+1} - \lambda^{\nu}| < \varepsilon'$) by a sequence of outer iterations, the desired parameter is perturbed to make λ approach unity. That is, first a converged λ is calculated for the initial system. The system is then altered by the amount specified in the input (the eigenvalue modifier) and a second converged λ is calculated. Subsequent parameter changes are determined using either linear or parabolic interpolation procedures. The iteration is continued until $|1- \lambda| < \varepsilon$.

3.2.2.1. Time Absorption (α calculation). For simplicity, consider the one group, time dependent neutron diffusion equation

$$\frac{1}{\nu}\frac{\partial\phi(\vec{r},t)}{\partial t} = D\nabla^2\phi(\vec{r},t) - \Sigma_a\phi(\vec{r},t) + \nu\Sigma_f\phi(\vec{r},t) \ . \tag{14}$$

If one now assumes that

$$\phi(\vec{r},t) = \phi(\vec{r})^{\alpha t} \quad . \tag{15}$$

then equation (14) can be rewritten in the form

$$D\nabla^2\phi(\vec{r}) - (\Sigma_a + \frac{\alpha}{\nu}\phi(\vec{r}) + \nu\Sigma_f\phi(\vec{r}) = 0 \ . \tag{16}$$

In a time absorption calculation, the parameter α, as defined and used in equations (15) and (16), is computed as the eigenvalue. Note that α/ν is effectively an absorption cross section -- hance the name "time absorption."

3.2.2.2 Material Concentration (C calculation). 2DBTOR can perform an extremely flexible and comprehensive criticality search on material composition. Any number of materials can simultaneously be added, depleted, or interchanged in any number of zones.

The format for specifying concentration searches can best be described by a simple example. Suppose that a zone mixture, say Mix 10, is to be composed of Materials 2 and 4. Further, assume that Material 2, with an initial density of 0.02 (atoms/barn-cm), shall be varied to obtain criticality, and Material 4 shall have a fixed density of 0.04 (atoms/barn-cm).

The I0, I1, and I2 arrays would then be set up as shown in the following tabulation.

Mix Number (I0)	Material Number(I1)	Density(I2)
10	0	0
10	2	0.02
10	10	0
10	4	0.04

The first row (10,0,0) instructs the code to clear a storage area for Mix 10. The second row (10,2,0.02) causes Material 2 to be added to Mix 10 with a density of 0.02. The third row (10,10,10) cuses the current contents of Mix 10 to be multiplied by the eigenvalue. Finally, the last row (10,4,0.04) instructs the code to add Material 4 to Mix 10 with a density of 0.04.

All of the foregoing can be summarized by the expression

$$\Sigma_{10} = 0.02\,\sigma_2\,EV + 0.04\sigma_4 \tag{17}$$

where:

Σ_{10} = macroscopic cross section for Mix 10,

σ_2= microscopic cross section for Material 2,

σ_4 = microscopic cross section for Material 4,

EV = the eigenvalue.

3.2.2.3 Zone Dimensions (δ calculation). 2DBTOR searches on reactor dimensions by varying the dimensions of each axial an radial mesh interval. Each mesh width, δr^i, is computed from the expression

$$\delta r^i = \delta r_0^i [1 + (\text{mesh modifier})^i \, \text{EV}] \ , \qquad (18)$$

where δr^i is the initial mesh spacing and EV is the eigenvalue. Different mesh modifiers can be specified for each axial and radial mesh interval.

3.2.2.4 Buckling (B^2 calculation). In a buckling search, the quantity $D_i B^2$, where D_i is the zone dependent diffusion constant for group i, is added to the ith group absorption cross section. The in-group scattering cross section, σ^i_{gg}, is reduced by the same amount so that the calculated total cross section remains equal to the input total cross section. The buckling is then computed as the eigenvalue.

3.2.3. Burnup Model. The basic burnup equation for each zone has the form

$$\frac{dN^i}{dt} = -\lambda^i N^i - \bar{\sigma}_a^i \bar{\phi} N^i + \lambda^k N^k + \sum_{j=1}^{2} \bar{\sigma}_c^j \bar{\phi} N^j + \sum_{m=1}^{7} \bar{\sigma}_f^m \bar{\phi} N^m \qquad (19)$$

where:

N^i = density of nuclide i,

N^i = decay constant for nuclide i,

$\bar{\sigma}_a^i$ = spectrum averaged absorption cross section for nuclide i,

$\bar{\sigma}_f^i$ = spectrum averaged fission cross section for nuclide i,

$\bar{\sigma}_c^i$ = spectrum averaged capture cross section for nuclide i,

$\bar{\phi}$ = total average flux.

The last two terms in equation (19) allow provision for two capture and seven fission sources. The latter option, for example, could be used to compute the fission product buildup.

Each input time step is arbitrarily subdivided into 10 smaller times steps. Equation (19) is then solved as a march-out problem using the subdivided time intervals. If one rewrites equation (19) in the form

$$\frac{d\vec{N}}{dt} = \vec{f}(\vec{N},t) \quad , \tag{20}$$

then the particular march-out algorithm used can be written as

$$\vec{N}_{j+1} = \vec{N}_j + \frac{\delta t}{2}(\vec{f}_j + \vec{f}_{j+1}) \quad , \tag{21}$$

where j is the index on time and δt is the fine-step time interval.

Observe that equation (21) is implicit in the sense that Nj+1 must be known in order to compute fj+1. One must therefore iterate on N at each time point. This procedure leads to the algorithm

$$\vec{N}_{j+1}^{v+1} = \vec{N}_j + \frac{\delta t}{2}(\vec{f}_j + \vec{f}_{j+1}) \ , \tag{22}$$

where v is the iteration index.

3.2.3.1. Remarks on Burnup Equations. The zone averaged flux and cross sections appearing in equation (19) are computed before each time step. The total reactor power (from the burnable isotopes) and flux profile (relative zone fluxes) are held constant during the fine-step march-out described by equation (22).

It should be clear from the mathematical model presented that relatively short time steps should be employed if rapid variations in isotopic concentrations or flux profiles are anticipated.

3.2.4. Source Problems. 2DBTOR will compute the flux profiles resulting from an extraneous (in space and energy) source distribution. The following suggestions will assist the user in running source problems:

1. A source problem is meaningless (and will not converge) if k>1.0.
2. Convergence can be accelerated by giving the code an estimate of k (Card 5, Word 1).
3. A good estimate of the initial total neutron production rate (Card 6, Word 6)

will enhance convergence. This value can be estimated using the simple expression

$$N = \frac{kS}{1 - k} \ , \tag{23}$$

where:

N = total neutron production rate from fission,

S = total neutron source rate from extraneous source,

k = multiplication constant.

3.2.5. Remarks on Code Operation.

1. Since the input data is inverted for transposed calculations, all group indicies in the output of adjoint cases are transposed. Furthermore, the balance tables in adjoint calculations do not have a direct physical interpretation.

2. The material inventory tables are inapplicable for a mixture specification more complex than a mix in a mix (e.g., a mix in a mix in a mix).

3. An isotope cannot be mentioned more than once in the same mix in burnup calculations. If mentioned more than once in other calculations, the printed inventory will be incorrect.

4. Although the new eigenvalue and material densities are computed and printed after the last time step, the zone averaged cross sections and reaction rates are not. These can easily be obtained, however, by simply taking 1 extra burnup step of

zero length. Similarly, the zone averaged cross sections and reaction rates can be obtained in non-burnup runs by simply calling for 1 (dummy) burnup step of zero length.

5. A flux dump is given only when:
    1) A dump is called for, and
    2) The burnup time is zero.

Thus, if a dump is called for in a burnup calculation, only one dump (the initial flux) is given.

6. Tight mesh spacing in the dimension perpendicular to line inversion can cause excessive running time. Thus, if tight mesh spacing is used, it should be along the dimension containing the most mesh intervals. For the same reason, the "dummy" dimension in one-dimensional problems should contain large mesh intervals.

3.3 2DBTOR Format Description

3.3.1 Operators. Generalized Format = OPnn xx

1. nn = first subfield with integer entry
2. OP = operator = second subfield with character entry
3. xx = third subfield with real or integer entry
4. There must not be a delimiter (blank or end of card) between the first and second subfields.

5. There must be a delimiter between the second and third subfields.
6. There must be a delimiter between the third subfield of one operator and and the first subfield of the following operator.

	Operator	Description
1.	C	Continue the current array nn times with the previous xx entries
2.	F	Fill the remainder of the current array with xx
3.	I	Linear interpolation; generate nn entries between xx and the previous number
4.	R	The value xx is generated nn times
5.	T	Terminate the current array

3.4. Printed Output Description

The first output section of 2DBTOR is a brief edit of the first 37 input data preceded by the title card. The size of the array required to run the problem is printed after the above edit. If more or less than the required 37 entries is read than an error message is printed and the calculation stops.

The next section is an edit of the cross section , source, mesh interval, and zone data. First, a cross section edit is printed with a list of the cross sections read followed by a consistency check. If the problem involves a great amount of inelastic scattering, then this check can be ignored. Second, if a source problem is run, then the source distribution is printed for each group. Each group source distribution is preceded by the required number of entries. Third, an edit of the mesh intervals is performed for both the radial and axial points. Again, each edit is preceded by the required number of entries. Finally, an edit of the zone numbers by mesh interval followed by an edit of the material numbers by zone is printed. Each of the above edits is preceded by the required number of entries.

The next section consists of an edit of the fission spectrum and mixing table data. The fission spectrum is printed for all groups. Next, the mixing table is printed for the I0, I1, and I2 arrays. Again, the required number of entries is printed.

The next section consists of a map of both the zone numbers by mesh interval and the material numbers by fine mesh interval. This provides a means to get a picture of the problem. If more than approximately 50 mesh intervals are used in the radial direction, then the printed inventory will leave off the excess due to problems with printing off the page. This will cause an error, which will not stop further running of the problem.

The next section is a brief edit of the time in days that the problem was started, of the mixing table in easy to read format, and of the cross sections (NPRT≥2) input into the problem for each group.

After printing out the input edits above, the number of inner iterations per outer iteration with the associate eigenvalue, eigenvalue slope, and lambda after the outer iteration are printed. This is followed by a balance table, which lists the number of fissions, in-scattering neutrons, out-scattering neutrons, absorptions, and leakage for each group. The total for each neutron process follows the above. The neutron multiplication constant (not k_{eff}) is printed, followed by an edit of the radii, average radii, axii, and average axii dimensions.

Next, if NPRT >2, then the flux by mesh interval for each group followed by the total flux by mesh interval is printed. This is followed by the power density (MWT/liter) for each mesh interval.

To end the problem (if no burnup is required), a brief edit of both the mass (kilograms) and volume (liters) inventory for each zone is printed followed by their totals. This concludes the problem.

If a burnup run is called for, then the amount of days in the burnup followed by another eigenvalue edit and balance table edit is printed. Next, a brief edit of the burnup input dat is printed in easy to read format. Finally, the absorption and fission

rates for each material burned by zone is printed with the number density of each material called in the burnup.

4. PROGRAMMER'S GUIDE TO 2DBTOR

4.1. 2DBTOR Files Description

Logical Unit	Name	Format	Usage
3	FOR3.DAT	Unformatted	Cross section storage
4	FOR4.DAT	Unformatted	Scratch storage
5	TORACT5.DAT	Formatted	Input

6	TORACT5.OUT	Formatted	Output
8	FOR8.DAT	Unformatted	Flux storage
9	FOR9.DAT	Unformatted	Source storage
10	FOR10.DAT	Unformatted	Scratch storage
11	FOR11.DAT	Unformatted	Scratch storage
12	FOR12.DAT	Unformatted	Scratch storage
14	FOR14.DAT	Unformatted	Input of a flux dump
15	FOR15.DAT	Unformatted	Scratch storage
16	FOR16.DAT	Unformatted	Output of a flux dump

4.2 2DBTOR Subroutines

Name	Function
CALC	Main program
INP	Controls reading and printing of all input dat, computes variable dimension pointers, and computes program constants
ERR02	Prints error messages
S860	Reads cross sections from cards, performs adjoint reversals if required, and writes cross section tape
S862	Reads input fluxes and prepares a flux tape
S864	Reads input source and prepares a source tape
REAG2	Reads floating point data

Name	Function
REAI2	Reads integer data
RREAG2	Reads toroidal data (major radius)
MAPR	Produces a picture by zone and by material
INIT	Performs adjoint reversals, mixes cross sections, modifies geometry, and calculates areas, volumes, and fission neutrons
CLEAR	Sets an array of a specified length to a constant
FISCAL	Calculates fission sums and performs outer iteration normalization
S8830	Prints time, eigenvalue, lambda, etc. after each outer iteration
OUTER	Performs a complete outer iteration
INNER	Calculates flux in a specified group
INNER1	Calculates coefficients for the flux equation
INNERT	Calculates coefficients for the flux equation in triangular geometry
INNER2	Calculates flux in a specified group
INNERP	Calculates flux in a specified group for periodic boundary conditions
IFLUXN	Normalizes flux before each group flux calculation
IFLUXL	Normalizes flux before each group flux calculation (used for toroidal geometry source problems)

Name	Function
CNNP	Performs convergence tests and computes a new eigenvalue for search calculations
S8850	Prints the monitor line, group fluxes, total flux, power density, and fission source rate
S8847	Computes and prints group totals
PRT	Prints any IM*JM array
GRAM	Calculates and print the zone volume and the mass of each material in each zone
INPB	Reads and prints the input burnup data
AVERAG	Calculates zone averaged fluxes, fission cross sections, absorption cross sections, and breeding ratio
MARCH	Calculates the time dependent isotopic concentrations

4.2.1. Subroutine Calling Sequences.

Subroutine	Called By	Calls
CALC	---	INP, INIT, FISCAL, S8830, ERR02, OUTER, CNNP, S8850, GRAM, INPB, AVERAG, MARCH
INP	CALC	S860,S862, S864, REAG2, REAI2, RREAG2, MAPR, ERR02

Subroutine	Called By	Calls
ERR02	CALC, INP, REAI2, REAG2, INIT, CNNP	--
S860	INP	--
S862	INP	REAG2
S864	INP	REAG2
REAG2	INP, S862, S864	ERR02
REAI2	INP	ERR02
RREAG2	INP	--
MAPR	INP	--
INIT	CALC	CLEAR, ERR02
CLEAR	INIT, GRAM	--
FISCAL	CALC	--
S8830	CALC, S8850	--
OUTER	CALC	INNER1, INNER, INNERP
INNER	OUTER	IFLUXN, IFLUXL
INNER1	OUTER	--
INNERT	OUTER	--
INNER2	OUTER	IFLUXN, IFLUXL
INNERP	OUTER	IFLUXN
IFLUXN	INNER, INNER2, INNERP	--
IFLUXL	INNER, INNER2	--

Subroutine	Called By	Calls
CNNP	CALC	ERR02, CLEAR
S8850	CALC	PRT, S8830, S8847
S8847	S8850	--
PRT	S8850	--
GRAM	CALC	CLEAR
INPB	CALC	--
AVERAG	CALC	--
MARCH	CALC	--

4.3. Selected Definitions of Variables and Arrays Used

Variable	Description
** INTERNAL VARIABLES **	
NINP	Input tape
NOUT	Output tape
NCR1	Cross section tape
NFLUX1	Flux tape
NSCRAT	Scratch tape
NSORCE	Source tape
ALA	Lambda
B07	Used for internal computation in FISCAL and INIT
CNT	Convergence trigger for lambda
CVT	Convergence trigger
DAY	Burnup time in days

Variable	Description
DELT	Length of time step (days)
E0(IGP)	Fission rate
E1(IGP)	Fission source
E2(IGP)	In-scatter and extraneous source
E3(IGP)	Out-scatter
E4(IGP)	Absorptions
E5(IGP)	Left leakage
E6(IGP)	Right leakage
E7(IGP)	Top leakage
E8(IGP)	Bottom leakage
E9(IGP)	Total leakage
E01	Temporary
E02	Temporary
E03	Temporary
EQ	Temporary for S852(CNNP)
EVP	Previous eigenvalue
EVPP	Eigenvalue for two iterations back
FEF	Energy released /fission (215MeV)
GBAR	Group indicator
GLH	Initial clock time in seconds
IGEP	IGE + 1
IGP	IGM + 1
IGV	Group indicator for inner and outer

Variable	Description
IHS	Position of sigma self scatter
IHT	Position of sigma transport
II	Inner iteration count for a single group
IMJM	IM*JM
IP	IM + 1
ITEMP	Temporary
ITEMP1	Temporary
ITEMP2	Temporary
ITL	Cross section table length
IZP	IZM + 1
JP	JM + 1
K07	Not used
KPAGE	Page counter for monitor print
LAP	Lambda for previous eigenvalue
LAPP	Lambda for two iterations back
LAR	Lambda for previous iteration
LC	Loop count (total II in a single outer iteration)
ML	MCR + MTP
NCON	-/0/+ = take time step of DELT/ end of problem/ take time step of DELT and read burnup data
NGOTO	Temporary
ORFP	ORF for (1-lambda) <10*EPS
P02	Outer iteration count

Variable	Description
PBAR	Temporary
SBAR	Temporary
SK7	Sum of K7 over all groups
T06	0/1 = no effect/delta calculation
T7	Alpha/velocity
T11	Previous fission total
TEMP	Temporary
TEMP1	Temporary
TEMP2	Temporary
TEMP3	Temporary
TEMP4	Temporary
TI	Time
TSD	(MW-sec)/(fissions)
V11	Total source for the group
INPUT VARIABLES	
ID(20)	ID card
A02	0/1 = flux calc./adjoint calc.
I04	Eigenvalue type (0/1/2/3/4 = source/keff/alpha/ concentration/delta/buckling)
S02	Parametric eigenvalue type (0/1/2 = none/keff/alpha)
IGM	Number of groups
NXCM	Number of downscattering terms
MCR	Number of materials from cards

Variable	Description
MTP	Number of materials from tape
G07	Inner iteration max per group (if neg., alt dir)
D05	Max number of outer iterations
MAXT	Max run time (minutes)
NPRT	Print option (0/1/2/3 = mini/midi/Xsec/fluxes)
M07	Flux guess (0/1 = none/flux from FOR14.DAT)
NPUN	Flux dump (0/1 = none/ flux dump to FOR16.DAT)
IGE	Geometry parameter (0/1/2/3/4 = X-Y/R-Z/R-θ/triangular/ toroidal)
IM	Number of radial intervals
JM	Number of axial intervals
IZM	Number of zones
MT	Total number of materials including mixes
M01	Number of mixture specifications
B01	Left B. C. (0/1 = vacuum/reflective)
B02	Right B. C. (0/1 = vacuum/reflective)
B03	Top B. C. (0/1/2 = vacuum/reflective/periodic)
B04	Bottom B. C. (0/1/2 = vacuum/reflective/periodic)
IZ	Radial zones (delta option only)
JZ	Axial zones (delta option only)
NACT	Number of activations
EV	Eigenvalue
EVM	Eigenvalue modifier

Variable	Description
S03	Parametric eigenvalue
BUCK	Buckling
LAL	Lambda lower
LAH	Lambda upper
EPS	Eigenvalue convergence criterion
EPSA	Pointwise convergence criterion
G06	Inner iteration test (if 0 no test)
POD	Parameter oscillation damper
ORF	Over-relaxation factor
S01	-/+ = power(MWT)/neuton source rate
ARRAY VARIABLES	
ATW(ML)	Material atomic weight
HOLN(ML,2)	Material name
ALAM(ML)	Decay constant (days^{-1})
C0(ITL,MT)	Cross section array for current group
N0(IM,JM)	Total flux (old)
N2(IM,JM)	Total flux (new)
A0(IP)	Radial area element
A1(IM)	Axial area element
F0(IM,JM)	Fissions (old)
F2(IM,JM)	Fissions (new)
I0(M01)	Mix number
I1(M01)	Material number for mix

Variable	Description
I2(M01)	Material density
I3(M01)	Material densities for gram calc.
K6(IGM)	Fission spectrum (effective)
K7(IGM)	Fission spectrum (input)
M0(IM,JM)	Zone numbers
M2(IM,JM)	Material numbers by zone
R0(IP)	Initial radii
R1(IP)	Current radii
R2(IM)	Radial zone numbers (delta calc. only)
R3(IZ)	Radial zone modifiers (delta calc. only)
R4(IM)	Average radii
R5(IM)	Delta-R
S2(IM,JM)	Fixed source
V0(IM,JM)	Volume elements
V7(IGM)	Neutron velocities
Z0(JP)	Initial axii
Z1(JP)	Current axii
Z2(JM)	Axial zone numbers (delta calc. only)
Z3(JZ)	Axial zone modifiers (delta calc. only)
Z4(JM)	Average axii
Z5(JM)	Delta-Z
CXS(IM,JM,3)	Constants involving cross sections for flux calculation
VOL(IZM)	Zone volume (liters)

Variable	Description
MASS(ML,IZM)	Material inventory in each zone
MATN(ML)	Material number for burnable isotopes
NBR(ML)	0/1/2 = none/fertile/fissile
LD(ML)	Source isotope for decay
LCN(ML,2)	Source isotopes for capture
LFN(ML,7)	Source isotope for fission
PHIB(IZM)	Zone averaged flux
AXS(ML,IZM)	Spectrum averaged absorption cross section
FXS(ML,IZM)	Spectrum averaged fission cross section
MASSP(ML,IZM)	Material inventory in each zone (previous)
CXR(JM)	Constants for right boundary
CXT(IM)	Constants for top boundary
HA(IM OR JM)	Temp storage for line inversion
PA(IM OR JM)	Temp storage for line inversion

5. REFERENCES

1. W. W. Little, Jr., and R. W. Hardie. 2DB User's Manual, BNWL - 831 Revision 1, Batelle Pacific Northwest Laboratory, Richland, Washington (Unpublished).

APPENDIX H

2DBTOR LISTING

```
C    PROGRAM 2DB
C    2              TAPE5=INPUT,TAPE6=OUTPUT,TAPE3=MCR1,TAPE4=MSCRAT,   CALC  <
C    3              TAPE8=MFLUX1,TAPE9=MSORCE,                          CALC  <
C    4              TAPE14=MFLXI,TAPE16=MFLXD,TAPE15=MXS)               CALC  <
C     * * * * *  DESCRIPTION OF SUBROUTINES  * * * * *                  CALC  9
C                                                                       CALC 10
C   CALC       MAIN PROGRAM--SETS UP TAPE UNITS AND CALLS INP, INIT,    CALC 11
C              FISCAL, S8830, ERRO2, OUTER, CNNP, S8850, GRAM,          CALC 12
C              INPB, AVERAG, AND MARCH.                                 CALC 13
C                                                                       CALC 14
C   INP        SUBROUTINE TO CONTROL THE READING AND PRINTING OF ALL    CALC 15
C              INPUT DATA, COMPUTE VARIABLE DIMENSION POINTERS AND      CALC 16
C              PROGRAM CONSTANTS.  INP IS CALLED BY CALC AND CALLS      CALC 1<
C              S860, S862, S864, REAG2, REAI2, MAPR, AND ERRO2.         CALC 18
C                                                                       CALC 19
C   ERRO2      ERRO2 IS USED TO PRINT AN ERROR MESSAGE.  IT IS CALLED   CALC 20
C              BY CALC, INP, REAI2, REAG2, INIT, AND CNNP.              CALC 21
C                                                                       CALC 22
C   S860       SUBROUTINE TO READ CROSS SECTIONS FROM CARDS, PERFORM    CALC 23
C              ADJOINT REVERSALS IF REQUIRED, AND WRITE CROSS SECTION   CALC 24
C              TAPE.  S860 IS CALLED BY INP.                            CALC 25
C                                                                       CALC 26
C   S862       S862 READS INPUT FLUXES AND PREPARES A FLUX TAPE.  IT IS CALC 27
C              CALLED BY INP AND CALLS REAG2.                           CALC 28
C                                                                       CALC 29
C   S864       S864 READS INPUT SOURCE AND PREPARES A SOURCE TAPE.  IT  CALC 30
C              IS CALLED BY INP AND CALLS REAG2.                        CALC 31
C                                                                       CALC 32
C   REAG2      SUBROUTINE TO READ FLOATING POINT DATA.  REAG2 IS CALLED CALC 33
C              BY INP, S862, AND S864.  REAG2 CALLS ERROR.              CALC 34
C                                                                       CALC 35
C   REAI2      SUBROUTINE TO READ INTEGER DATA.  REAI2 IS CALLED BY INP CALC 36
C              AND CALLS ERRO2.                                         CALC 37
C                                                                       CALC 38
C   MAPR       SUBROUTINE TO PRODUCE A PICTURE BY ZONE AND              CALC 39
C              MATERIAL.  MAPR IS CALLED BY INP.                        CALC 40
C                                                                       CALC 41
C   INIT       INIT PERFORMS ADJOINT REVERSALS(S806), MIXES CROSS       CALC 42
```

```
C               SECTIONS(S807), MODIFIES GEOMETRY(S810), AND CALCULATES CALC  43
C               AREAS AND VOLUMES(S811), AND FISSION NEUTRONS(S821).    CALC  44
C               INIT IS CALLED BY CALC AND CALLS CLEAR AND ERRO2.       CALC  45
C                                                                       CALC  46
C   CLEAR       CLEAR SETS AN ARRAY OF A SPECIFIED LENGTH TO A GIVEN    CALC  47
C               CONSTANT.  THE SUBROUTINE IS CALLED BY INIT AND GRAM.   CALC  48
C                                                                       CALC  49
C   FISCAL      CALCULATES FISSION SUMS(S822) AND PERFORMS              CALC  50
C               NORMALIZATION(S823).  FISCAL IS CALLED BY CALC.         CALC  51
C                                                                       CALC  52
C   S8830       S8830 IS THE MONITOR PRINT SUBROUTINE--PRINTS TIME,     CALC  53
C               EIGENVALUE, LAMBDA, ETC. AFTER EACH OUTER ITERATION.    CALC  54
C               IT IS CALLED BY CALC AND S8850.                         CALC  55
C                                                                       CALC  56
C   OUTER       PERFORMS A COMPLETE OUTER ITERATION.  CALLS INNER1,     CALC  57
C               INNER, AND INNERP.  OUTER IS CALLED BY CALC.            CALC  58
C                                                                       CALC  59
C   INNER       CALCULATES THE FLUX IN SPECIFIED GROUP.  IT IS CALLED   CALC  60
C               BY OUTER AND CALLS IFLUXN.                              CALC  61
C                                                                       CALC  62
C   INNER1      CALCULATES COEFFICIENTS FOR THE FLUX EQUATION.  INNER1  CALC  63
C               IS CALLED BY OUTER.                                     CALC  64
C                                                                       CALC  65
C   INNERT      INNERT CALCULATES COEFFICIENTS FOR THE FLUX EQUATION FOR CALC  66
C               TRIANGULAR GEOMETRY.  INNERT IS CALLED BY OUTER.        CALC  67
C                                                                       CALC  68
C   INNER2      CALCULATES THE FLUX IN SPECIFIED GROUP.  IT IS CALLED   CALC  69
C               BY OUTER AND CALLS IFLUXN.                              CALC  70
C                                                                       CALC  71
C   INNERP      CALCULATES THE FLUX IN SPECIFIED GROUP FOR PERIODIC B. C.CALC  72
C               IT IS CALLED BY OUTER AND CALLS IFLUXN.                 CALC  73
C                                                                       CALC  74
C   IFLUXN      SUBROUTINE TO NORMALIZE THE FLUXES BEFORE EACH          CALC  75
C               GROUP FLUX CALCULATION.  IT IS CALLED BY INNER, INNER2, CALC  76
C               AND INNERP.                                             CALC  77
C                                                                       CALC  78
C   CNNP        PERFORMS CONVERGENCE TESTS(S851) AND COMPUTES A NEW     CALC  79
C               EIGENVALUE FOR SEARCH OPTIONS(S852).  CNNP IS CALLED    CALC  80
```

```
C               BY CALC AND CALLS ERRO2 AND CLEAR.                      CALC  81
C                                                                       CALC  82
C     S8850     FINAL PRINT SUBROUTINE--PRINTS THE MONITOR LINE,        CALC  83
C               GROUP FLUXES, TOTAL FLUX, POWER DENSITY, AND FISSION    CALC  84
C               SOURCE RATE.  IT IS CALLED BY CALC AND CALLS PRT, S8830, CALC  85
C               AND S8847.                                              CALC  86
C                                                                       CALC  87
C     S8847     SUBROUTINE TO COMPUTE AND PRINT GROUP TOTALS.  S8847 IS CALC  88
C               CALLED BY S8850.                                        CALC  89
C                                                                       CALC  90
C     PRT       SUBROUTINE TO PRINT ANY IM*JM ARRAY.  IT IS CALLED BY   CALC  91
C               S8850.                                                  CALC  92
C                                                                       CALC  93
C     GRAM      CALCULATES AND PRINTS THE MASS OF EACH MATERIAL IN EACH CALC  94
C               ZONE AND THE ZONE VOLUME.  IT IS CALLED BY CALC AND     CALC  95
C               CALLS CLEAR.                                            CALC  96
C                                                                       CALC  97
C     INPB      SUBROUTINE TO READ AND PRINT THE INPUT BURNUP DATA.  IT CALC  98
C               IS CALLED BY CALC.                                      CALC  99
C                                                                       CALC 100
C     AVERAG    AVERAG CALCULATES ZONE AVERAGED FLUXES, FISSION CROSS   CALC 101
C               SECTIONS, ABSORPTION CROSS SECTIONS, AND BREEDING RATIO. CALC 102
C               THE SUBROUTINE IS CALLED BY CALC.                       CALC 103
C                                                                       CALC 104
C     MARCH     SUBROUTINE TO CALCULATE THE TIME DEPENDENT ISOTOPIC     CALC 105
C               CONCENTRATIONS.  MARCH IS CALLED BY CALC.               CALC 106
C                                                                       CALC 107
C      * * * * *  INTERNAL VARIABLES  * * * * *                         CALC 108
C                                                                       CALC 109
C      NINP         INPUT TAPE                                          CALC 110
C      NOUT         OUTPUT TAPE                                         CALC 111
C      NCR1         CROSS SECTION TAPE                                  CALC 112
C      NFLUX1       FLUX TAPE                                           CALC 113
C      NSCRAT       SCRATCH TAPE                                        CALC 114
C      NCR1         SOURCE TAPE                                         CALC 115
C      ALA          LAMBDA                                              CALC 116
C      B07          USED FOR INTERNAL COMPUTATION IN FISCAL AND INIT    CALC 117
C      CNT          CONVERGENCE TRIGGER FOR LAMBDA                      CALC 118
```

```
C     CVT          CONVERGENCE TRIGGER                                CALC 119
C     DAY          BURNUP TIME IN DAYS                                CALC 120
C     DELT         LENGTH OF TIME STEP (DAYS)                         CALC 121
C     E0(IGP)      FISSION RATE                                       CALC 122
C     E1(IGP)      FISSION SOURCE                                     CALC 123
C     E2(IGP)      IN-SCATTER (AND EXTRANEOUS SOURCE)                 CALC 124
C     E3(IGP)      OUT-SCATTER                                        CALC 125
C     E4(IGP)      ABSORPTIONS                                        CALC 126
C     E5(IGP)      LEFT LEAKAGE                                       CALC 127
C     E6(IGP)      RIGHT LEAKAGE                                      CALC 128
C     E7(IGP)      TOP LEAKAGE                                        CALC 129
C     E8(IGP)      BOTTOM LEAKAGE                                     CALC 130
C     E9(IGP)      TOTAL LEAKAGE                                      CALC 131
C     E01          TEMPORARY                                          CALC 132
C     E02          TEMPORARY                                          CALC 133
C     E03          TEMPORARY                                          CALC 134
C     EQ           TEMPORARY FOR S852  (CNNP)                         CALC 135
C     EVP          PREVIOUS EIGENVALUE                                CALC 136
C     EVPP         EIGENVALUE FOR TWO ITERATIONS BACK                 CALC 137
C     FEF          ENERGY RELEASED PER FISSION (=215 MEV)             CALC 138
C     GBAR         GROUP INDICATOR FOR TAPE MOTION IN S824  (OUTER)   CALC 139
C     GLN          INITIAL CLOCK TIME IN SECONDS (INTEGER)
C     IGEP         IGE + 1                                            CALC 141
C     IGP          IGM + 1                                            CALC 142
C     IGV          GROUP INDICATOR FOR INNER AND OUTER                CALC 143
C     IHS          POSITION OF SIGMA SELF SCATTER                     CALC 144
C     IHT          POSITION OF SIGMA TRANSPORT                        CALC 145
C     II           INNER ITERATION COUNT FOR A SINGLE GROUP           CALC 146
C     IMJM         IM*JM                                              CALC 147
C     IP           IM + 1                                             CALC 148
C     ITEMP        TEMPORARY                                          CALC 149
C     ITEMP1       TEMPORARY                                          CALC 150
C     ITEMP2       TEMPORARY                                          CALC 151
C     ITL          CROSS SECTION TABLE LENGTH                         CALC 152
C     IZP          IZM + 1                                            CALC 153
C     JP           JM + 1                                             CALC 154
C     K07          NOT USED                                           CALC 155
C     KPAGE        PAGE COUNTER FOR MONITOR PRINT                     CALC 156
```

```
C     LAP          LAMBDA FOR PREVIOUS EIGENVALUE                    CALC 157
C     LAPP         LAMBDA FOR TWO ITERATIONS BACK                    CALC 158
C     LAR          LAMBDA FOR PREVIOUS ITERATION                     CALC 159
C     LC           LOOP COUNT  (TOTAL II IN A SINGLE OUTER ITERATION)  CALC 160
C     ML           MCR + MTP                                         CALC 161
C     NCON         NEG/ZERO/POS=TAKE TIME STEP OF DELT/END OF PROBLEM/ CALC 162
C                  TAKE TIME STEP OF DELT AND READ BURNUP DATA       CALC 163
C     NGOTO        TEMPORARY                                         CALC 164
C     NPRT         0/1/2/3=MINI/MIDI/CROSS SECTION/FLUX PRINT
C     ORFP         ORF FOR 1 - LAMBDA LESS THAN 10*EPS               CALC 166
C     P02          OUTER ITERATION COUNT                             CALC 167
C     PBAR         TEMPORARY                                         CALC 168
C     SBAR         TEMPORARY                                         CALC 169
C     SK7          SUM OF K7 OVER ALL GROUPS                         CALC 170
C     T06          0/1=NOT DELTA/DELTA CALCULATION                   CALC 171
C     T7           ALPHA/VELOCITY                                    CALC 172
C     T11          PREVIOUS FISSION TOTAL                            CALC 173
C     TEMP         TEMPORARY                                         CALC 174
C     TEMP1        TEMPORARY                                         CALC 175
C     TEMP2        TEMPORARY                                         CALC 176
C     TEMP3        TEMPORARY                                         CALC 177
C     TEMP4        TEMPORARY                                         CALC 178
C     TI           TIME                                              CALC 179
C     TSD          (MW-SEC)/(FISSIONS)                               CALC 180
C     V11          TOTAL SOURCE FOR THE GROUP                        CALC 181
C                                                                    CALC 182
C     * * * * *  INPUT VARIABLES (CARDS 1-5)  * * * * *              CALC 183
C                                                                    CALC 184
C     ID(20)       IDENTIFICATION CARD                               CALC 185
C     A02          0/1=FLUX CALCULATION/ADJOINT CALCULATION          CALC 187
C     I04          EIGENVALUE TYPE (1/2/3/4/5=KEFF/ALPHA/CONCENTRATION/CALC 188
C                  DELTA/BUCKLING)                                   CALC 189
C     S02          PARAMETRIC EIGENVALUE TYPE  (0/1/2=NONE/KEFF/ALPHA) CALC 190
C     IGM          NUMBER OF GROUPS                                  CALC 191
C     NXCM         NUMBER OF DOWNSCATTERING TERMS                    CALC 192
C     MCR          NUMBER OF MATERIALS FROM CARDS/TAPE (+N/-N)
C     MTP          NUMBER OF MATERIALS FROM TAPE                     CALC 194
C     G07          INNER ITERATION MAX PER GROUP (IF NEG, ALT DIR)
```

```
C     S04          INVERSION DIRECTION (0/1=NO EFFECT/ALTERNATE DIRECTICALC 199
C     D05          MAXIMUM NUMBER OF OUTER ITERATIONS
C     MAXT         MAXIMUM TIME (MINUTES)
C     NPRT         PRINT OPTION (0/1/2=MINI/MIDI/MAXI)
C     M07          FLUX GUESS (0/1=NONE/INPUT FROM TAPE 14)
C     NPUN         FLUX DUMP (0/1=NONE/DUMP TO TAPE 16)
C     IGE          GEOMETRY  (0/1/2/3=X-Y/R-Z/R-THETA/TRIANGULAR)       CALC 202
C     IM           NUMBER OF RADIAL INTERVALS                           CALC 203
C     JM           NUMBER OF AXIAL INTERVALS                            CALC 204
C     IZM          NUMBER OF MATERIAL ZONES                             CALC 205
C     MT           TOTAL NUMBER OF MATERIALS INCLUDING MIXES            CALC 206
C     M01          NUMBER OF MIXTURE SPECIFICATIONS                     CALC 207
C     B01          LEFT   BOUNDARY CONDITION  (0/1=VACUUM/REFLECTIVE)   CALC 208
C     B02          RIGHT  BOUNDARY CONDITION  (0/1=VACUUM/REFLECTIVE)   CALC 209
C     B03          TOP    BOUNDARY CONDITION  (0/1/2=VAC/REFL/PERIODIC)CALC 210
C     B04          BOTTOM BOUNDARY CONDITION  (0/1/2=VAC/REFL/PERIODIC)CALC 211
C     IZ           RADIAL ZONES  (DELTA-OPTION ONLY)                    CALC 212
C     JZ           AXIAL ZONES  (DELTA-OPTION ONLY)                     CALC 213
C     NACT         NUMBER OF ACTIVATIONS                                CALC 214
C     EV           FIRST EIGENVALUE GUESS                               CALC 220
C     EVM          EIGENVALUE MODIFIER                                  CALC 221
C     S03          PARAMETRIC EIGENVALUE                                CALC 222
C     BUCK         BUCKLING                                             CALC 223
C     LAL          LAMBDA LOWER                                         CALC 224
C     LAH          LAMBDA UPPER                                         CALC 225
C     EPS          EIGENVALUE CONVERGENCE CRITERIA                      CALC 226
C     EPSA         POINTWISE CONVERGENCE CRITERIA                       CALC 227
C     G06          INNER ITERATION TEST  (IF ZERO, NO TEST)             CALC 228
C     POD          PARAMETER OSCILLATION DAMPER                         CALC 229
C     ORF          OVER-RELAXATION FACTOR                               CALC 230
C     S01          NEG/POS=POWER (MWT)/NEUTRON SOURCE RATE              CALC 231
C                                                                       CALC 232
C     * * * * *  SUBSCRIPTED VARIABLES  * * * * *                       CALC 233
C                                                                       CALC 234
C     ATW(ML)      MATERIAL ATOMIC WEIGHT                               CALC 235
C     HOLN(ML,2)   MATERIAL NAME                                        CALC 236
C     ALAM(ML)     DECAY CONSTANT  (DAYS-1)                             CALC 237
C     CO(ITL,MT)   CROSS SECTION ARRAY FOR CURRENT GROUP                CALC 238
```

```
C     NO(IM,JM)      TOTAL FLUX  (OLD)                                  CALC 239
C     N2(IM,JM)      TOTAL FLUX  (NEW)                                  CALC 240
C     A0(IP)         RADIAL AREA ELEMENT                                CALC 241
C     A1(IM)         AXIAL  AREA ELEMENT                                CALC 242
C     F0(IM,JM)      FISSIONS  (OLD)                                    CALC 243
C     F2(IM,JM)      FISSIONS  (NEW)                                    CALC 244
C     I0(M01)        MIX NUMBER                                         CALC 245
C     I1(M01)        MATERIAL NUMBER FOR MIX                            CALC 246
C     I2(M01)        MATERIAL DENSITY                                   CALC 247
C     I3(M01)        MATERIAL DENSITIES FOR GRAM CALCULATION            CALC 248
C     K6(IGM)        FISSION SPECTRUM  (EFFECTIVE)                      CALC 249
C     K7(IGM)        FISSION SPECTRUM  (INPUT)                          CALC 250
C     M0(IM,JM)      ZONE NUMBERS                                       CALC 251
C     M2(IZM)        MATERIAL NUMBERS BY ZONE                           CALC 252
C     R0(IP)         INITIAL RADII                                      CALC 253
C     R1(IP)         CURRENT RADII                                      CALC 254
C     R2(IM)         RADIAL ZONE NUMBERS  (DELTA CALCULATION ONLY)      CALC 255
C     R3(IZ)         RADIAL ZONE MODIFIERS  (DELTA CALCULATION ONLY)    CALC 256
C     R4(IM)         AVERAGE RADII                                      CALC 257
C     R5(IM)         DELTA-R                                            CALC 258
C     S2(IM,JM)      FIXED SOURCE                                       CALC 259
C     V0(IM,JM)      VOLUME ELEMENTS                                    CALC 260
C     V7(IGM)        NEUTRON VELOCITIES                                 CALC 261
C     Z0(JP)         INITIAL AXII                                       CALC 262
C     Z1(JP)         CURRENT AXII                                       CALC 263
C     Z2(JM)         AXIAL ZONE NUMBERS  (DELTA CALCULATION ONLY)       CALC 264
C     Z3(JZ)         AXIAL ZONE MODIFIERS  (DELTA CALCULATION ONLY)     CALC 265
C     Z4(JM)         AVERAGE AXII                                       CALC 266
C     Z5(JM)         DELTA-Z                                            CALC 267
C     CXS(IM,JM,3)   CONSTANTS INVOLVING CROSS SECTIONS FOR FLUX CALC.  CALC 268
C     VOL(IZM)       ZONE VOLUME (LITERS)                               CALC 269
C     MASS(ML,IZM)   MATERIAL INVENTORY IN EACH ZONE                    CALC 270
C     MATN(ML)       MATERIAL NUMBER FOR BURNABLE ISOTOPES              CALC 271
C     NBR(ML)        0/1/2=NO EFFECT/FERTILE/FISSILE ISOTOPE            CALC 272
C     LD(ML)         SOURCE ISOTOPE FOR DECAY                           CALC 273
C     LCN(ML,2)      SOURCE ISOTOPES FOR CAPTURE                        CALC 274
C     LFN(ML,7)      SOURCE ISOTOPES FOR FISSION                        CALC 275
C     PHIB(IZM)      ZONE AVERAGED FLUX                                 CALC 276
```

```
C     AXS(ML,IZM)   SPECTRUM AVERAGED ABSORPTION CROSS SECTION          CALC 277
C     FXS(ML,IZM)   SPECTRUM AVERAGED FISSION CROSS SECTION             CALC 278
C     MASSP(ML,IZM) MATERIAL INVENTORY IN EACH ZONE (PREVIOUS)          CALC 279
C     CXR(JM)       CONSTANTS FOR RIGHT BOUNDARY                        CALC 280
C     CXT(IM)       CONSTANTS FOR TOP BOUNDARY                          CALC 281
C     HA(IM OR JM)  TEMP STORAGE FOR LINE INVERSION                     CALC 282
C     PA(IM OR JM)  TEMP STORAGE FOR LINE INVERSION                     CALC 283
C                                                                       CALC 284
      INCLUDE 'ABC.FOR'

      COMMON/PACKED/A(50000)
      OPEN(UNIT=3,STATUS='SCRATCH',FORM='UNFORMATTED')
      OPEN(UNIT=4,STATUS='SCRATCH',FORM='UNFORMATTED')
C   USE BELOW ON A VAX                                                 HRA2
C      OPEN(UNIT=3,FILE='FOR3.DAT',STATUS='SCRATCH',FORM='UNFORMATTED')  HRA2
C      OPEN(UNIT=4,FILE='FOR4.DAT',STATUS='SCRATCH',FORM='UNFORMATTED')  HRA2
      OPEN(UNIT=5,FILE='toract5.DAT',STATUS='OLD',FORM='FORMATTED')
      OPEN(UNIT=6,FILE='toract5.OUT',STATUS='UNKNOWN',FORM='FORMATTED')
      OPEN(UNIT=8,STATUS='SCRATCH',FORM='UNFORMATTED')
      OPEN(UNIT=9,STATUS='SCRATCH',FORM='UNFORMATTED')
C   USE BELOW ON A VAX                                                 HRA2
C      OPEN(UNIT=8,FILE='FOR8.DAT',STATUS='SCRATCH',FORM='UNFORMATTED')  HRA2
C      OPEN(UNIT=9,FILE='FOR9.DAT',STATUS='SCRATCH',FORM='UNFORMATTED')  HRA2
      OPEN(UNIT=10,FILE='FOR10.DAT',STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN(UNIT=11,FILE='FOR11.DAT',STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN(UNIT=12,FILE='FOR12.DAT',STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN(UNIT=14,FILE='FOR14.DAT',STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN(UNIT=15,FILE='FOR15.DAT',STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN(UNIT=16,FILE='FOR16.DAT',STATUS='UNKNOWN',FORM='UNFORMATTED')

1     CONTINUE
      REWIND 3                                                         CALC 291
      REWIND 4                                                         CALC 292
      REWIND 8                                                         CALC 293
      REWIND 9                                                         CALC 294
      CALL INP(BIGR)                                                   HRA2 295
102   CALL INIT(A(LK6), A(LK7), A(LI0), A(LI1), A(LI2), A(LM0), A(LM2), CALC 296
     1          A(LN0), A(LR0), A(LR1), A(LR2), A(LR3), A(LR4), A(LR5), CALC 297
```

```
     2          A(LZ0), A(LZ1), A(LZ2), A(LZ3), A(LZ4), A(LZ5), A(LA0), CALC 298
     3          A(LA1), A(LF0), A(LC0), A(LV0),    ITL,     IM, A(LV7), CALC 299
     4          JM, MT, ML, A(LGAM), A(LHOLN))
      CALL FISCAL (A(LN0),A(LF0) ,A(LV0) ,A(LC0) ,A(LK6) ,              CALC 301
     2            A(LM0) ,A(LM2),ITL,MT)                               CALC 302
C     CALL MONITOR PRINT                                               CALC 303
101   CALL S8830                                                       CALC 304
      GO TO (100, 106, 106, 107), NGOTO                                CALC 30#
106   CALL ERRO2('*MONPR',106,1)                                       HVX
C     PERFORM AN OUTER ITERATION                                       CALC 307
107   CALL OUTER( A(LA0), A(LA1), A(LC0), A(LF0), A(LK6),              CALC 308
     1            A(LM0), A(LM2), A(LN0), A(LN2),                      CALC 309
     2             A(LS2), A(LV0), A(LV7), A(LZ5),                     CALC 310
     3            A(LF2), ITL, MT, A(LCXS), IM, JM, A(LR5), A(LR4),    CALC 311
     4     A(LZ4), A(LCXR), A(LCXT), A(LHA), A(LPA),A(LR1),A(LZ1))     HRA2
C     PERFORM FISSION CALCULATION                                      CALC 313
      CALL FISCAL (A(LN0),A(LF0) ,A(LV0) ,A(LC0) ,A(LK6) ,              CALC 314
     2             A(LM0) ,A(LM2),ITL,MT)                              CALC 315
C     PERFORM CONVERGENCE AND NEW PARAMETER CALCULATIONS               CALC 316
      CALL CNNP (A(LF2), A(LK6))                                       CALC 317
      GO TO (100, 101, 102), NGOTO                                     CALC 31#
C     100/101/102=FINAL PRINT/MONITOR PRINT/SEARCH CALCULATION         CALC 319
  100 CALL S8850(A(LF2),A(LN2),A(LR1),A(LZ1),A(LR4),A(LZ4),A(LV7),     CALC 320
     1 IM,JM,A(LN2),A(LC0),A(LN0),A(LM0),A(LM2),A(LF0),ITL,MT)
      CALL GRAM(A(LMASS), A(LVOL),  A(LATW), A(LHOLN), IM, JM,         CALC 323
     1          A(LM0), A(LM2), A(LV0), A(LI0), A(LI1), A(LI2), ML,    CALC 324
     2          A(LI3))                                                CALC 325
      CALL INPB(A(LMATN),A(LNBR),A(LLD),A(LLCN),A(LLFN),A(LALAM),      CALC 326
     1          A(LHOLN),ML,A(LI2))                                    CALC 327
      IF (NCON) 170,1,170
170   CALL AVERAG(A(LPHIB),A(LAXS),A(LFXS),A(LMATN),A(LMASS),A(LATW),  WLP
     1            A(LVOL),A(LC0),A(LN2),A(LM0),A(LV0),A(LHOLN),ML,ITL, CALC 330
C    2             A(LNBR),A(LT6),A(LT8),A(LNTWON))                    HRA2
     2            A(LNBR),A(LACT),NACT,A(LACPOS))                      HRA3
      IF (DELT) 180,1,180                                              WLP
180   CONTINUE                                                         WLP
      CALL MARCH(A(LPHIB),A(LMATN),A(LFXS),A(LAXS),A(LVOL),A(LMASS),   WLP
     1          A(LMASSP),A(LALAM),A(LLD),A(LLCN),A(LLFN),ML,          CALC 334
```

```
     2          A(LI0),A(LI1),A(LI2),A(LM2))                           CALC 335
      GO TO 102                                                        CALC 336
      END                                                              CALC 337
      SUBROUTINE OUTER (A0, A1, C0, F0, K6, M0, M2, N0, N2,            OUTE   2
     1                 S2, V0, V7, Z5, F2, JTL,JMT,  CXS,              OUTE   3
     2             JIM, JJM,  R5,  R4,  Z4, CXR, CXT, HA, PA,R1,Z1)    HRA2
      DIMENSION XR(50),XD(50)                                          HRA2
      DIMENSION  A0(1), A1(1), F0(1), K6(1), M0(1), M2(1),             OUTE   5
     1          N0(1), N2(1),R1(1),Z1(1),   S2(1),                     HRA2   6
     2          V0(1), V7(1), Z5(1), F2(1), C0(JTL,JMT),  HA(1), PA(1),OUTE   7
     3         CXS(JIM,JJM,3), R5(1), R4(1), Z4(1), CXR(1), CXT(1)     OUTE   8
      INTEGER       GBAR, PBAR, SBAR                                   OUTE   9
       INCLUDE 'ABC.FOR'

      IGV=1                                                            OUTE  13
C     SOURCE CALCULATION                                               OUTE  14
      REWIND 12                                                        HRA2
10    CONTINUE                                                         OUTE  15
      READ(NCR1)     ((C0(I,M),I=1,ITL),M=1,MT)                        OUTE  16
C     READ(11) (XR(M),M=1,MT)                                          HRA2
      READ(11) (XD(M),M=1,MT)                                          HRA2
      DO 1110 M=1,MT                                                   HRA2
1110  CONTINUE                                                         HRA2
      IF (I04) 15,12,15                                                OUTE  17
12    READ (NSORCE) (S2(I), I = 1,IMJM)                                OUTE  18
      GO TO 30                                                         OUTE  19
15    DO 20  I=1,IMJM                                                  OUTE  20
20    S2(I)=0.                                                         OUTE  21
30    IF(A02) 60, 40, 60                                               OUTE  22
40    DO 50  I=1,IMJM                                                  OUTE  23
50    S2(I)=S2(I)+K6(IGV)*F0(I)                                        OUTE  24
      GO TO 80                                                         OUTE  25
60    DO 70   I=1,IMJM                                                 OUTE  26
      ITEMP1=M0(I)                                                     OUTE  27
      ITEMP1=M2(ITEMP1)                                                OUTE  28
70    S2(I)=S2(I)+C0(IMT-1,ITEMP1)*F0(I)                               OUTE  29
80    GBAR=IGV+IHS-ITL                                                 OUTE  30
      IF(GBAR-1)  90, 100, 100                                         OUTE  31
```

```
90     GBAR=1                                              OUTE  32
100    PBAR = IHS + IGV - 1                                OUTE  33
       IF(PBAR - ITL)  115,115,110                         OUTE  34
110    PBAR = ITL                                          OUTE  35
115    IF(GBAR - IGV)  120, 140, 140                       OUTE  36
120    READ (NSCRAT) (N2(I),I=1,IMJM)                      OUTE  37
       DO 130  I=1,IMJM                                    OUTE  38
       ITEMP1=M0(I)                                        OUTE  39
       ITEMP1=M2(ITEMP1)                                   OUTE  40
       ITEMP=ITEMP1                                        OUTE  41
       TEMP=C0(PBAR,ITEMP)                                 OUTE  42
130    S2(I)=S2(I)+N2(I)*TEMP                              OUTE  43
       GO TO 150                                           OUTE  44
140    READ (NFLUX1) (N2(I),I=1,IMJM)                      OUTE  45
150    GBAR=GBAR+1                                         OUTE  46
       PBAR=PBAR-1                                         OUTE  47
       IF(GBAR - IGV)  120, 140, 160                       OUTE  48
160    IF(IGV - IGM)  180, 170, 180                        OUTE  49
170    REWIND NCR1                                         OUTE  50
       REWIND 11                                           HRA2
       REWIND 12                                           HRA2
180    V11=0.                                              OUTE  51
       DO 190  I=1,IMJM                                    OUTE  52
       S2(I)=S2(I)*V0(I)                                   OUTE  53
190    V11=V11+S2(I)                                       OUTE  54
       E2(IGV) = V11 - E1(IGV)                             OUTE  55
C      SOURCE-ALPHA                                        OUTE  56
200    IF(I04 - 2)  210, 240, 210                          OUTE  57
210    IF(S02 - 2)  230, 220, 230                          OUTE  58
220    T7 = S03/V7(IGV)                                    OUTE  59
       GO TO 250                                           OUTE  60
230    T7 = 0.0                                            OUTE  61
       GO TO 270                                           OUTE  62
240    T7 = EV/V7(IGV)                                     OUTE  63
250    DO 260  K = 1, IZM                                  OUTE  64
       ITEMP1 = M2(K)                                      OUTE  65
260    C0(IHS, ITEMP1) = C0(IHS,ITEMP1) - T7               OUTE  66
270    CONTINUE                                            OUTE  67
```

```
C     GROUP FLUX CALCULATION                                          OUTE  68
280   II=0                                                            OUTE  69
      IF(V11) 290, 370, 290                                           OUTE  70
290   IF(IGE  - 3)  292,294,292                                       OUTE  71
292   CALL INNER1(MO, M2, CXS, VO, CO, A0, Z5, R5, R4, Z4, A1, IM, JM,  OUTE  72
     1           ITL, CXR,CXT,XR,XD,R1,Z1)                            HRA2  73
      GO TO 296                                                       OUTE  74
294   CALL INNERT(MO, M2, CXS, VO, CO, A0, Z5, R5, R4, Z4, A1, IM, JM,  OUTE  75
     1           ITL, CXR,CXT)                                        OUTE  76
      GO TO 340                                                       OUTE  77
296   IF (B03 - 1) 310, 310, 300                                      OUTE  78
300   CALL INNERP(NO, N2, CXS, S2, MO, M2, VO, CO, IM, JM, ITL, CXR,CXT,OUTE  79
     1          HA, PA)                                               OUTE  80
      GO TO 350                                                       OUTE  81
310   IF (IGE - 2) 320, 330, 320                                      OUTE  82
320   IF (S04 - 1) 325,322,325                                        OUTE  83
322   IF (P02 - 2 * (P02/2))   330,330,340                            OUTE  84
325   IF (IM - JM) 330,330,340                                        OUTE  85
330   CALL INNER(NO, N2, CXS, S2, MO, M2, VO, CO, IM, JM, ITL, CXR, CXT,OUTE  86
     1           HA, PA,XR,XD)                                        HRA2
      GO TO 350                                                       OUTE  88
340   CALLINNER2(NO, N2, CXS, S2, MO, M2, VO, CO, IM, JM, ITL, CXR, CXT,OUTE  89
     1           HA, PA)
350   DO 360  K = 1,IZM                                               OUTE  91
      ITEMP1 = M2(K)                                                  OUTE  92
360   CO(IHS,ITEMP1) = CO(IHS,ITEMP1) + T7                            OUTE  93
      GO TO 390                                                       OUTE  94
370   DO 380 I=1,IMJM                                                 OUTE  95
      N2(I)=0.                                                        OUTE  96
380   NO(I)=0.                                                        OUTE  97
390   CONTINUE                                                        OUTE  98
      WRITE (NSCRAT) (N2(I),I=1,IMJM)                                 OUTE  99
      REWIND NSCRAT                                                   OUTE 106
      SBAR=IGV-(ITL-IHS)                                              OUTE 107
      IF(SBAR) 440, 440, 420                                          OUTE 108
420   DO 430 IS=1,SBAR                                                OUTE 109
  430 READ(NSCRAT)                                                    OUTE 110
440   CONTINUE                                                        OUTE 111
```

```
      IF(V11) 450, 500, 450                                     OUTE 112
450   IF(A02) 460, 480, 460                                     OUTE 113
460   E0(IGV)=0.0                                               OUTE 114
      DO 470  I=1,IMJM                                          OUTE 115
      ITEMP1=M0(I)                                              OUTE 116
      ITEMP1=M2(ITEMP1)                                         OUTE 117
      E0(IGV)=E0(IGV) + C0(IHT-3,ITEMP1)*N2(I)*V0(I)            HRA2 118
470   F2(I)=F2(I)+K6(IGV)*N2(I)                                 OUTE 119
      GO TO 500                                                 OUTE 120
480   E0(IGV)=0.0                                               OUTE 121
      DO 490  I=1,IMJM                                          OUTE 122
      ITEMP1=M0(I)                                              OUTE 123
      ITEMP1=M2(ITEMP1)                                         OUTE 124
      E0(IGV)=E0(IGV) + C0(IHT-3,ITEMP1)*N2(I)*V0(I)            HRA2 125
490   F2(I)=F2(I)+C0(IHT-1,ITEMP1)*N2(I)                        OUTE 126
500   CONTINUE                                                  OUTE 127
      IGV=IGV+1                                                 OUTE 128
      IF(IGV-IGM) 10, 10, 510                                   OUTE 129
510   T11 = E1(IGP)                                             OUTE 130
C     SWITCH TAPE DESIGNATIONS                                  OUTE 131
      REWIND NCR1                                               OUTE 132
      REWIND NSCRAT                                             OUTE 133
      REWIND NFLUX1                                             OUTE 134
      REWIND 11                                                 HRA2
      REWIND 12                                                 HRA2
      ITEMP = NSCRAT                                            OUTE 135
      NSCRAT = NFLUX1                                           OUTE 136
      NFLUX1 = ITEMP                                            OUTE 137
      IF (I04) 514,512,514                                      OUTE 138
512   REWIND NSORCE                                             OUTE 139
514   CONTINUE                                                  OUTE 140
C                                                               OUTE 141
C     OVER-RELAX FISSION SOURCE                                 OUTE 142
      ORFF = 1.0 + .6*(ORF - 1.0)                               OUTE 143
      E02 = .0                                                  OUTE 144
      IF(A02)  520,580,520                                      OUTE 145
520   E1(IGP) = .0                                              OUTE 146
C     FOR ADJOINT CALCULATION, S2(I) STORES ORFED F2(I)         OUTE 147
```

```
      DO 522  I=1,IMJM                                          OUTE 148
522   S2(I) = FO(I) + ORFF*(F2(I) - FO(I))                      OUTE 149
      DO 540  IIG = 1,IGM                                       OUTE 150
      READ(NCR1)  ((CO(I,J), I=1,ITL), J=1,MT)                  OUTE 151
      E1(IIG) = .0                                              OUTE 152
      DO 530  I=1,IMJM                                          OUTE 153
      ITEMP = MO(I)                                             OUTE 154
      ITEMP = M2(ITEMP)                                         OUTE 155
      E1(IIG) = E1(IIG) + CO(IHT-1,ITEMP)*F2(I)*VO(I)           HRA2 156
530   E02 = E02         + CO(IHT-1,ITEMP)*S2(I)*VO(I)           HRA2 157
540   E1(IGP) = E1(IGP) + E1(IIG)                               OUTE 158
      TEMP1 = E1(IGP)/E02                                       OUTE 159
      DO 550  I=1,IMJM                                          OUTE 160
550   FO(I) = TEMP1*S2(I)                                       OUTE 161
      REWIND NCR1                                               OUTE 162
      REWIND 11                                                 HRA2
      REWIND 12                                                 HRA2
      GO TO 620                                                 OUTE 163
580   E01 = 0.0                                                 OUTE 164
      DO 590  I=1,IMJM                                          OUTE 165
      E01 = E01 + VO(I)*F2(I)                                   OUTE 166
      F2(I) = FO(I) + ORFF*(F2(I) - FO(I))                      OUTE 167
590   E02 = E02 + VO(I)*F2(I)                                   OUTE 169
      TEMP1=0.                                                  OUTE 170
      IF(E02.NE.0.0)TEMP1=E01/E02                               OUTE 171
      DO 600  I=1,IMJM                                          OUTE 172
600   FO(I) = TEMP1*F2(I)                                       OUTE 173
      DO 610  IIG = 1,IGM                                       OUTE 174
610   E1(IIG) = K6(IIG)*E01                                     OUTE 175
      IF(I04) 620,609,620                                       OUTE 176
609   TEMP1 = .0
      IF(E01.EQ. 0.0) GO TO 613                                 OUTE 178
C     ACCELERATION FOR EXTRANEOUS SOURCE PROBLEMS               OUTE 179
611   TEMP1 = (1.0 - EV*T11/E01)/(1.0 - EV)                     OUTE 180
      IF (T11/E01 - .01) 620,620,612                            OUTE 181
612   IF (T11/E01 - 1./(EV + .0001)) 613,613,620                OUTE 182
613   DO 614 I = 1,IMJM                                         OUTE 183
614   FO(I) = TEMP1 * FO(I)                                     OUTE 184
```

```
      DO 616 IIG = 1,IGM                                          OUTE 185
      E0(IIG) = TEMP1*E0(IIG)                                     OUTE 186
616   E1(IIG) = TEMP1*E1(IIG)                                     OUTE 187
620   E1(IGP) = 0.0                                               OUTE 188
      E0(IGP) = 0.0                                               OUTE 189
      DO 640  IIG = 1,IGM                                         OUTE 190
      E0(IGP) = E0(IGP) + E0(IIG)                                 OUTE 191
640   E1(IGP) = E1(IGP) + E1(IIG)                                 OUTE 192
      RETURN                                                      OUTE 193
      END                                                         OUTE 194
      SUBROUTINE PRT (JIM,JJM, N2, Z4, NOUT)                      PRT    2
      DIMENSION N2(JIM,JJM), Z4(1)                                PRT    3
      REAL N2                                                     PRT    4
C     DATA XRR/6HXRR   /,YZT/6HYZT   /                            PRT    5
      CHARACTER*6 XRR, YZT                                        HVX
      DATA XRR/'XRR   '/,YZT/'YZT   '/                            HVX
      DATA LINES/0/                                               PRT    6
      IM = JIM                                                    PRT    7
      JM = JJM                                                    PRT    8
      DO 50 I=1,IM,6
      I1=I                                                        PRT   10
      I2=I+5
      IF(I2-IM) 20, 20, 10                                        PRT   12
10    I2=IM                                                       PRT   13
   20 WRITE(NOUT,30) YZT,(XRR,JJ,JJ=I1,I2)                        PRT   14
   30 FORMAT(1X,A3,6(3X,A3,I4,1X))
      DO 50  JJ=1,JM                                              PRT   16
      J=JJ                                                        PRT   17
40    FORMAT (I4, 6E11.4, F9.3)
      IF(J.EQ.1)GO TO 45                                          PRT   19
      DO 42 K=I1,I2                                               PRT   20
      IF(N2(K,J).NE.N2(K,J-1)) GO TO 43                           PRT   21
   42 CONTINUE                                                    PRT   22
      LINES=LINES+1                                               PRT   24
      IF(J-JM) 50,43,43                                           PRT   25
   43 IF(LINES.EQ.0)GO TO 45                                      PRT   26
      WRITE(NOUT,44) LINES                                        PRT   27
   44 FORMAT('      NEXT',I5,' LINES SAME AS PRECEDING LINE')
```

```
      LINES=0                                                           PRT   29
      IF(J-JM) 45,50,45                                                 PRT   30
   45 WRITE(NOUT,40)J,(N2(K,J),K=I1,I2), Z4(J)
   50 CONTINUE                                                          PRT   34
      RETURN                                                            PRT   35
      END                                                               PRT   36
      SUBROUTINE AVERAG(PHIB,AXS,FXS,MATN,MASS,ATW,VOL,CO,N2,MO,VO,     AVER   2
C     1         HOLN, JML, JTL, NBR,T6,T8,NTWON)                         HRA2
     1         HOLN, JML, JTL, NBR,ACT,JNACT,IACPOS)                    HRA3
      DIMENSION PHIB(1), AXS(JML,1), FXS(JML,1), MATN(1), MASS(JML,1),  AVER   4
     1           ATW(1), VOL(1), CO(JTL,1), N2(1), MO(1), VO(1),
     2           HOLN(JML,1), NBR(1)
C      DIMENSION T6(JML,1),T8(JML,1)                                     HRA2
      DIMENSION ACT(JML,1,JML,1),IACPOS(JNACT),T1(20,20),ACTIV(20,20)   HRA3
      INCLUDE 'ABC.FOR'
C     THIS SUBROUTINE CALCULATES ZONE AVERAGED FLUXES, FISSION CROSS    AVER   8
C     SECTIONS, AND ABSORPTION CROSS SECTIONS.                          AVER   9
      RL = 0.0                                                          AVER  10
      RC = 0.0                                                          AVER  11
      DO 10  KZ=1,IZM                                                   AVER  12
      PHIB(KZ) = 0.0                                                    AVER  13
      DO 10  KN =1,NCON                                                 AVER  14
      AXS(KN,KZ) = 0.0                                                  AVER  15
      FXS(KN,KZ) = 0.0                                                  AVER  16
C      T6(KN,KZ) = 0.0                                                   HRA2
C      TTOTAL = 0.0                                                      HRA2
C      T8(KN,KZ) = 0.0                                                   HRA2
      DO 1000 I=1,NACT                                                  HRA3
      K=IACPOS(I)                                                       HRA3
      LN = MATN(KN)                                                     HRAH
      ACT(LN,K,KN,KZ) = 0.0                                             HRA3
1000  CONTINUE                                                          HRA3
      LN = MATN(KN)                                                     AVER  17
      IF (MASS(LN,KZ) .EQ. 0) GO TO 10
      MASS(LN,KZ) = (MASS(LN,KZ)*.6023)/(ATW(LN)*VOL(KZ))               AVER  18
10    CONTINUE
      DO 100  IIG=1,IGM                                                 AVER  19
      READ(NCR1)  ((CO(II,J), II=1,ITL),J=1,MT)                         AVER  20
```

```
      READ(NFLUX1)  (N2(I), I=1,IMJM)                              AVER  21
      DO 11 J=1,NCR                                                HRA2
C      CALL REAG2('    RO',ACO(1,MT),3)                            HRA2
11    CONTINUE                                                     HRA2
      DO 100  I=1,IMJM                                             AVER  22
      KZ = MO(I)                                                   AVER  23
      PHIB(KZ) = PHIB(KZ) + N2(I)*VO(I)                            AVER  24
      DO 100  KN=1,NCON                                            AVER  25
      LN = MATN(KN)                                                AVER  26
C   CO(5,1 AND 2) ARE ACTIVITIES FOR TPROD                         HRA2
C      IF(LN.EQ.1)THEN                                              HRA2
C       T6(KN,KZ) = T6(KN,KZ) + CO(5,LN)*N2(I)*VO(I)                HRA2
C      ENDIF                                                        HRA2
C      IF(LN.EQ.2)THEN                                              HRA2
C       T8(KN,KZ) = T8(KN,KZ) + CO(5,LN)*N2(I)*VO(I)                HRA2
C      ENDIF                                                        HRA2
      DO 2000 K=1,NACT                                             HRA3
       J=IACPOS(K)                                                 HRA3
       ACT(LN,J,KN,KZ) = ACT(LN,J,KN,KZ) + CO(J,LN)*N2(I)*VO(I)    HRA3
C      WRITE(6,*)J,ACT(J,KN,KZ),IACPOS(K)
2000  CONTINUE                                                     HRA3
      AXS(KN,KZ) = AXS(KN,KZ) + CO(IHT-2,LN)*N2(I)*VO(I)           HRA2  27
100   FXS(KN,KZ) = FXS(KN,KZ) + CO(IHT-3,LN)*N2(I)*VO(I)           HRA2  28
C      DO 200  KZ=1,IZM                                            HRA2  29
      DO 209  KZ=1,IZM                                             HRA2  29
C      TRIT6=0.0                                                    HRA2
C      TRIT7=0.0                                                    HRA2
      DO 2500 KI=1,NACT                                            HRA3
      DO 2480 LL=1,NCON                                            HRAH
      MM=MATN(LL)                                                  HRAH
      ACTIV(MM,KI)=0.0                                             HRA3
2480  CONTINUE                                                     HRAH
2500  CONTINUE                                                     HRA3
      TEMP3 = PHIB(KZ)                                             AVER  30
      IF (PHIB (KZ) .EQ. 0) GO TO 105
      PHIB(KZ) = PHIB(KZ)/(VOL(KZ)*1000.)                          AVER  31
105   CONTINUE
      WRITE(NOUT,110)  KZ, PHIB(KZ), VOL(KZ)                       AVER  32
```

```
110   FORMAT(1H1,2X,9H Z O N E ,I3,5X,7H FLUX =,1PE10.4,5X,9H VOLUME =,
     1 1PE10.4,7H LITERS/)                                         AVER  34
      WRITE(NOUT,120)                                              AVER  35
120   FORMAT (1X,'BURNABLE MAT.    NAME      ATOM  '
     1,'   FISSION   ABSORPTION   SIGMA    SIGMA '/
     3 1X,'ISOTOPE  NO. ', 15X,'DENSITY',4X,'RATE',8X,'RATE',4X,
     3 'FISSION',2X,'ABSORPTION'/)
      DO 200  KN=1,NCON                                            AVER  41
      LN = MATN(KN)                                                AVER  42
      TEMP1 = AXS(KN,KZ)*MASS(LN,KZ)                               AVER  43
      TEMP2 = FXS(KN,KZ)*MASS(LN,KZ)                               AVER  44
C      T1 = T6(KN,KZ)*MASS(LN,KZ)                                   HRA2
C      T2 = T8(KN,KZ)*MASS(LN,KZ)                                   HRA2
C      T6(KN,KZ) = T6(KN,KZ)/TEMP3                                  HRA2
C      T8(KN,KZ) = T8(KN,KZ)/TEMP3                                  HRA2
      DO 3000 K=1,NACT                                             HRA3
       J=IACPOS(K)                                                 HRA3
      T1(LN,K) = ACT(LN,J,KN,KZ)*MASS(LN,KZ)                       HRA3
      ACT(LN,J,KN,KZ) = ACT(LN,J,KN,KZ)/TEMP3                      HRA3
3000  CONTINUE                                                     HRA3
      AXS(KN,KZ) = AXS(KN,KZ)/TEMP3                                AVER  45
      FXS(KN,KZ) = FXS(KN,KZ)/TEMP3                                AVER  46
130   FORMAT (1X, I3, 5X, I3, 4X, 2A4, 5E11.4)                     HRA2
      WRITE(NOUT,130)  KN, LN, (HOLN(LN,K),K=1,2), MASS(LN,KZ), TEMP2,
     1 TEMP1,FXS(KN,KZ), AXS(KN,KZ)
C      TRIT6=TRIT6+T1                                               HRA2
C      TRIT7=TRIT7+T2                                               HRA2
C      TTOTAL=TTOTAL+T1+T2                                          HRA2
      DO 4000 KKK=1,NACT                                           HRA3
      ACTIV(LN,KKK)=ACTIV(LN,KKK)+T1(LN,KKK)                       HRA3
4000  CONTINUE
      ITEMP = NBR(KN)                                              AVER  50
      IF(ITEMP - 1)  200, 140, 160                                 AVER  51
140   RC = RC + TEMP1 - TEMP2                                      AVER  52
      GO TO 200                                                    AVER  53
160   RL = RL + TEMP1                                              AVER  54
200   CONTINUE                                                     AVER  55
C      WRITE(6,134)TRIT6,TRIT7                                      HRA2
```

```
C134   FORMAT(' T6=',E11.4,3X,'T7=',E11.4)                            HRA2
       DO 5500 JJ=1,NCON                                              HRAH
        LN=MATN(JJ)                                                   HRAH
       DO 5000 KKK=1,NACT                                             HRA3
       WRITE(6,1340)KKK,LN,ACTIV(LN,KKK)                              HRA3
1340   FORMAT(' ACTIVITY',I2,' FOR MAT. NO.',I2,' =',E11.4)           HRA3
5000   CONTINUE                                                       HRA3
5500   CONTINUE                                                       HRAH
       WRITE(6,*)
       WRITE(6,*)
       WRITE(6,*)
209    CONTINUE                                                       HRA2
       IF(RL.EQ.0.0)THEN                                              HRA2
          TEMP=0.0                                                    HRA2
          GO TO 340                                                   HRA2
       ENDIF                                                          HRA2
       TEMP = RC/RL                                                   AVER  56
C       WRITE(NOUT,350)  TEMP                                          AVER  5
7
340    WRITE(NOUT,350)  TEMP                                          HRA2
350    FORMAT(1H ///' BREEDING RATIO        =',F7.4)
C       WRITE(6,401)  TTOTAL                                           HRA2
C401    FORMAT(' TTOTAL=',E11.4)                                       HRA2
       REWIND NCR1                                                    AVER  59
       REWIND NFLUX1                                                  AVER  60
       RETURN                                                         AVER  61
       END                                                            AVER  62
       SUBROUTINE CLEAR (X,Y,N)                                       CLER   2
       DIMENSION Y(1)                                                 CLER   3
       DO 1 I=1,N                                                     CLER   4
1      Y(I)=X                                                         CLER   5
       RETURN                                                         CLER   6
       END                                                            CLER   7
       SUBROUTINE CNNP (F2,K6)                                        CNNP   2
       DIMENSION F2(1), K6(1)                                         CNNP   3
       INCLUDE 'ABC.FOR'

       IF (MAXT) 25,25,10
```

```
10      JUMP=2                                                          CNNP   5
        CALL TCHEK(GLH,JUMP)                                            CNNP   6
        GO TO (15,25),JUMP                                              CNNP   #
   15 WRITE(NOUT,20)                                                    CNNP   8
20      FORMAT(53H1 * *  RUNNING TIME EXCEEDED--FORCED CONVERGENCE  * *//)CNNP 9
        GO TO 90                                                        CNNP  10
25      CONTINUE                                                        CNNP  11
30      E01=1.0-ALA                                                     CNNP  12
        IF(ABS (E01)-10.0*EPS)  40,  40,  45                            CNNP  13
40      ORF = ORFP                                                      CNNP  14
45      CONTINUE                                                        CNNP  15
        E02=ABS(E01)                                                    CNNP  16
50      IF(E1(IGP)) 55, 130, 55                                         CNNP  17
55      IF (E02 - EPS)  60, 60, 70                                      CNNP  18
60      CVT=1                                                           CNNP  19
70      CALL CLEAR (0.0, F2, IMJM)                                      CNNP  23
        GO TO 105                                                       CNNP  24
80      EV=EV+POD*EQ*E01                                                CNNP  25
        GO TO 170                                                       CNNP  26
C       FINAL PRINT                                                     CNNP  27
90      NGOTO=1                                                         CNNP  28
        IF (I04 - 1)  135,  95,  80                                     CNNP  29
95      EV=0.0                                                          CNNP  30
        DO 100   I=1,IGM                                                CNNP  31
100     EV=EV+K6(I)                                                     CNNP  32
        EV=SK7/EV                                                       CNNP  33
        GO TO 135                                                       CNNP  34
105     IF(CVT-1) 110, 90, 110                                          CNNP  35
110     IF(I04-1) 115, 120, 140                                         CNNP  36
C       MONITOR PRINT                                                   CNNP  37
115     NGOTO=2                                                         CNNP  38
        GO TO 135                                                       CNNP  39
120     EV=0.                                                           CNNP  40
        DO 125 I=1,IGM                                                  CNNP  41
125     EV=EV+K6(I)                                                     CNNP  42
        EV=SK7/EV                                                       CNNP  43
        GO TO 115                                                       CNNP  44
  130 IF(I04.EQ.0)GO TO 55                                              CNNP  45
```

```
      CALL ERRO2(6H**CNNP,130,1)                                    CNNP  46
135   RETURN                                                        CNNP  47
140   CONTINUE                                                      CNNP  48
C                                                                   CNNP  49
C     CALCULATE NEW PARAMETERS FOR SEARCH CALCULATIONS              CNNP  50
145   E03=ABS (ALA-LAR)                                             CNNP  51
      IF (LAPP)  270, 150, 270                                      CNNP  52
150   IF (LAP)  230, 155, 230                                       CNNP  53
155   IF (EQ) 200, 160, 200                                         CNNP  54
160   IF (E03-EPSA) 175, 175, 165                                   CNNP  55
C     MONITOR PRINT.                                                CNNP  56
165   NGOTO=2                                                       CNNP  57
      RETURN                                                        CNNP  58
C     FINAL PRINT EXIT.                                             CNNP  59
170   NGOTO=1                                                       CNNP  60
      RETURN                                                        CNNP  61
175   LAP=ALA                                                       CNNP  62
      EVP=EV                                                        CNNP  63
      IF (E01) 185,185,180                                          CNNP  64
180   EV=EV-EVM                                                     CNNP  65
      GO TO 190                                                     CNNP  66
185   EV=EV+EVM                                                     CNNP  67
190   IF (I04-2)   195,  165,  195                                  CNNP  68
C     MIX X-SECS.                                                   CNNP  69
195   NGOTO=3                                                       CNNP  70
      RETURN                                                        CNNP  71
200   IF (CVT) 170, 205,170                                         CNNP  72
205   EV=EV+POD*EQ*E01                                              CNNP  73
210   IF ((LAPP-1.0)/(LAP-1.0))  215, 190, 190                      CNNP  74
215   TEMP1=AMIN1(EVP,EVPP)                                         CNNP  75
      IF (EV-TEMP1)  220, 225, 225                                  CNNP  76
220   EV=(EVPP+EVP)/2.                                              CNNP  77
      GO TO 190                                                     CNNP  78
225   TEMP1=AMAX1(EVP,EVPP)                                         CNNP  79
      IF (EV-TEMP1)  190, 220, 220                                  CNNP  80
230   IF (E03-EPSA)  235, 235, 165                                  CNNP  81
235   EQ=(EVP-EV)/(LAP-ALA)                                         CNNP  82
240   IF (CNT)  260, 245, 260                                       CNNP  83
```

```
245   IF (E02-LAL)  265, 265, 250                             CNNP  84
250   IF (E02-LAH)  260, 260, 255                             CNNP  85
255   E01=SIGN (LAH,E01)                                      CNNP  86
260   LAPP=LAP                                                CNNP  87
      LAP=ALA                                                 CNNP  88
      EVPP=EVP                                                CNNP  89
      EVP=EV                                                  CNNP  90
      GO TO 205                                               CNNP  91
265   CNT=1                                                   CNNP  92
      LAP=0.0                                                 CNNP  93
      LAPP=0.0                                                CNNP  94
      GO TO 205                                               CNNP  95
270   IF (E03-EPSA) 275, 275, 165                             CNNP  96
C     CALCULATE QUADRATIC COEFFICIENTS.                       CNNP  97
275   TEMP1=EVP-EV                                            CNNP  98
      TEMP2=EVPP-EV                                           CNNP  99
      TEMP3=EVPP-EVP                                          CNNP 100
      TEMP4=TEMP1*(EVP+EV)                                    CNNP 101
      TEMP5=-TEMP2*(EV+EVPP)                                  CNNP 102
      TEMP6=TEMP3*(EVPP+EVP)                                  CNNP 103
      DENOM=TEMP3*TEMP2*TEMP1                                 CNNP 104
      EQA=((LAPP-1.0)*TEMP1*EVP*EV-(LAP-1.0)*TEMP2            CNNP 105
     1*EV*EVPP+(ALA-1.0)*TEMP3*EVPP*EVP)/DENOM                CNNP 106
      EQB=-(LAPP*TEMP4+LAP*TEMP5+ALA*TEMP6)/DENOM             CNNP 107
      EQC=(LAPP*TEMP1-LAP*TEMP2+ALA*TEMP3)/DENOM              CNNP 108
      DISCR=EQB*EQB-4.0*EQA*EQC                               CNNP 109
      IF (DISCR)  235, 280, 280                               CNNP 110
280   IF (E02-LAL)  265, 265, 285                             CNNP 111
285   TEMP1=EQC+EQC                                           CNNP 112
      TEMP=SQRT (DISCR)                                       CNNP 113
      EQ=1.0/(EQB+EV*TEMP1)                                   CNNP 114
      LAPP=LAP                                                CNNP 115
      LAP=ALA                                                 CNNP 116
      EVPP=EVP                                                CNNP 117
      EVP=EV                                                  CNNP 118
      EV1=(TEMP-EQB)/TEMP1                                    CNNP 119
      EV2=-(TEMP+EQB)/TEMP1                                   CNNP 120
      EVA=ABS (EV-EV1)                                        CNNP 121
```

```
      EVB=ABS (EV-EV2)                                                 CNNP 122
      IF (EVA-EVB)  290, 290, 295                                      CNNP 123
290   EV=EV1                                                           CNNP 124
      GO TO 210                                                        CNNP 125
295   EV=EV2                                                           CNNP 126
      GO TO 210                                                        CNNP 127
      END                                                              CNNP 128
      SUBROUTINE ERRO2( HOL,JSUBR,I)                                   ERR2   2
      COMMON  NSORCE,  NINP,  NOUT,  NCR1,  NFLUX1,  NSCRAT            ERR2   3
      CHARACTER*6 HOL                                                  HVX
      DATA NERR/0/                                                     ERR2   4
      NERR=NERR+1                                                      ERR2   7
      WRITE (NOUT,1)          HOL,JSUBR                                ERR2   8
1     FORMAT(2H */9H ERROR IN,A6,3H AT,I6/2H */2H *)                   ERR2   9
      IF(NERR.EQ.100)GO TO 3
      GO TO (3,4),I                                                    ERR2  1#
3     STOP                                                             ERR2  12
4     RETURN                                                           ERR2  13
      END                                                              ERR2  14
      SUBROUTINE FISCAL  (NO, FO, VO, CO, K6, MO, M2, JTL,JMT)         FISC   2
       INCLUDE 'ABC.FOR'

      DIMENSION  NO(1), FO(1), VO(1), CO(JTL,1),K6(1), MO(1), M2(1)
      LAR = ALA                                                        FISC   5
C     FISSION SUMS                                                     FISC   6
      IF(B07)  90,90,10                                                FISC   7
10    IF(A02) 20, 40, 20                                               FISC   8
20    DO 30  IIG=1,IGM                                                 FISC   9
      READ (NCR1)     ((CO(I,J),I=1,ITL),J=1,MT)                       FISC  10
      E1(IIG)=0.                                                       FISC  11
      DO 30  I=1,IMJM                                                  FISC  12
      ITEMP=MO(I)                                                      FISC  13
      ITEMP=M2(ITEMP)                                                  FISC  14
30    E1(IIG)=E1(IIG)+CO(IHT-1,ITEMP)*FO(I)*VO(I)                      FISC  15
      REWIND NCR1                                                      FISC  16
      GO TO 70                                                         FISC  17
40    E01=0.                                                           FISC  18
      DO 50  I=1,IMJM                                                  FISC  19
```

```
50     E01=E01+V0(I)*F0(I)                                    FISC  20
       DO 60  IIG=1,IGM                                       FISC  21
60     E1(IIG)=K6(IIG)*E01                                    FISC  22
70     E1(IGP)=0.                                             FISC  23
       E0(IGP)=0.                                             FISC  24
       DO 80  IIG=1,IGM                                       FISC  25
       E0(IGP)=E0(IGP)+E0(IIG)                                FISC  26
80     E1(IGP)=E1(IGP)+E1(IIG)                                FISC  27
       IF(B07)  140, 90, 140                                  FISC  28
90     IF(T11.EQ. 0.0)GO TO 95
       ALA=E1(IGP)/T11
       TEMP=1.0/ALA                                           FISC  31
95     IF(I04-1) 230,100,140
100    DO 110  IIG=1,IGM                                      FISC  32
       E1(IIG)=E1(IIG)*TEMP                                   FISC  33
110    K6(IIG)=K6(IIG)*TEMP                                   FISC  34
       E1(IGP)=E1(IGP)*TEMP                                   FISC  35
       IF(A02) 120, 140, 120                                  FISC  36
120    DO 130  I=1,IMJM                                       FISC  37
130    F0(I)=F0(I)*TEMP                                       FISC  38
140    CONTINUE                                               FISC  39
C                                                             FISC  40
C      NORMALIZATION                                          FISC  41
       B07=0                                                  FISC  42
150    IF(S01)  160, 230, 170                                 FISC  43
160    E01 = ABS(S01)/(E0(IGP)*TSD)                           FISC  44
       GO TO 180                                              FISC  45
170    E01=S01/E1(IGP)                                        FISC  46
180    DO 190 IIG=1,IGP                                       FISC  47
190    E1(IIG)=E01*E1(IIG)                                    FISC  48
       DO 200 I=1,IMJM                                        FISC  49
200    F0(I)=E01*F0(I)                                        FISC  50
230    RETURN                                                 FISC  51
       END                                                    FISC  52
       SUBROUTINE GRAM(MASS, VOL,  ATW, HOLN,JIM,JJM, NO, M2, V0,   GRAM   2
     1                I0, I1, I2,JML, I3)                     GRAM   3
        INCLUDE 'ABC.FOR'
```

```
      DIMENSION MASS(JML,1), VOL(1), ATW(1), HOLN(JML,1), MO(JIM,JJM),
     1          M2(1), VO(JIM,JJM), IO(1), I1(1), I2(1), I3(1)
C     THIS SUBROUTINE CALCULATES THE MASS OF THE VARIOUS MATERIALS
      WRITE(NOUT,10) (ID(I), I=1,20)
10    FORMAT(1H1,20A4///)
      WRITE(NOUT, 20)
20    FORMAT(45H MATERIAL INVENTORY (KILOGRAMS) FOR EACH ZONE /  )
      CALL CLEAR(0.0,VOL,IZM)
      ITEMP = ML*IZM
      CALL CLEAR(0.0,MASS,ITEMP)
      DO 30  J = 1, JM
      DO 30  I = 1, IM
      K = MO(I,J)
30    VOL(K) = VOL(K) + VO(I, J)*.001
      DO 39  M=1,M01
      I3(M) = I2(M)
      IF(IO(M) - I1(M))  39,35,39
35    IF(I2(M))  39,36,39
36    DO 38  MM=1,M
      IF(IO(M) - IO(MM))  38,37,38
37    I3(MM) = I2(MM)*EV
38    CONTINUE
39    CONTINUE
      DO 190 N =1, IZM
      NN = M2(N)
      DO 190 M = 1,M01
      IF(IO(M) - NN)   190, 40, 190
40    L = I1(M)
      IF(L - ML)  170, 170, 50
50    NNAA = L
      IF(L - IO(M))  130,190, 130
130   DO 160 MAA = 1, M01
      IF(IO(MAA) - NNAA)  160, 140, 160
140   L = I1(MAA)
      IF(L)  160, 160, 150
150   E01 = I3(MAA)*I3(M)
      MASS(L,N) = ((E01*ATW(L)*VOL(N))/.6023) + MASS(L,N)
160   CONTINUE
```

```
      GO TO 190                                                    GRAM  43
170   IF(L)  190, 190, 180                                         GRAM  44
180   E01 = I3(M)                                                  GRAM  45
      MASS(L,N) = ((E01*ATW(L)*VOL(N))/.6023) + MASS(L,N)          GRAM  46
190   CONTINUE                                                     GRAM  47
      DO 260 L = 1, IZM, 5                                         GRAM  49
      LL = L + 4                                                   GRAM  50
      IF(LL - IZM) 210, 210, 200                                   GRAM  51
200   LL = IZM                                                     GRAM  52
210   WRITE(NOUT,220)  ((K), K=L, LL)
220   FORMAT (//' MATERIAL     ATOMIC WT.   ',2X,5(' ZONE',I3,3X)/)
      DO 240 K = 1, ML                                             GRAM  57
      DO 233 I=L,LL                                                UPD1   3
      IF(MASS(K,I) .NE. 0.) GO TO 238                              UPD1   4
  233 CONTINUE                                                     UPD1   5
      GO TO 240                                                    UPD1   6
  238 WRITE(NOUT,250) K,(HOLN(K,N),N=1,2),ATW(K), (MASS(K,I), I=L,LL)

  240 CONTINUE                                                     UPD1   8
250   FORMAT (1X,I3,1X,2A4, F12.2, 1X, 5E11.3)
      IF(LL - IZM)  260, 270, 270                                  GRAM  60
260   CONTINUE                                                     GRAM  61
C   COMPUTE TOTAL MASSES                                           UPD1   9
270   WRITE (NOUT,275)
275   FORMAT (//' MATERIAL     ATOMIC WT.     TOTAL'/)
      DO 310 K=1,ML                                                UPD1  16
      TEMP=0.0
      DO 280 L=1,IZM
  280 TEMP=TEMP+MASS(K,L)
310   WRITE(NOUT,250) K,(HOLN(K,N),N=1,2),ATW(K),TEMP

      WRITE (NOUT,350)
350   FORMAT (///,' ZONE NUMBER    VOLUME (LITERS)'/)
      DO 400 L=1,IZM
      WRITE (NOUT,360) L,VOL(L)
360   FORMAT (6X,I4,6X,1PE12.3)
400   CONTINUE
      RETURN                                                       UPD1  18
```

```
      END                                                               GRAM  63
      SUBROUTINE IFLUXN (N2, CO, VO, CXS, MO, M2, JTL,JIM,JJM, CXR, CXT HRA2   2
     1  ,XR,XD)                                                         HRA2
      INCLUDE 'ABC.FOR'

      DIMENSION  N2(1), CO(JTL,1), VO(1),CXS(JIM,JJM,3),MO(1), M2(1),   IFLU   4
     1          CXR(1), CXT(1)                                          IFLU   5
      DIMENSION XD(50),XR(50)                                           HRA2
C     THIS SUBROUTINE NORMALIZES FLUXES BEFORE EACH INNER ITERATION     IFLU   6
C     ABSORPTION AND OUT-SCATTER                                        IFLU   7
      E3(IGV) = 0.0                                                     IFLU   8
      E4(IGV) = 0.0                                                     IFLU   9
      DO 10  I=1, IMJM                                                  IFLU  10
      TEMP = VO(I)*N2(I)                                                IFLU  11
      ITEMP = MO(I)                                                     IFLU  12
      ITEMP = M2(ITEMP)                                                 IFLU  13
      E3(IGV) = E3(IGV) + (XD(ITEMP))*TEMP                              HRA2  14
10    E4(IGV) = E4(IGV) + CO(IHT-2,ITEMP)*TEMP                          HRA2  15
C     LEFT LEAKAGE                                                      IFLU  16
      IF(B01)  20, 20, 40                                               IFLU  17
20    E5(IGV) = 0.0                                                     IFLU  18
      DO 30  KJ = 1, JM                                                 IFLU  19
      I = (KJ - 1)*IM + 1                                               IFLU  20
30    E5(IGV) = E5(IGV) + CXS(1,KJ,1)*N2(I)                             IFLU  21
      GO TO 50                                                          IFLU  22
40    E5(IGV) = .0                                                      IFLU  23
C     RIGHT LEAKAGE                                                     IFLU  24
50    IF(B02)  60, 60, 80                                               IFLU  25
60    E6(IGV) = 0.0                                                     IFLU  26
      DO 70  KJ = 1, JM                                                 IFLU  27
      I = KJ*IM                                                         IFLU  28
70    E6(IGV) = E6(IGV) + CXR(KJ)*N2(I)                                 IFLU  29
      GO TO 90                                                          IFLU  30
80    E6(IGV) = 0.0                                                     IFLU  31
C     TOP LEAKAGE                                                       IFLU  32
90    IF(B03-1)  120, 140, 100                                          IFLU  33
100   E7(IGV) =.0                                                       IFLU  34
      DO 110   KI = 1,IM                                                IFLU  35
```

```
      I = IMJM - IM + KI                                          IFLU  36
110   E7(IGV) = E7(IGV) + CXS(KI,1,2)*(N2(I) - N2(KI))            IFLU  37
      E8(IGV) = - E7(IGV)                                         IFLU  38
      GO TO 190                                                   IFLU  39
120   E7(IGV) = 0.0                                               IFLU  40
      DO 130  KI = 1, IM                                          IFLU  41
      I = IMJM - IM + KI                                          IFLU  42
130   E7(IGV) = E7(IGV) + CXT(KI)*N2(I)                           IFLU  43
      GO TO 150                                                   IFLU  44
140   E7(IGV) = 0.0                                               IFLU  45
C     BOTTOM LEAKAGE                                              IFLU  46
150   IF(B04)  160, 160, 180                                      IFLU  47
160   E8(IGV) = 0.0                                               IFLU  48
      DO 170 KI = 1, IM                                           IFLU  49
170   E8(IGV) = E8(IGV) + CXS(KI,1,2)*N2(KI)                      IFLU  50
      GO TO 190                                                   IFLU  51
180   E8(IGV) = 0.0                                               IFLU  52
190   E9(IGV) = E5(IGV) + E6(IGV) + E7(IGV) + E8(IGV)             IFLU  53
      TEMP = (E1(IGV) + E2(IGV))/(E3(IGV) + E4(IGV) + E9(IGV))    IFLU  54
      DO 200 I = 1, IMJM                                          IFLU  55
200   N2(I) = TEMP*N2(I)                                          IFLU  56
      E3(IGV) = TEMP*E3(IGV)                                      IFLU  57
      E4(IGV) = TEMP*E4(IGV)                                      IFLU  58
      E5(IGV) = TEMP*E5(IGV)                                      IFLU  59
      E6(IGV) = TEMP*E6(IGV)                                      IFLU  60
      E7(IGV) = TEMP*E7(IGV)                                      IFLU  61
      E8(IGV) = TEMP*E8(IGV)                                      IFLU  62
      E9(IGV) = TEMP*E9(IGV)                                      IFLU  63
      RETURN                                                      IFLU  64
      END                                                         IFLU  65
      SUBROUTINE INIT (K6, K7, I0, I1, I2, M0, M2, N0, R0, R1, R2,  INIT   2
     1                 R3, R4, R5, Z0, Z1, Z2, Z3, Z4, Z5, A0, A1,  INIT   3
     2                F0,C0,V0,JTL,JIM,V7,JJM,JMT,JML,GAM,HOLN)
       INCLUDE 'ABC.FOR'

      DIMENSION  K6(1), K7(1), I0(1), I1(1), I2(1), R0(1), R1(1),   INIT   6
     1           R2(1), R3(1), R4(1), R5(1), Z0(1), Z1(1), Z2(1),   INIT   7
     2           Z3(1), Z4(1), Z5(1), A0(1), A1(1), C0(JTL,JMT),    INIT   8
```

```
    3          VO(JIM,JJM), MO(1), M2(1), NO(1), FO(1), V7(1)         INIT   9
    4         ,GAM(1), HOLN(JML,1)
      DIMENSION XD(50),XR(50)                                         HRA2
      IF (PO2) 20, 10, 20                                             INIT  11
10    WRITE(NOUT,15)  DAY                                             INIT  12
15    FORMAT(1H1,24X,' T I M E  =',F8.3,8H D A Y S///)
20    CONTINUE                                                        INIT  14
C     ADJOINT REVERSALS                                               INIT  15
      IF(AO2) 25, 45, 25                                              INIT  16
25    IF(PO2) 45, 30, 45                                              INIT  17
30    IF(NCON)  45, 35, 45                                            INIT  18
35    IIG=1                                                           INIT  19
      IGBAR=IGM                                                       INIT  20
40    TEMP=K7(IIG)                                                    INIT  21
      K7(IIG)=K7(IGBAR)                                               INIT  22
      K7(IGBAR)=TEMP                                                  INIT  23
      TEMP=V7(IIG)                                                    INIT  24
      V7(IIG)=V7(IGBAR)                                               INIT  25
      V7(IGBAR)=TEMP                                                  INIT  26
      IIG=IIG+1                                                       INIT  27
      IGBAR=IGBAR-1                                                   INIT  28
      IF(IIG-IGBAR) 40, 45, 45                                        INIT  29
45    CONTINUE                                                        INIT  30
C     MIX CROSS-SECTIONS                                              INIT  32
      BO7=1                                                           INIT  33
      IF(PO2) 50, 55, 50                                              INIT  34
50    GO TO (245,245,85,245,185),  IO4                                INIT  3#
55    IF(MO1)  70, 70, 60                                             INIT  36
   60 WRITE(NOUT,61)                                                  UPD1  22
   61 FORMAT(1H0,4X,16H MIXTURE NUMBER  ,18H   MIX COMMAND           , UPD1  23
     124H MATERIAL ATOMIC DENSITY     /)                              UPD1  24
      DO 67 J=1,MO1                                                   UPD1  25
      WORD = 0
      WORD2 = WORD
      I=I1(J)                                                         UPD1  27
      IF(I .GT. 0 .AND.I .LE. NL) WORD=HOLN(I,1)                       HVX
      IF (I .GT. 0 .AND. I .LE. ML) WORD2=HOLN(I,2)
      WRITE(NOUT,63) J,IO(J),I1(J),I2(J),WORD,WORD2
```

```
   63 FORMAT(I5,I9,I15,E28.8,12X,2A4)
      IF(J .EQ. M01 .OR. IO(J).EQ. IO(J+1)) GO TO 67                    UPD1  31
66    FORMAT (1X, 15(5H-----))
      WRITE(NOUT,66)                                                    UPD1  33
   67 CONTINUE                                                          UPD1  34
70    IF(NPRT-1)  85, 85, 75
75    WRITE (NOUT,80 )                                                  INIT  41
80    FORMAT(/19H1CROSS-SECTION EDIT)                                   INIT  42
85    REWIND NCR1                                                       INIT  43
      DO 180  IIG=1,IGM                                                 INIT  44
      READ (NCR1)    ((CO(I,J),I=1,ITL),J=1,MT)                         INIT  45
C      READ(11) (XR(J),J=1,MT)                                           HRA2
      READ(12) (XD(J),J=1,MT)                                           HRA2
      IF(M01) 90, 145, 90                                               INIT  46
90    DO 140  M=1,M01                                                   INIT  47
      IF(IO(M)-MT) 100, 100, 95                                         INIT  48
95    CALL ERRO2('**INIT',95,1)                                         HVX
100   IF(I1(M)-MT) 105, 105, 95                                         INIT  50
105   N=IO(M)                                                           INIT  51
      L=I1(M)                                                           INIT  52
      E01=I2(M)                                                         INIT  53
      IF(L) 125, 125, 110                                               INIT  54
110   IF(E01) 125, 115, 125                                             INIT  55
115   IF (N-L) 125, 120, 125                                            INIT  56
120   E01 = EV                                                          INIT  57
      L = 0                                                             INIT  58
125   DO 140  I=1,ITL                                                   INIT  59
      IF (L) 130, 135, 130                                              INIT  60
130   CO(I,N)=CO(I,N)+CO(I,L)*E01                                       INIT  61
      GO TO 140                                                         INIT  62
135   CO(I,N)=CO(I,N)*E01                                               INIT  63
140   CONTINUE                                                          INIT  64
      IF(M01) 900, 145, 900                                             HRA2  46
900    DO 1140  M=1,M01                                                 HRA2  47
      IF(IO(M)-MT) 1000, 1000, 950                                      HRA2  48
950    CALL ERRO2('**INIT',950,1)                                       HRA2 HVX
1000   IF(I1(M)-MT) 1050, 1050, 950                                     HRA2  50
1050   N=IO(M)                                                          HRA2  51
```

```
      L=I1(M)                                                   HRA2  52
      E01=I2(M)                                                 HRA2  53
      IF(L) 1125, 1125, 1110                                    HRA2  54
1110   IF(E01) 1125, 1115, 1125                                 HRA2  55
1115   IF (N-L) 1125, 1120, 1125                                HRA2  56
1120   E01 = EV                                                 HRA2  57
      L = 0                                                     HRA2  58
1125  CONTINUE                                                  HRA2  59
      IF (L) 1130, 1135, 1130                                   HRA2  60
1130   XD(N)=XD(N)+XD(L)*E01                                    HRA2  61
      GO TO 1140                                                HRA2  62
1135   XD(N)=XD(N)*E01                                          HRA2  63
1140   CONTINUE                                                 HRA2  64
145   IF(P02) 175, 150, 175                                     INIT  65
150   IF(NPRT-1)  175, 175, 155
155   WRITE(NOUT,160) IIG                                       INIT  68
160   FORMAT (' GROUP ', I3, 'CROSS-SECTIONS')
      DO 165  N=1,MT                                            INIT  70
165   WRITE (NOUT,170 )          N,(CO(I,N),I=1,ITL)            INIT  71
 170  FORMAT(4H MAT,I3,(6E12.4))
175   WRITE (NSCRAT)   ((CO(I,J),I=1,ITL),J=1,MT)               INIT  73
      WRITE(11) (XD(J),J=1,MT)                                  HRA2
180   CONTINUE                                                  INIT  74
      REWIND NCR1                                               INIT  75
      REWIND 11                                                 HRA2  75
      REWIND 12                                                 HRA2  75
C      WRITE(12) (XD(J),J=1,MT)                                 HRA2  75
C      WRITE(11) (XR(J),J=1,MT)                                 HRA2  75
      REWIND 11                                                 HRA2  75
      REWIND 12                                                 HRA2  75
      REWIND NSCRAT                                             INIT  76
C     SWITCH TAPE DESIGNATIONS                                  INIT  77
      ITEMP=NSCRAT                                              INIT  78
      NSCRAT=NCR1                                               INIT  79
      NCR1=ITEMP                                                INIT  80
185   IF(I04-5)  190, 205, 190                                  INIT  81
190   IF(BUCK)  200, 245, 200                                   INIT  82
200   TEMP = BUCK                                               INIT  83
```

```
      GO TO 220                                                         INIT  84
205   IF(P02)  210, 210, 215                                            INIT  85
210   BUCK = 0.                                                         INIT  86
215   TEMP = EV - BUCK                                                  INIT  87
      BUCK = EV                                                         INIT  88
220   DO 240   IIG=1,IGM                                                INIT  89
      READ(NCR1)  ((CO(I,J), I=1,ITL),J=1,MT)                           INIT  90
      DO 235  MTZ = 1,MT                                                INIT  91
      DO 230   KZ=1,IZM                                                 INIT  92
      IF(M2(KZ) - MTZ)  230, 225, 230                                   INIT  93
225   TEMP1=(TEMP*GAM(KZ))/(3.*CO(IHT,MTZ))                             HRA2  94
      CO(IHT-2,MTZ) = CO(IHT-2,MTZ) + TEMP1                             HRA2  95
      CO(IHS,MTZ) = CO(IHS,MTZ) - TEMP1                                 HRA2  96
      GO TO 235                                                         INIT  97
230   CONTINUE                                                          INIT  98
235   CONTINUE                                                          INIT  99
      WRITE(NSCRAT)  ((CO(I,J), I=1,ITL),J=1,MT)                        INIT 100
240   CONTINUE                                                          INIT 101
      REWIND NCR1                                                       INIT 102
      REWIND NSCRAT                                                     INIT 103
C     SWITCH TAPE DESIGNATIONS                                          INIT 104
      ITEMP = NSCRAT                                                    INIT 105
      NSCRAT = NCR1                                                     INIT 106
      NCR1 = ITEMP                                                      INIT 107
245   CONTINUE                                                          INIT 108
C                                                                        WLP
C     MODIFY GEOMETRY                                                    WLP
      IF(P02)270, 250, 270                                               WLP
250   IF(NCON)  375, 255, 375                                            WLP
255   DO 260  I=1,IP                                                     WLP
260   R1(I)=R0(I)                                                        WLP
      DO 265 J=1,JP                                                      WLP
265   Z1(J)=Z0(J)                                                        WLP
270   IF(I04-4)  305, 275, 305                                           WLP
275   DO 280 I=1,IM                                                      WLP
      K=R2(I)                                                            WLP
280   R1(I+1)=R1(I)+(R0(I+1)-R0(I))*(1.0+ EV*R3(K))                      WLP
      DO 285 J=1,JM                                                      WLP
```

```
      K=Z2(J)                                                   WLP
285   Z1(J+1)=Z1(J)+(Z0(J+1)-Z0(J))*(1.0+ EV*Z3(K))            WLP
      IF(IGE-2) 305, 290, 305                                   WLP
290   IF(ABS (Z1(JP)-1.0)-1.0E-04) 305, 305, 300                WLP
300   CALL ERRO2('**INIT',300,1)                                HVX
305   CONTINUE                                                  WLP
C                                                               WLP
C     AREAS AND VOLUMES                                         WLP
      PI2=6.28318                                               WLP
      IF(PO2) 310, 315, 310                                     WLP
310   IF(I04 - 4)  375, 315, 375                                WLP
315   CONTINUE                                                  WLP
      DO 345 I=1,IM                                             WLP
      R4(I)=(R1(I+1)+R1(I))*0.5                                 WLP
      R5(I)=R1(I+1)-R1(I)                                       WLP
      IF( R5(I) ) 320, 320, 325                                 WLP
320   CALL ERRO2 ('*R5(I)',320,1)                               HVX
325   CONTINUE                                                  WLP
      GO TO (330,335,340,342), IGEP                             WLP
330   A0(I)=1.0                                                 WLP
      A0(IP)=1.0                                                WLP
      A1(I)=R5(I)                                               WLP
      GO TO 345                                                 WLP
335   A0(I)=PI2*R1(I)                                           WLP
      A0(IP)=PI2*R1(IP)                                         WLP
      A1(I)=PI2*R5(I)*R4(I)                                     WLP
      GO TO 345                                                 WLP
340   A0(I)=R1(I)                                               HRA2

      A0(IP)=R1(IP)                                             HRA2
      A1(I)=R5(I)                                               HRA2

      GO TO 345                                                 WLP
342   A0(I) = 2.*R5(I)                                          WLP
      A0(IP) = 2.*R5(I)                                         WLP
      A1(I) = 2.*R5(I)                                          WLP
345   CONTINUE                                                  WLP
      DO 370  J=1,JM                                            WLP
```

```
      Z4(J)=(Z1(J+1)+Z1(J))*0.5                                  WLP
      Z5(J)=Z1(J+1)-Z1(J)                                        WLP
      IF(Z5(J)) 350, 350, 355                                    WLP
350   CALL ERRO2 ('*Z5(J)',350,1)                                HVX
355   CONTINUE                                                   WLP
      DO 370  I=1,IM                                             WLP
      GO TO (360,365,367,360), IGEP                              HRA2
360   V0(I,J)=R5(I)*Z5(J)                                        WLP
      GO TO 370                                                  WLP
365   V0(I,J)=PI2*R5(I)*Z5(J)*R4(I)                              HRA2
      GO TO 370                                                  HRA2

367   V0(I,J)=(R5(I)*Z5(J)*R4(I)+((R1(I+1))**3-(R1(I))**3)/3.0   HRA2
     1 /BIGR*(SIN(Z1(J+1))-SIN(Z1(J))))                          HRA2
370   CONTINUE                                                   WLP
375   CONTINUE                                                   WLP
C                                                                INIT 170
C     MATERIAL ADDRESSES                                         INIT 171
380   IF(P02) 405, 385, 405                                      INIT 172
385   SK7=0.                                                     INIT 173
      DO 400 IIG=1,IGM                                           INIT 174
      IF(S02-1) 395, 390, 395                                    INIT 175
390   K6(IIG)=K7(IIG)/S03                                        INIT 176
      GO TO 400                                                  INIT 177
395   K6(IIG)=K7(IIG)                                            INIT 178
400   SK7=SK7+K7(IIG)                                            INIT 179
405   CONTINUE                                                   INIT 180
C                                                                INIT 181
C     FISSION NEUTRONS                                           INIT 182
      T11=E1(IGP)                                                INIT 183
410   CALL CLEAR(0.0,F0,IMJM)                                    INIT 184
      DO 425 IIG=1,IGM                                           INIT 185
      E0(IIG) = .0                                               INIT 186
      READ (NFLUX1)     (N0(I),I=1,IMJM)                         INIT 187
      READ (NCR1)    ((C0(I,J),I=1,ITL),J=1,MT)                  INIT 188
      DO 425  J = 1, JM                                          INIT 189
      DO 425  K = 1, IM                                          INIT 190
      I = K + (J-1)*IM                                           INIT 191
```

```
      ITEMP=N0(I)                                                  INIT 192
      ITEMP=N2(ITEMP)                                              INIT 193
      EO(IIG) = EO(IIG) + VO(K,J)*NO(I)*CO(IHT-3,ITEMP)            HRA2 194
      IF(A02) 415, 420, 415                                        INIT 195
415   FO(I)=FO(I)+K7(IIG)*NO(I)                                    INIT 196
      GO TO 425                                                    INIT 197
420   FO(I)=FO(I)+CO(IHT-1,ITEMP)*NO(I)                            INIT 198
425   CONTINUE                                                     INIT 199
      REWIND NFLUX1                                                INIT 200
      REWIND NCR1                                                  INIT 201
      RETURN                                                       INIT 202
      END                                                          INIT 203
      SUBROUTINE INNER(NO, N2, CXS, S2, MO, M2, VO, CO,JIM,JJM, JTL,  INNR   2
     1                CXR,CXT, HA, PA,XR,XD)                       HRA2
      INCLUDE 'ABC.FOR'

      DIMENSION  NO(1), N2(1),CXS(JIM,JJM,3),S2(1), MO(1), M2(1),    INNR   6
     1           VO(1), CO(JTL,1), CXR(1), CXT(1), HA(1), PA(1)      INNR   7
      DIMENSION XD(50),XR(50)                                       HRA2
      CALL IFLUXN (N2,CO,VO,CXS,MO, M2, ITL, IM, JM, CXR,CXT,XR,XD)  HRA2   9
2     DO 4  I=1, IMJM                                               INNR  18
4     NO(I) = N2(I)                                                 INNR  19
C     BEGIN FLUX CALCULATION                                        INNR  20
      IKB = IM - 1                                                  INNR  21
      JKB = JM - 1                                                  INNR  22
C     FLUX CALCULATION USING SOR WITH LINE INVERSION                INNR  23
C                                                                   INNR  24
C     CALCULATION OF LEFT BOUNDARY FLUX                             INNR  25
      KI = 1                                                        INNR  26
      KJ = 1                                                        INNR  27
      I = KI + (KJ - 1)*IM                                          INNR  28
      HA(KJ)= CXS(KI,KJ+1,2)/CXS(KI,KJ,3)                           INNR  29
      PA(KJ)= (S2(I) + CXS(KI+1,KJ,1)*N2(I+1))/CXS(KI,KJ,3)         INNR  30
      DO  5 KJ = 2,JKB                                              INNR  31
      I = KI + (KJ - 1)*IM                                          INNR  32
      HA(KJ) = CXS(KI,KJ+1,2)/(CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))  INNR  33
5     PA(KJ) = (S2(I) + CXS(KI+1,KJ,1)*N2(I+1) + CXS(KI,KJ,2)*PA(KJ-1))/INNR  34
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))                       INNR  35
```

```
      KJ = JM
      I = KI + (KJ - 1)*IM
      N2(I)  = (S2(I) + CXS(KI+1,KJ,1)*N2(I+1) + CXS(KI,KJ,2)*PA(KJ-1))/
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))
      DO 10 KJJ = 2,JM
      KJ = JM - KJJ + 1
      I = KI + (KJ - 1)*IM
10    N2(I) = PA(KJ) + HA(KJ) * N2(I+IM)
      DO 15 KJ = 1,JM
      I = KI + (KJ - 1)*IM
      N2(I)=NO(I)+ORF*(N2(I)-NO(I))
   15 IF(N2(I).LE.0)N2(I)=ABS(NO(I)+(N2(I)-NO(I))/ORF)
C     PRINCIPAL FLUX LOOP
      DO 40 KI = 2,IKB
      KJ = 1
      I = KI + (KJ - 1)*IM
      HA(KJ)= CXS(KI,KJ+1,2)/CXS(KI,KJ,3)
      PA(KJ)= (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI+1,KJ,1)*N2(I+1))/
     1  CXS(KI,KJ,3)
      DO 25 KJ = 2,JKB
      I = KI + (KJ - 1)*IM
      HA(KJ) = CXS(KI,KJ+1,2)/(CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))
25    PA(KJ) = (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI+1,KJ,1)*N2(I+1) +
     1 CXS(KI,KJ,2)*PA(KJ-1))/(CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))
      KJ = JM
      I = KI + (KJ - 1)*IM
      N2(I)  = (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI+1,KJ,1)*N2(I+1) +
     1 CXS(KI,KJ,2)*PA(KJ-1))/(CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))
      DO 30 KJJ = 2,JM
      KJ = JM - KJJ + 1
      I = KI + (KJ - 1)*IM
30    N2(I) = PA(KJ) + HA(KJ) * N2(I+IM)
      DO 35 KJ = 1,JM
      I = KI + (KJ - 1)*IM
      N2(I)=NO(I)+ORF*(N2(I)-NO(I))
   35 IF(N2(I).LE.0)N2(I)=ABS(NO(I)+(N2(I)-NO(I))/ORF)
40    CONTINUE
C     CALCULATION OF RIGHT BOUNDARY FLUX
```

```
      KI = IM                                                           INNR  74
      KJ = 1                                                            INNR  75
      I = KI + (KJ - 1)*IM                                              INNR  76
      HA(KJ)= CXS(KI,KJ+1,2)/CXS(KI,KJ,3)                               INNR  77
      PA(KJ)= (S2(I) + CXS(KI,KJ,1)*N2(I-1))/CXS(KI,KJ,3)               INNR  78
      DO 45 KJ = 2,JKB                                                  INNR  79
      I = KI + (KJ - 1)*IM                                              INNR  80
      HA(KJ) = CXS(KI,KJ+1,2)/(CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))     INNR  81
45    PA(KJ) = (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI,KJ,2)*PA(KJ-1))/  INNR  82
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))                          INNR  83
      KJ = JM                                                           INNR  84
      I = KI + (KJ - 1)*IM                                              INNR  85
      N2(I)  = (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI,KJ,2)*PA(KJ-1))/  INNR  86
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))                          INNR  87
      DO 50 KJJ = 2,JM                                                  INNR  88
      KJ = JM - KJJ + 1                                                 INNR  89
      I = KI + (KJ - 1)*IM                                              INNR  90
50    N2(I) = PA(KJ) + HA(KJ) * N2(I+IM)                                INNR  91
      DO 55 KJ = 1,JM                                                   INNR  92
      I = KI + (KJ - 1)*IM                                              INNR  93
      N2(I)=NO(I)+ORF*(N2(I)-NO(I))                                     INNR  94
   55 IF(N2(I).LE.0)N2(I)=ABS(NO(I)+(N2(I)-NO(I))/ORF)                  INNR  95
      TEMP1 = .0                                                        INNR  96
      DO 90 I = 1,INJM                                                  INNR  97
      TEMP2 = ABS (1.0 - NO(I)/N2(I))                                   INNR  98
      IF (TEMP1 - TEMP2) 80,90,90                                       INNR 100
80    TEMP1 = TEMP2                                                     INNR 101
90    CONTINUE                                                          INNR 103
C                                                                       INNR 104
C     INNER ITERATION CONTROL                                           INNR 105
133   LC = LC + 1                                                       INNR 106
      II = II + 1                                                       INNR 107
      IF (II - G07) 533, 1033, 1033                                     INNR 117
533   IF(TEMP1-EPS) 633,633,2                                           INNR 118
633   IF(G06)  733, 1033, 733                                           INNR 119
733   IF(TEMP1-G06)1033,1033,2                                          INNR 120
C1033  CONTINUE
1033  WRITE(NOUT,213)                                                   HRA2
```

```
213   FORMAT('                 GROUP     IN. IT.        ')          HRA2
      WRITE(NOUT,2133)IGV,II                                          HRA2
2133  FORMAT('             ',I3,'      ',I4)
      IF(I04.NE.0)GO TO 1133                                          HRA2
      CALL IFLUXL (N2,C0,V0,CXS,M0, M2, ITL, IM, JM, CXR,CXT,XR,XD)   HRA2
1133  CONTINUE                                                        HRA2
      RETURN                                                          INNR 125
      END                                                             INNR 129
      SUBROUTINE INNER1(M0, M2, CXS, V0, C0, A0, Z5, R5, R4, Z4, A1,  INN1   2
     2                 JIM,JJM,JTL,CXR,CXT,XR,XD,R1,Z1)               HRA2   3
      DIMENSION XR(50),XD(50)                                         HRA2
      DIMENSION  M0(1), M2(1),CXS(JIM,JJM,3),V0(1), C0(JTL,1),        INN1   4
     1          A0(1), Z5(1), R5(1), R4(1), Z4(1), A1(1),CXR(1), CXT(1)INN1   5
     2          ,R1(1),Z1(1)                                          HRA2
      INCLUDE 'ABC.FOR'

C     THIS SUBROUTINE CALCULATES COEFFICIENTS FOR THE FLUX EQUATION   INN1   7
      PI2 = 6.28318                                                   INN1   8
      DO  45  KJ = 1, JM                                              INN1   9
      DO  45  KI = 1, IM                                              INN1  10
      TEMPA=A0(KI)                                                    HRA2
      TEMPB=A0(IP)                                                    HRA2
      TEMPC=A1(KI)                                                    HRA2
      GO TO (10,10, 5), IGEP                                          INN1  1#
5     TEMP = (Z4(KJ) - Z4(KJ-1))*R4(KI)                               HRA2  12
      A0(KI)=A0(KI)*(1.0+R1(KI)/BIGR*(SIN(Z1(KJ+1))-SIN(Z1(KJ)))/Z5(KJ))HRA2
      A0(IP)=A0(IP)*(1.0+R1(IP)/BIGR*(SIN(Z1(KJ+1))-SIN(Z1(KJ)))/Z5(KJ))HRA2
      A1(KI)=A1(KI)*(1.0+COS(Z4(KJ))/BIGR)                            HRA2
      GO TO 15                                                        INN1  13
10    TEMP = Z4(KJ) - Z4(KJ-1)                                        INN1  14
15    I = KI + (KJ-1)*IM                                              INN1  15
      ITEMP = M0(I)                                                   INN1  16
      ITEMP = M2(ITEMP)                                               INN1  17
      CXS(KI,KJ,3) = V0(I)*(C0(IHT,ITEMP) - C0(IHS,ITEMP))            INN1  18
      IF(I - 1) 44,44,18                                              HRA2  19
18    ITEMP1 = M0(I-1)                                                INN1  20
      ITEMP1 = M2(ITEMP1)                                             INN1  21
      IF (ITEMP - ITEMP1) 25,20,25                                    INN1  22
```

```
20    CXS(KI,KJ,1)=AO(KI   )*Z5(KJ)/(3.*CO(IHT,ITEMP)*(R4(KI)-R4(KI-1))) INN1  23
      GO TO 30                                                            INN1  24
25    CXS(KI,KJ,1) = AO(KI)*Z5(KJ)*(R5(KI-1)+R5(KI))/((R4(KI)-R4(KI-1))*INN1  25
     1 (3.*(R5(KI-1)*CO(IHT,ITEMP1) + R5(KI)*CO(IHT,ITEMP))))            INN1  26
30    IF(I - IM)  44,44,32                                                INN1  27
32    ITEMP3 = MO(I - IM)                                                 INN1  28
      ITEMP3 = M2(ITEMP3)                                                 INN1  29
      IF (ITEMP - ITEMP3) 40,35,40                                        INN1  30
35    CXS(KI,KJ,2) = A1(KI)/(3.*CO(IHT,ITEMP)*TEMP)                       INN1  31
      GO TO 44                                                            HRA2  32
40    CXS(KI,KJ,2) = A1(KI)*(Z5(KJ-1) + Z5(KJ))/(TEMP*                    INN1  33
     1 (3.*(Z5(KJ-1)*CO(IHT,ITEMP3) + Z5(KJ)*CO(IHT,ITEMP))))            INN1  34
44    AO(KI)=TEMPA                                                        HRA1
      AO(IP)=TEMPB                                                        HRA1
      A1(KI)=TEMPC                                                        HRA1
45    CONTINUE                                                            HRA1  35
      DO 200  KJ = 1, JM                                                  INN1  36
      DO 200  KI = 1, IM                                                  INN1  37
      TEMPA=AO(KI)                                                        HRA2
      TEMPB=AO(IP)                                                        HRA2
      TEMPC=A1(KI)                                                        HRA2
      GO TO (55,55,50), IGEP                                              INN1  3#
50    TEMP = .5*Z5(KJ)*R4(KI)                                             HRA2  39
      AO(KI)=AO(KI)*(1.0+R1(KI)/BIGR*(SIN(Z1(KJ+1))-SIN(Z1(KJ)))/Z5(KJ))HRA2
      AO(IP)=AO(IP)*(1.0+R1(IP)/BIGR*(SIN(Z1(KJ+1))-SIN(Z1(KJ)))/Z5(KJ))HRA2
      A1(KI)=A1(KI)*(1.0+COS(Z4(KJ))/BIGR)                                HRA2
      GO TO 60                                                            INN1  40
55    TEMP = .5*Z5(KJ)                                                    INN1  41
60    I = KI + (KJ-1)*IM                                                  INN1  42
      ITEMP = MO(I)                                                       INN1  43
      ITEMP = M2(ITEMP)                                                   INN1  44
      TEMP1 = CXS(KI+1,KJ,1)                                              INN1  45
      TEMP2 = CXS(KI,KJ+1,2)                                              INN1  46
      IF(KJ - 1)  65,65,100                                               INN1  47
65    IF(B04 - 1)  90,95,70                                               INN1  48
70    GO TO (  80,  80,  75),  IGEP                                       INN1  4#
75    TEMP3 = PI2*.5*(Z5(KJ) + Z5(JM))                                    HRA2  50
      GO TO 85                                                            INN1  51
```

```
80     TEMP3 = .5*(Z5(KJ) + Z5(JM))
85     ITEMP3 = I + IM*(JM - 1)
       ITEMP3 = M0(ITEMP3)
       ITEMP3 = M2(ITEMP3)
       CXS(KI,KJ,2) = A1(KI)*(Z5(JM) + Z5(KJ))/(TEMP3*
      1         (3.*(Z5(JM)*CO(IHT,ITEMP3) + Z5(KJ)*CO(IHT,ITEMP))))
       GO TO 125
90     CXS(KI,KJ,2) = A1(KI)/(3.*CO(IHT,ITEMP)*( TEMP    +.71/
      1 CO(IHT,ITEMP)))
       GO TO 125
95     CXS(KI,KJ,2) = .0
       GO TO 125
100    IF (KJ - JM) 125,105,105
105    IF (B03 - 1) 115,120,110
110    TEMP2 = CXS(KI,1,2)
       CXT(KI) = TEMP2
       GO TO 125
115    TEMP2        = A1(KI)/(3.*CO(IHT,ITEMP)*( TEMP    +.71/
      1 CO(IHT,ITEMP)))
       CXT(KI) = TEMP2
       GO TO 125
120    TEMP2 = .0
       CXT(KI) = TEMP2
125    IF (KI - 1) 130,130,145
130    IF(B01)  135,135,140
135    CXS(KI,KJ,1) = A0(KI  )*Z5(KJ)/(3.*CO(IHT,ITEMP)*
      1 (.5*R5(KI) + .71/CO(IHT,ITEMP)))
       GO TO 165
140    CXS(KI,KJ,1) = .0
       GO TO 165
145    IF (KI - IM) 165,150,150
150    IF(B02)  155,155,160
155    TEMP1        = A0(KI+1)*Z5(KJ)/(3.*CO(IHT,ITEMP)*
      1 (.5*R5(KI) + .71/CO(IHT,ITEMP)))
       CXR(KJ) = TEMP1
       GO TO 165
160    TEMP1 = .0
       CXR(KJ) = TEMP1
```

```
165   CXS(KI,KJ,3) = CXS(KI,KJ,3) + CXS(KI,KJ,1) + CXS(KI,KJ,2)           INN1  90
     1  + TEMP1 + TEMP2                                                    INN1  91
      A0(KI)=TEMPA                                                         HRA1
      A0(IP)=TEMPB                                                         HRA1
      A1(KI)=TEMPC                                                         HRA1
200   CONTINUE                                                             INN1  92
      RETURN                                                               INN1  93
      END                                                                  INN1  94
      SUBROUTINE INNER2(NO, N2, CXS, S2, MO, M2, VO, CO,JIM,JJM, JTL,
     1               CXR,CXT, HA, PA)                                      INN2   3
       INCLUDE 'ABC.FOR'

      DIMENSION  NO(1), N2(1),CXS(JIM,JJM,3),S2(1), MO(1), M2(1),          INN2   6
     1           VO(1), CO(JTL,1), CXR(1), CXT(1), HA(1), PA(1)            INN2   7
      CALL IFLUXN (N2, CO, VO, CXS, MO, M2, ITL, IM, JM, CXR,CXT,XR,XD) HRA2   9
2     DO 4  I=1, IMJM                                                      INN2  18
4     NO(I) = N2(I)                                                        INN2  19
C     BEGIN FLUX CALCULATION                                               INN2  20
      IKB = IM - 1                                                         INN2  21
      JKB = JM - 1                                                         INN2  22
C     FLUX CALCULATION USING SOR WITH LINE INVERSION                       INN2  23
C                                                                          INN2  24
C     CALCULATION OF BOTTOM BOUNDARY FLUX                                  INN2  25
      KI = 1                                                               INN2  26
      KJ = 1                                                               INN2  27
      I = KI + (KJ - 1)*IM                                                 INN2  28
      HA(KI)= CXS(KI+1,KJ,1)/CXS(KI,KJ,3)                                  INN2  29
      PA(KI)= (S2(I) + CXS(KI,KJ+1,2)*N2(I+IM))/CXS(KI,KJ,3)               INN2  30
      DO  5 KI = 2,IKB                                                     INN2  31
      I = KI + (KJ - 1)*IM                                                 INN2  32
      HA(KI) = CXS(KI+1,KJ,1)/(CXS(KI,KJ,3)- CXS(KI,KJ,1)*HA(KI-1))        INN2  33
5     PA(KI) = (S2(I) + CXS(KI,KJ+1,2)*N2(I+IM)+ CXS(KI,KJ,1)*PA(KI-1))/INN2  34
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,1)*HA(KI-1))                             INN2  35
      KI = IM                                                              INN2  36
      I = KI + (KJ - 1)*IM                                                 INN2  37
     N2(I)  = (S2(I) + CXS(KI,KJ+1,2)*N2(I+IM)+ CXS(KI,KJ,1)*PA(KI-1))/INN2  38
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,1)*HA(KI-1))                             INN2  39
      DO 10 KII = 2,IM                                                     INN2  40
```

```
      KI = IM - KII + 1
      I = KI + (KJ - 1)*IM
10    N2(I) = PA(KI) + HA(KI) * N2(I+1)
      DO 15 KI = 1,IM
      I = KI + (KJ - 1)*IM
      N2(I)=NO(I)+ORF*(N2(I)-NO(I))
   15 IF(N2(I).LE.0)N2(I)=ABS(NO(I)+(N2(I)-NO(I))/ORF)
C     PRINCIPAL FLUX LOOP
      DO 40 KJ = 2,JKB
      KI = 1
      I = KI + (KJ - 1)*IM
      HA(KI)= CXS(KI+1,KJ,1)/CXS(KI,KJ,3)
      PA(KI)= (S2(I) + CXS(KI,KJ,2)*N2(I-IM)+ CXS(KI,KJ+1,2)*N2(I+IM))/
     1  CXS(KI,KJ,3)
      DO 25 KI = 2,IKB
      I = KI + (KJ - 1)*IM
      HA(KI) = CXS(KI+1,KJ,1)/(CXS(KI,KJ,3)- CXS(KI,KJ,1)*HA(KI-1))
25    PA(KI) = (S2(I) + CXS(KI,KJ,2)*N2(I-IM)+ CXS(KI,KJ+1,2)*N2(I+IM)+
     1 CXS(KI,KJ,1)*PA(KI-1))/(CXS(KI,KJ,3) - CXS(KI,KJ,1)*HA(KI-1))
      KI = IM
      I = KI + (KJ - 1)*IM
      N2(I)  = (S2(I) + CXS(KI,KJ,2)*N2(I-IM)+ CXS(KI,KJ+1,2)*N2(I+IM)+
     1 CXS(KI,KJ,1)*PA(KI-1))/(CXS(KI,KJ,3) - CXS(KI,KJ,1)*HA(KI-1))
      DO 30 KII = 2,IM
      KI = IM - KII + 1
      I = KI + (KJ - 1)*IM
30    N2(I) = PA(KI) + HA(KI) * N2(I+1)
      DO 35 KI = 1,IM
      I = KI + (KJ - 1)*IM
      N2(I)=NO(I)+ORF*(N2(I)-NO(I))
   35 IF(N2(I).LE.0)N2(I)=ABS(NO(I)+(N2(I)-NO(I))/ORF)
40    CONTINUE
C     CALCULATION OF TOP BOUNDARY FLUX
      KJ = JM
      KI = 1
      I = KI + (KJ - 1)*IM
      HA(KI)= CXS(KI+1,KJ,1)/CXS(KI,KJ,3)
      PA(KI) = (S2(I) + CXS(KI,KJ,2)*N2(I-IM))/CXS(KI,KJ,3)
```

```
      DO 45 KI = 2,IKB                                                    INN2  79
      I = KI + (KJ - 1)*IM                                                INN2  80
      HA(KI) = CXS(KI+1,KJ,1)/(CXS(KI,KJ,3)- CXS(KI,KJ,1)*HA(KI-1))       INN2  81
45    PA(KI) = (S2(I) + CXS(KI,KJ,2)*N2(I-IM)+ CXS(KI,KJ,1)*PA(KI-1))/    INN2  82
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,1)*HA(KI-1))                            INN2  83
      KI = IM                                                             INN2  84
      I = KI + (KJ - 1)*IM                                                INN2  85
      N2(I)  = (S2(I) + CXS(KI,KJ,2)*N2(I-IM)+ CXS(KI,KJ,1)*PA(KI-1))/    INN2  86
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,1)*HA(KI-1))                            INN2  87
      DO 50 KII = 2,IM                                                    INN2  88
      KI = IM - KII + 1                                                   INN2  89
      I = KI + (KJ - 1)*IM                                                INN2  90
50    N2(I) = PA(KI) + HA(KI) * N2(I+1)                                   INN2  91
      DO 55 KI = 1,IM                                                     INN2  92
      I = KI + (KJ - 1)*IM                                                INN2  93
      N2(I)=NO(I)+ORF*(N2(I)-NO(I))                                       INN2  94
   55 IF(N2(I).LE.0)N2(I)=ABS(NO(I)+(N2(I)-NO(I))/ORF)                    INN2  95
      TEMP1 = .0                                                          INN2  96
      DO 90 I = 1,IMJM                                                    INN2  97
      TEMP2 = ABS (1.0 - NO(I)/N2(I))                                     INN2  98
      IF (TEMP1 - TEMP2) 80,90,90                                         INN2 100
80    TEMP1 = TEMP2                                                       INN2 101
90    CONTINUE                                                            INN2 103
C                                                                         INN2 104
C     INNER ITERATION CONTROL                                             INN2 105
133   LC = LC + 1                                                         INN2 106
      II = II + 1                                                         INN2 107
      IF (II - G07) 533, 1033, 1033                                       INN2 117
533   IF(TEMP1-EPS) 633,633,2                                             INN2 118
633   IF(G06)  733, 1033, 733                                             INN2 119
733   IF(TEMP1-G06)1033,1033,2                                            INN2 120
1033  CONTINUE
      RETURN                                                              INN2 125
      END                                                                 INN2 129
      SUBROUTINE INNERP(NO, N2, CXS, S2, MO, M2, VO, CO,JIM,JJM, JTL,     INNP   2
     1                 CXR,CXT, HA, PA)                                   INNP   3
      INCLUDE 'ABC.FOR'
```

```
      DIMENSION  NO(1), N2(1),CXS(JIM,JJM,3),S2(1), MO(1), M2(1),       INNP
     1           VO(1), CO(JTL,1), CXR(1), CXT(1), HA(1), PA(1)         INNP
C     THIS SUBROUTINE CALCULATES THE FLUX FOR PERIODIC B. C.            INNP
      CALL IFLUXN (N2, CO, VO, CXS, MO, M2, ITL, IM, JM, CXR,CXT,XR,XD) HRA2
2     DO 4  I=1, IMJM                                                   INNP
4     NO(I) = N2(I)                                                     INNP
C     BEGIN FLUX CALCULATION                                            INNP
      IKB = IM - 1                                                      INNP
      JKB = JM - 1                                                      INNP
C     FLUX CALCULATION USING SOR WITH LINE INVERSION                    INNP
C                                                                       INNP
C     CALCULATION OF LEFT BOUNDARY FLUX                                 INNP
      KI = 1                                                            INNP
      KJ = 1                                                            INNP
      I = KI + (KJ - 1)*IM                                              INNP
      N2(I) = CXS(KI,1,2)/CXS(KI,KJ,3)                                  INNP
      HA(KJ)= CXS(KI,KJ+1,2)/CXS(KI,KJ,3)                               INNP
      TEMP1 = N2(I)                                                     INNP
      TEMP = HA(1)                                                      INNP
      PA(KJ)= (S2(I) + CXS(KI+1,KJ,1)*N2(I+1))/CXS(KI,KJ,3)             INNP
      TEMP2 = PA(KJ)                                                    INNP
      DO  5 KJ = 2,JKB                                                  INNP
      I = KI + (KJ - 1)*IM                                              INNP
      HA(KJ) = CXS(KI,KJ+1,2)/(CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))     INNP
      N2(I) = CXS(KI,KJ,2) * N2(I-IM)/                                  INNP
     1                         (CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))     INNP
      TEMP1 = TEMP1 + TEMP*N2(I)                                        INNP
      PA(KJ) = (S2(I) + CXS(KI+1,KJ,1)*N2(I+1) + CXS(KI,KJ,2)*PA(KJ-1))/INNP
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))                          INNP
      TEMP2 = TEMP2 + TEMP*PA(KJ)                                       INNP
5     TEMP = TEMP*HA(KJ)                                                INNP
      KJ = JM                                                           INNP
      I = KI + (KJ - 1)*IM                                              INNP
      TEMP1 =(TEMP1 +  TEMP)*CXS(KI,1,2) + CXS(KI,KJ,2)*N2(I-IM)        INNP
      N2(I)  = (S2(I) + CXS(KI+1,KJ,1)*N2(I+1) + CXS(KI,KJ,2)*PA(KJ-1)  INNP
     1  + CXS(KI,1,2)*TEMP2 )/                                          INNP
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1) - TEMP1)                  INNP
      DO 10 KJJ = 2,JM                                                  INNP
```

```
      KJ = JM - KJJ + 1
      I = KI + (KJ - 1)*IM
      KII = (JM-1)*IM + KI
10    N2(I) = PA(KJ) + HA(KJ) * N2(I+IM)  + N2(I) * N2(KII)
      DO 15 KJ = 1,JM
      I = KI + (KJ - 1)*IM
15    N2(I) = NO(I) + ORF*(N2(I) - NO(I))
C     PRINCIPAL FLUX LOOP
      DO 40 KI = 2,IKB
      KJ = 1
      I = KI + (KJ - 1)*IM
      HA(KJ)= CXS(KI,KJ+1,2)/CXS(KI,KJ,3)
      N2(I) = CXS(KI,1,2)/CXS(KI,KJ,3)
      TEMP1 = N2(I)
      TEMP = HA(1)
      PA(KJ)= (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI+1,KJ,1)*N2(I+1))/
     1  CXS(KI,KJ,3)
      TEMP2 = PA(KJ)
      DO 25 KJ = 2,JKB
      I = KI + (KJ - 1)*IM
      HA(KJ) = CXS(KI,KJ+1,2)/(CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))
      N2(I) = CXS(KI,KJ,2) * N2(I-IM)/
     1                          (CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))
      TEMP1 = TEMP1 + TEMP*N2(I)
      PA(KJ) = (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI+1,KJ,1)*N2(I+1) +
     1 CXS(KI,KJ,2)*PA(KJ-1))/(CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))
      TEMP2 = TEMP2 + TEMP*PA(KJ)
25    TEMP = TEMP*HA(KJ)
      KJ = JM
      I = KI + (KJ - 1)*IM
      TEMP1 =(TEMP1 +  TEMP)*CXS(KI,1,2) + CXS(KI,KJ,2)*N2(I-IM)
      N2(I)  = (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI+1,KJ,1)*N2(I+1) +
     1 CXS(KI,1,2)*TEMP2 +
     1 CXS(KI,KJ,2)*PA(KJ-1))/(CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1) -
     2 TEMP1)
      DO 30 KJJ = 2,JM
      KJ = JM - KJJ + 1
      I = KI + (KJ - 1)*IM
```

```
      KII = (JM-1)*IM + KI
30    N2(I) = PA(KJ) + HA(KJ) * N2(I+IM)  + N2(I) * N2(KII)
      DO 35 KJ = 1,JM
      I = KI + (KJ - 1)*IM
35    N2(I) = NO(I) + ORF*(N2(I) - NO(I))
40    CONTINUE
C     CALCULATION OF RIGHT BOUNDARY FLUX
      KI = IM
      KJ = 1
      I = KI + (KJ - 1)*IM
      HA(KJ)= CXS(KI,KJ+1,2)/CXS(KI,KJ,3)
      N2(I) = CXS(KI,1,2)/CXS(KI,KJ,3)
      TEMP1 = N2(I)
      TEMP = HA(1)
      PA(KJ)= (S2(I) + CXS(KI,KJ,1)*N2(I-1))/CXS(KI,KJ,3)
      TEMP2 = PA(KJ)
      DO 45 KJ = 2,JKB
      I = KI + (KJ - 1)*IM
      HA(KJ) = CXS(KI,KJ+1,2)/(CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))
      N2(I) = CXS(KI,KJ,2) * N2(I-IM)/
     1                      (CXS(KI,KJ,3)- CXS(KI,KJ,2)*HA(KJ-1))
      TEMP1 = TEMP1 + TEMP*N2(I)
      PA(KJ) = (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI,KJ,2)*PA(KJ-1))/
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1))
      TEMP2 = TEMP2 + TEMP*PA(KJ)
45    TEMP = TEMP*HA(KJ)
      KJ = JM
      I = KI + (KJ - 1)*IM
      TEMP1 =(TEMP1 +  TEMP)*CXS(KI,1,2) + CXS(KI,KJ,2)*N2(I-IM)
      N2(I)  = (S2(I) + CXS(KI,KJ,1)*N2(I-1) + CXS(KI,KJ,2)*PA(KJ-1)
     1  + CXS(KI,1,2)*TEMP2)/
     1  (CXS(KI,KJ,3) - CXS(KI,KJ,2)*HA(KJ-1) - TEMP1)
      DO 50 KJJ = 2,JM
      KJ = JM - KJJ + 1
      I = KI + (KJ - 1)*IM
      KII = (JM-1)*IM + KI
50    N2(I) = PA(KJ) + HA(KJ) * N2(I+IM)  + N2(I) * N2(KII)
      DO 55 KJ = 1,JM
```

```
      I = KI + (KJ - 1)*IM                                            INNP 119
55    N2(I) = NO(I) + ORF*(N2(I) - NO(I))                             INNP 120
C     CALCULATION OF ERROR CRITERION                                  INNP 121
      TEMP1 = .0                                                      INNP 122
      DO 90 I = 1,IMJM                                                INNP 123
      TEMP2 = ABS (1.0 - NO(I)/N2(I))                                 INNP 124
      IF (TEMP1 - TEMP2) 80,90,90                                     INNP 125
80    TEMP1 = TEMP2                                                   INNP 126
90    CONTINUE                                                        INNP 128
C                                                                     INNP 129
C     INNER ITERATION CONTROL                                         INNP 130
133   LC = LC + 1                                                     INNP 131
      II = II + 1                                                     INNP 132
      IF (II - G07) 533, 1033, 1033                                   INNP 139
533   IF (TEMP1 - EPS)  633,633,2                                     INNP 140
633   IF(G06)  733, 1033, 733                                         INNP 141
733   IF (TEMP1 - G06) 1033, 1033, 2                                  INNP 142
1033  CONTINUE
      RETURN                                                          INNP 144
      END                                                             INNP 145
      SUBROUTINE INNERT(MO, M2, CXS, VO, CO, AO, Z5, R5, R4, Z4, A1,  INNT   2
     2                JIM,JJM,JTL,CXR,CXT)                            INNT   3
      DIMENSION  MO(1), M2(1),CXS(JIM,JJM,3),VO(1), CO(JTL,1),        INNT   4
     1           AO(1), Z5(1), R5(1), R4(1), Z4(1), A1(1),CXR(1), CXT(1)INNT   5
      INCLUDE 'ABC.FOR'

C     THIS SUBROUTINE CALCULATES COEFFICIENTS FOR TRIANGULAR GEOMETRY  INNT   7
      DO 45  KJ = 1, JM                                               INNT   8
      DO 45  KI = 1, IM                                               INNT   9
      TEMP = KI - 2* (KI/2) - (KJ - 2* (KJ/2))                        INNT  10
      TEMP = ABS(TEMP)                                                INNT  11
      I = KI + (KJ-1) *IM                                             INNT  12
      ITEMP = MO(I)                                                   INNT  13
      ITEMP = M2(ITEMP)                                               INNT  14
      CXS(KI,KJ,3) = VO(I)*(CO(IHT,ITEMP) - CO(IHS,ITEMP))            INNT  15
      IF(I - 1)  45, 45, 18                                           INNT  16
18    ITEMP1 = MO(I-1)                                                INNT  17
      ITEMP1 = M2(ITEMP1)                                             INNT  18
```

```
      IF(ITEMP - ITEMP1)  25, 20, 25
20    CXS(KI,KJ,1) = A0(KI)/(2.*CO(IHT,ITEMP)*Z5(1))
      GO TO 30
25    CXS(KI,KJ,1) = A0(KI)/((CO(IHT,ITEMP1) + CO(IHT,ITEMP))*Z5(1))
30    IF(I - IM)  45, 45, 32
32    ITEMP3 = M0(I - IM)
      ITEMP3 = M2(ITEMP3)
      IF(ITEMP - ITEMP3)  40, 35, 40
35    CXS(KI,KJ,2) = A1(KI)*TEMP/(2.* CO(IHT,ITEMP)*Z5(1))
      GO TO 45
40    CXS(KI,KJ,2) = A1(KI)*TEMP/((CO(IHT,ITEMP3)+CO(IHT,ITEMP))*Z5(1))
45    CONTINUE
      DO 200  KJ = 1, JM
      DO 200  KI = 1, IM
      TEMP = KI - 2*(KI/2) - (KJ-2*(KJ/2))
      TEMP = ABS(TEMP)
      I = KI + (KJ-1)*IM
      ITEMP = M0(I)
      ITEMP = M2(ITEMP)
      TEMP1 = CXS(KI+1,KJ,1)
      TEMP2 = CXS(KI,KJ+1,2)
      IF(KJ-1)  65, 65, 100
65    IF(B04-1)  90, 95, 95
90    CXS(KI,KJ,2) = A1(KI)*TEMP/(3.*CO(IHT,ITEMP)*(Z5(1)/3. + .71/
     1 CO(IHT,ITEMP)))
      GO TO 125
95    CXS(KI,KJ,2) = .0
      GO TO 125
100   IF(KJ - JM)  125, 105, 105
105   IF(B03 - 1)  115, 120, 120
115   TEMP = KI - 2*(KI/2) - (KJ + 1- 2*((KJ+1)/2))
      TEMP = ABS(TEMP)
      TEMP2=A1(KI)*TEMP/( 3.*CO(IHT,ITEMP)*(Z5(1)/3.+.71/CO(IHT,ITEMP)))
      CXT(KI) = TEMP2
      GO TO 125
120   TEMP2 = .0
      CXT(KI) = TEMP2
125   IF(KI-1)  130, 130, 145
```

```
130   IF(B01)  135, 135, 140                                            INNT  57
135   CXS(KI,KJ,1) = A0(KI)/(3.*CO(IHT,ITEMP)*(Z5(1)/3.                 INNT  58
     1  + .71/CO(IHT,ITEMP)))                                           INNT  59
      GO TO 165                                                         INNT  60
140   CXS(KI,KJ,1) = .0                                                 INNT  61
      GO TO 165                                                         INNT  62
145   IF(KI - IM)  165, 150, 150                                        INNT  63
150   IF(B02)  155, 155, 160                                            INNT  64
155   TEMP1 = A0(KI+1)/(3.*CO(IHT,ITEMP)*(Z5(1)/3.+.71/CO(IHT,ITEMP)))  INNT  65
      CXR(KJ) = TEMP1                                                   INNT  66
      GO TO 165                                                         INNT  67
160   TEMP1 = .0                                                        INNT  68
      CXR(KJ) = TEMP1                                                   INNT  69
165   CXS(KI,KJ,3) = CXS(KI,KJ,3) + CXS(KI,KJ,1) + CXS(KI,KJ,2)         INNT  70
     1  + TEMP1 + TEMP2                                                 INNT  71
200   CONTINUE                                                          INNT  72
      RETURN                                                            INNT  73
      END                                                               INNT  74
      SUBROUTINE INP                                                    INP    2
       INCLUDE 'ABC.FOR'

      COMMON/PACKED/A(50000)
      DIMENSION IDUM(25), DUM(12)
C     THIS SUBROUTINE CONTROLS THE READING OF ALL INPUT DATA            INP    8
      NCR1 = 3
      NSCRAT = 4                                                        INP   10
      NINP = 5                                                          INP   11
      NOUT = 6                                                          INP   12
      NFLUX1 = 8                                                        INP   13
      NSORCE = 9                                                        INP   14
      WRITE(NOUT,9)                                                     INP   15
    9 FORMAT(1H1)                                                       INP   16
      WRITE (NOUT,10)
10    FORMAT (24X, '* * * * * *  P 2 D B * * * * * *')
      READ(NINP,20,END=14) (ID(I),I=1,20)
      GO TO 15
   14 STOP
   15 CONTINUE
```

```
20     FORMAT(20A4)
       WRITE(NOUT,30) (ID(I),I=1,20)
30     FORMAT (1X, 20A4//)
       I=13                                                            HRA2
       CALL REAI2 ('    INP', IDUM(1),I)
       I=14                                                            HRA2
       CALL REAI2 ('    INP', IDUM(14),I)                              HRA2
       I=6
       CALL REAG2 ('    INP', DUM(1),I)
       CALL REAG2 ('    INP', DUM(7),I)
       WRITE (NSCRAT) (IDUM(I), I=1,27)                                HRA2
       WRITE (NSCRAT) (DUM(I), I=1,12)
       REWIND NSCRAT
       READ (NSCRAT) A02, I04, S02, IGM,IHT, NXCM, MCR, G07, D05, MAXT,  HRA2
      1 NPRT,M07, NPUN, IGE, ITOR, NACT, IM, JM, IZM, MT, M01, B01,     HRA2
      2 B02, B03, B04,IZ, JZ
       WRITE(NOUT,60)   A02, I04, S02, IGM,IHT, NXCM, MCR              HRA2
60     FORMAT(
      1 I6,'  A02   0/1=REGULAR CALCULATION/ADJOINT CALCULATION'/
      2 I6,'  I04   EIGENVALUE TYPE (0/1/2/3/4/5=SOURCE/K/ALPHA/CONC/'
      3,'DELTA/BUCK)'/
      5 I6,'  S02   PARAMETRIC EIGENVALUE TYPE (0/1/2=NONE/K/ALPHA)'/
      7 I6,'  IGM   NUMBER OF GROUPS'/
      4 I6,'  IHT   POSITION OF SIGMA TRANSPORT'/                      HRA2
      9 I6,' NXCM   NUMBER OF DOWNSCATTERING TERMS'/
      2 I6,'  MCR   NUMBER OF MATERIALS FROM CARDS/TAPE (+N/-N)')
       WRITE (NOUT,70) G07, D05, MAXT, NPRT, M07, NPUN
70     FORMAT(
      1 I6,'  G07   INNER ITERATION MAX PER GROUP (IF NEG, ALT DIR)'/
      3,I6,'  D05   MAXIMUN NUMBER OF OUTER ITERATIONS'/
      5 I6,' MAXT   MAXIMUM TIME IN MINUTES (IF 0, NO EFFECT)'/
      7 I6,' NPRT   PRINT OPTION (0/1/2/3=MINI/MIDI/XS/FLUXES)'/
      9 I6,'  M07   FLUX GUESS (0/1=NONE/FLUX FROM TAPE 14)'/
      2 I6,' NPUN   FLUX DUMP (0/1=NONE/FLUX DUMP TO TAPE 16)'/)
       WRITE(NOUT,80)  IGE, ITOR , NACT, IM, JM, IZM, MT, M01          HRA2  60
80     FORMAT(                                                         INP   61
      1 I6,'  IGE   GEOMETRY (0/1/2/3=X-Y/R-Z/R-THETA/TRIANGULAR)'/
      4 I6,'  ITOR  TOROIDAL SPECIFIER (0/1=R-THETA/TOROIDAL)'/        HRA2
```

```
    6 I6,'  NACT  ACTIVITIES (0/>0/=NO EFFECT/READ TABLE POSITIONS'    HRA2
    8,' FOR ACTIVITIES)'/                                              HRA2
    3 I6,'   IM   NUMBER OF RADIAL INTERVALS'/
    5 I6,'   JM   NUMBER OF AXIAL INTERVALS'/
    7 I6,'  IZM   NUMBER OF ZONES'/
    9 I6,'   MT   TOTAL NUMBER OF MATERIALS INCLUDING MIXES'/
    2 I6,'  M01   NUMBER OF MIXTURE SPECIFICATIONS')
     WRITE(NOUT,90)  B01, B02, B03, B04, IZ, JZ                        INP   74
90   FORMAT(                                                           INP   75
    1 I6,'  B01   LEFT BOUNDARY CONDITION (0/1=VACUUN/REFLECTIVE)'/
    3 I6,'  B02   RIGHT BOUNDARY CONDITION (0/1=VACUUM/REFLECTIVE)'/
    5 I6,'  B03   TOP BOUNDARY CONDITION (0/1/2=VACUUM/REFLECT/'
    6,'PERIODIC)'/
    7 I6,'  B04   BOTTOM BOUNDARY CONDITION (0/1/2=VACUUM/REFL/'
    8,'PERIODIC)'/
    9 I6,'   IZ   NUMBER OF RADIAL ZONES (DELTA OPTION ONLY)'/
    2 I6,'   JZ   NUMBER OF AXIAL ZONES (DELTA OPTION ONLY)'/)
     READ(NSCRAT)  EV, EVM, S03, BUCK,  LAL,  LAH, EPS, EPSA, G06,
    1                POD, ORF, S01                                     INP   99
     REWIND NSCRAT
     WRITE(NOUT,110)  EV, EVM, S03, BUCK, LAL, LAH                     INP  101
110  FORMAT(                                                           INP  102
    1 1X,1PE11.4,'   EV    FIRST EIGENVALUE GUESS'/
    3 1X,1PE11.4,'  EVM    EIGENVALUE MODIFIER'/
    5 1X,1PE11.4,'  S03    PARAMETRIC EIGENVALUE'/
    7 1X,1PE11.4,' BUCK    BUCKLING (CM-2)'/
    9 1X,1PE11.4,'  LAL    LAMBDA LOWER'/
    2 1X,1PE11.4,'  LAH    LAMBDA UPPER'/)
     WRITE (NOUT,120) EPS,EPSA,G06,POD,ORF,S01
120  FORMAT(                                                           INP  116
    1 1X,1PE11.4,'  EPS    EIGENVALUE CONVERGENCE CRITERION'/
    3 1X,1PE11.4,' EPSA    PARAMETER CONVERGENCE CRITERION'/
    5 1X,1PE11.4,'  G06    INNER ITERATION TEST (IF ZERO, NO TEST)'/
    7 1X,1PE11.4,'  POD    PARAMETER OSCILLATION DAMPER'/
    9 1X,1PE11.4,'  ORF    OVER-RELAXATION FACTOR'/
    2 1X,1PE11.4,'  S01    -N/+N=POWER (MWT)/NEUTRON SOURCE RATE')
     IF(ITOR.EQ.1)THEN                                                 HRA2
     I=1                                                               HRA2
```

```
      CALL RREAG2 ('   INP', BIGR,1)                                   HRA2
      WRITE(6,11)BIGR                                                   HRA2
11    FORMAT(2X,'BIGR=',E13.6)                                          HRA2
      ELSE                                                              HRA2
      BIGR=1.0E+20                                                      HRA2
      ENDIF
      S04=0
      IF (G07.LE.0) S04=1
      G07=ABS(G07)
      MTP=0
      IF (MCR .LE. 0) MTP=-MCR
      IF (MCR .LE. 0) MCR=0
      IF(IZ + JZ)  230, 210, 230                                        INP  129
210   IF(I04 - 4)  230, 220, 230                                        INP  130
220   CALL ERRO2 ('***I04',220,1)                                       HVX
230   CONTINUE                                                          INP  132
      IF(S02)   240, 260, 240                                           INP  133
240   IF(S03)  260, 250, 260                                            INP  134
250   CALL ERRO2 ('***S03',250,1)
260   CONTINUE                                                          INP  136
      FEF = 200.0                                                       INP  137
      TSD = FEF*1.602*10.**(-19)                                        INP  138
CCCC  COMMENT OUT CALL SETTIM ON A VAX                                  HRA2
C      CALL SETTIM (0,0,0,0)                                             HRA2
      KPAGE = 100                                                       INP  140
      IHS = IHT+1                                                       HRA  141
      ITL = NXCM + IHS                                                  HRA  142
C      IHT = 4                                                           HRA  14
3
      IZP = IZM + 1                                                     INP  144
      IP  = IM + 1                                                      INP  145
      JP  = JM + 1                                                      INP  146
      ML = MCR + MTP
      IGP = IGM + 1                                                     INP  148
      IGEP = IGE + 1                                                    INP  149
      IMJM = IM*JM                                                      INP  150
      EQ = .0                                                           INP  151
      LAP = .0                                                          INP  152
```

```
      LAPP = .0                                                         INP  153
      LAR = 0.0                                                         INP  154
      DAY = 0.0                                                         INP  155
      ALA = .0                                                          INP  156
      LC = 0                                                            INP  157
      P02 = 0                                                           INP  158
      CVT = 0                                                           INP  159
      CNT = 0                                                           INP  160
      NCON = 0                                                          INP  161
      T06 = 0                                                           INP  163
      IF(I04-4)  310, 300, 310                                          INP  164
300   T06 = 1                                                           INP  165
310   CONTINUE                                                          INP  166
      ORFP =1.0*(ORF - 1.0) + 1.0                                       INP  167
C     COMPUTE DIMENSION POINTERS
      LHOLN = 1                                                         HVX
      LATW = LHOLN + 2*ML
      LALAM = LATW + ML                                                 HVX
      LCO = LALAM + ML
      LNO = LCO + ITL*MT                                                INP  174
      LN2 = LNO + IMJM                                                  INP  175
      LAO = LN2 + IMJM                                                  INP  176
      LA1 = LAO + IP                                                    INP  177
      LFO = LA1 + IM                                                    INP  178
      LF2 = LFO + IMJM                                                  INP  179
      LIO = LF2 + IMJM                                                  INP  180
      LI1 = LIO + M01                                                   INP  181
      LI2 = LI1 + M01                                                   INP  182
      LI3 = LI2 + M01                                                   INP  183
      LK6 = LI3 + M01                                                   INP  184
      LK7 = LK6 + IGM                                                   INP  185
      LMO = LK7 + IGM                                                   INP  186
      LM2 = LMO + IMJM                                                  INP  187
      LRO = LM2 + IZM                                                   INP  188
      LR1 = LRO + IP                                                    INP  189
      LR2 = LR1 + IP                                                    INP  190
      LR3 = LR2 + T06*IM                                                INP  191
      LR4 = LR3 + T06*IZ                                                INP  192
```

```
      LR5 = LR4 + IM                                                    INP  193
      LS2 = LR5 + IM                                                    INP  194
      LVO = LS2 + IMJM                                                  INP  195
      LV7 = LVO + IMJM                                                  INP  196
      LZO = LV7 + IGM                                                   INP  197
      LZ1 = LZO + JP                                                    INP  198
      LZ2 = LZ1 + JP                                                    INP  199
      LZ3 = LZ2 + JM*TO6                                                INP  200
      LZ4 = LZ3 + JZ*TO6                                                INP  201
      LZ5 = LZ4 + JM                                                    INP  202
      LCXS = LZ5 + JM                                                   INP  203
      LVOL = LCXS + IMJM*3                                              INP  204
      LMASS = LVOL + IZM                                                INP  205
      LMATN = LMASS + ML*IZM                                            INP  206
      LNBR = LMATN + ML                                                 INP  207
      LLD = LNBR + ML                                                   INP  208
      LLCN = LLD + ML                                                   INP  209
      LLFN = LLCN + ML*2                                                INP  210
      LPHIB = LLFN + ML*7                                               INP  211
      LAXS = LPHIB + IZM                                                INP  212
C      LT6=LAXS+ML*IZM                                                   HRA
C      LT8=LT6+ML*IZM                                                    HRA
C      LNTWON=LT8+ML*IZM                                                 HRA
CC      LFXS = LAXS + ML*IZM                                              INP  2
13
C      LFXS = LNTWON + ML*IZM                                            HRA  21
3
      LACT=LAXS+ML*IZM                                                  HRA3
      LACPOS=LACT+ML*IZM*NACT*ML                                          HRA3
      LFXS=LACPOS+NACT                                                  HRA3
      LMASSP = LFXS + ML*IZM                                            INP  214
      LCXR = LMASSP + ML*IZM                                            INP  215
      LCXT = LCXR + JM                                                  INP  216
      LHA = LCXT + IM                                                   INP  217
      LPA = LHA + MAXO(IM,JM)                                           INP  218
      LGAM=LPA + MAXO(IM,JM)                                            INP  219
      LAST = LGAM + IZM
      ITEMP = 1 + ML*(3+IGM*ITL) + IABS(MTP)  + MT*ITL
```

```
      LAST=MAXO(LAST,ITEMP)
      WRITE (NOUT,316) LAST
315   IF (LAST.GT.50000) CALL ERRO2 ('   INP',315,1)
 316  FORMAT(7HOLAST = ,I7)
      DO 317 I=1,ITEMP
317   A(I)=0.0
C     READ CROSS SECTIONS AND WRITE CROSS SECTION TAPE              INP  237
      CALL S860(A(LNO),A(LCO),ITL,IGM,MT,ML,A(LATW),A(LHOLN),A(LALAM))
      DO 325  I=LCO, LAST                                              WLP

325   A(I) = .0                                                     INP  240
C     READ FLUXES AND WRITE FLUX TAPE                               INP  241
      CALL S862(A(LNO), A(LRO), A(LZO))                             INP  242
C     READ EXTERNAL SOURCE                                          INP  243
      IF (IO4) 328,326,328                                          INP  244
326   CALL S864 (A(LS2))
328   CONTINUE                                                      INP  246
      WRITE(NOUT,330)                                               INP  247
330   FORMAT(51HOMESH BOUNDARIES (RO/ZO=RADIAL POINTS/AXIAL POINTS))  INP  248
C     READ RADIAL INTERVALS                                         INP  249
      CALL REAG2('    RO',A(LRO),IP)                                HVX
C     READ AXIAL INTERVALS                                          INP  251
      CALL REAG2('    ZO',A(LZO),JP)                                HVX
C     READ ZONE NUMBERS                                             INP  253
      IF (NPRT .GT. 1) GO TO 335
      CALL REARI2 ('   INP',A(LMO),IMJM)                            HRA2
      GO TO 345
335   WRITE(NOUT,340)                                               INP  254
340   FORMAT(30HOZONE NUMBERS BY MESH INTERVAL)                     INP  255
      CALL REARI2 ('    MO',A(LMO),IMJM)                            HRA2
C     READ MATERIAL NUMBERS                                         INP  257
345   WRITE(NOUT,350)                                               INP  258
350   FORMAT(25HOMATERIAL NUMBERS BY ZONE)                          INP  259
      CALL REARI2('    M2',A(LM2),IZM)                              HRA2
      IF(IO4-5) 351,352,351                                         INP  261
351   IF(BUCK) 352,358,352                                          INP  262
352   WRITE(NOUT,354)                                               INP  263
354   FORMAT(30HOBUCKLING COEFFICIENTS BY ZONE)                     INP  264
```

```
      CALL REAG2('   GAM', A(LGAM),IZM)                                  HVX
358   CONTINUE                                                           INP 266
C     READ FISSION FRACTIONS                                             INP 267
      WRITE(NOUT,360)                                                    INP 268
360   FORMAT(17H0FISSION SPECTRUM)                                       INP 269
      CALL REAG2('    K7',A(LK7),IGM)                                    HVX
C     READ VELOCITIES                                                    INP 271
      IF (I04 .EQ. 2 .OR. S02 .EQ. 2) GO TO 365
      GO TO 375
365   WRITE(NOUT,370)                                                    INP 272
370   FORMAT(17H0NEUTRON VELOCITY)                                       INP 273
      CALL REAG2('    V7',A(LV7),IGM)                                    HVX
375   IF(M01) 400, 400, 380                                              INP 275
380   WRITE(NOUT,390)                                                    INP 276
390   FORMAT ('0MIXTURE SPECIFICATIONS (I0/I1/I2=MIX NUMBER/MAT. NUMBER
     1FOR MIX/MAT. DENSITY)')
      CALL REARI2('    I0',A(LI0),M01)                                   HRA2
      CALL REARI2('    I1',A(LI1),M01)                                   HRA2
      CALL REAG2('    I2',A(LI2),M01)                                    HVX
400   CONTINUE                                                           INP 282
C     CHECK FOR DELTA CALCULATION                                        INP 283
      IF(I04 - 4)  440, 410, 440                                         INP 284
410   WRITE(NOUT,420)                                                    INP 285
420   FORMAT (1H0,'DELTA OPTION DATA (R2/R3=RADIAL ZONE NUMBERS/ZONE MOD
     1IFIERS)')
      CALL REARI2('    R2',A(LR2),IM)                                    HRA2
      CALL REAG2('    R3',A(LR3),IZ)                                     HVX
      WRITE(NOUT,430)                                                    HRA2
430   FORMAT (1H0,'DELTA OPTION DATA (Z2/Z3=AXIAL ZONE NUMBERS/ZONE MODI
     1FIERS)')
      CALL REARI2('    Z2',A(LZ2),JM)                                    HRA2
      CALL REAG2('    Z3',A(LZ3),JZ)                                     HVX
440   CONTINUE                                                           INP 292
      IF(NACT.GT.0)THEN                                                  HRA3
      WRITE(NOUT,490)                                                    HRA3
490   FORMAT (1H0,'ACTIVITY POSITION DATA')                              HRA3
      CALL REARI2(' ACPOS',A(LACPOS),NACT)                               HRA3
      WRITE(NOUT,*)                                                      HRA3
```

```
      ENDIF                                                             HRA3
      CALL MAPR(A(LMO),A(LM2),IM,JM,A(LCO))                             INP  300
      RETURN
      END                                                               INP  302
      SUBROUTINE INPB(MATN,NBR,LD,LCN,LFN,ALAM,HOLN,JML,I2)             INPB   2
       INCLUDE 'ABC.FOR'

      DIMENSION  MATN(1), NBR(1), LD(1),LCN(JML,1),LFN(JML,1), ALAM(1), INPB   4
     1 HOLN(JML,1), I2(1)
      DIMENSION  IDUM(12)
C     THIS SUBROUTINE READS AND PRINTS THE BURNUP DATA                  INPB   6
      CALL REAI2 ('  INPB',ITEMP,1)
      DELT = .0
      IF (ITEMP .NE. 0) CALL REAG2 ('  INPB',DELT,1)
      DAY = DAY + DELT                                                  INPB   9
      CVT = 0                                                           INPB  10
      CNT = 0                                                           INPB  11
      PO2 = 0                                                           INPB  12
      ALA = 0.0                                                         INPB  13
      LAP = 0.0                                                         INPB  14
      LAPP = 0.0                                                        INPB  15
      LAR = 0.0                                                         INPB  16
      KPAGE = 100                                                       INPB  17
      IF(ITEMP)  100, 15, 20                                             WLP
15    NCON = ITEMP                                                      INPB  19
      GO TO 100                                                         INPB  24
20    NCON = ITEMP                                                      INPB  25
      DO 40  N = 1, NCON                                                INPB  26
      REWIND NSCRAT
      CALL REAI2 ('  INPB',IDUM(1),12)
      WRITE (NSCRAT) (IDUM(I),I=1,12)
      REWIND NSCRAT
      READ (NSCRAT)    MATN(N),NBR(N),LD(N),(LCN(N,K),K=1,2),(LFN(N,K), INPB  28
     1 K=1,7)                                                           INPB  29
      REWIND NSCRAT
      ITEMP=MATN(N)
      CALL REAG2 ('  INPB',ALAM(ITEMP),1)
40    CONTINUE
```

```
      WRITE(NOUT,60)                                                    INPB  30
60    FORMAT(12H1BURNUP DATA///)                                        INPB  31
      WRITE(NOUT,70)                                                    INPB  32
70    FORMAT(1H ,'BURNABLE   MAT.    NAME      LAMBDA     NBR  * * *'
     1,' SOURCE ISOTOPE FOR * * * * *'/
     2 1H ,' ISOTOPE   NO.              (DAYS-1)        DECAY',
     1 2X, 'CAPTURE          FISSION'/)
      DO 90  N=1, NCON                                                  INPB  39
80    FORMAT (I6,5X,I3,5X,2A4,1X,E9.2,2X,I3,2X,I3,2X,2I3,2X,7I3)
      ITEMP = MATN(N)                                                   INPB  42
      WRITE(NOUT,80)   N, MATN(N), (HOLN(ITEMP,K),K=1,2), ALAM(ITEMP),
     1 NBR(N), LD(N),   (LCN(N,K),K=1,2), (LFN(N,K),K=1,7)

90    ALAM(ITEMP) = ALAM(ITEMP)/(3600.*24.)                             INPB  46
100   CONTINUE                                                           WLP
      RETURN
      END                                                               INPB  48
      SUBROUTINE MAPR (M0,M2,JIM,JJM,K)                                 MAR    1
      INCLUDE 'ABC.FOR'
      DIMENSION M0(JIM,JJM), M2(1), K(1)                                MAR    3
      DIMENSION FMT1(3), FMT2(3), FMT3(3), PARK(2), PICT(2), MARK(2)    MAR    4
      CHARACTER*6 FMT1, FMT2, FMT3, MARK, PARK, PICT, MARKI             HVX
      DATA FMT1/'(5X,I3',',2H- ,',')     '/                             HVX
      DATA FMT2/'(10X, ','      ',')     '/                             HVX
      DATA FMT3/'(10X, ','      ',')     '/                             HVX
      DATA MARK/' ''','  '''/                                           HVX
      DATA PARK/'60A2 )','40A3 )'/                                      HVX
      DATA PICT/'60I2 )','40I3 )'/                                      HVX
C     PRODUCE A PICTURE PRINT BY ZONE AND MATERIAL                      MAR   11
      NI=0                                                              MAR   22
      DO 100 KZM=1,IZM                                                  MAR   23
100   NI=MAX0(NI,M2(KZM)/100)                                           MAR   24
      DO 110 I=1,IM                                                     MAR   25
      DO 110 J=1,JM                                                     MAR   26
110   NI=MAX0(NI,M0(I,J)/100)                                           MAR   27
      NI=NI+1                                                           MAR   28
      IF (NI.GT.2) NI=2                                                 MAR   29
      FMT1(3)=PICT(NI)                                                  MAR   30
```

```
      FMT2(3)=PICT(NI)
      FMT3(3)=PARK(NI)
      MARKI=MARK(NI)
      NN=1
      MM=60
      IF (NI.EQ.2) MM=40
120   IF (MM.GT.IM) MM=IM
      WRITE (NOUT,190) (ID(I),I=1,20)
      DO 130 JJ=1,JM
      J=JM-JJ+1
130   WRITE (NOUT,FMT1) J,(MO(I,J),I=NN,MM)
      WRITE (NOUT,FMT3) (MARKI,I=NN,MM)
      WRITE (NOUT,FMT2) (I,I=NN,MM)
      WRITE (NOUT,200)
      IF (MM.EQ.IM) GO TO 140
      NN=MM+1
      MM=MM+NN-1
      GO TO 120
140   CONTINUE
      NN=1
      MM=60
      IF (NI.EQ.2) MM=40
150   IF (MM.GT.IM) MM=IM
      WRITE (NOUT,190) (ID(I),I=1,20)
      DO 170 JJ=1,JM
      J=JM-JJ+1
      DO 160 L=NN,MM
      N=MO(L,J)
160   K(L)=IABS(M2(N))
170   WRITE (NOUT,FMT1) J,(K(L),L=NN,MM)
      WRITE (NOUT,FMT3) (MARKI,I=NN,MM)
      WRITE (NOUT,FMT2) (I,I=NN,MM)
      WRITE (NOUT,200)
      IF (MM.EQ.IM) GO TO 180
      NN=MM+1
      MM=MM+NN-1
      GO TO 150
180   RETURN
```

```
190   FORMAT (1H1,20A4///)                                                MAR  70
200   FORMAT (2H A/2H X/2H I/2H A/2H L//8H  RADIAL)                       MAR  71
      END                                                                 MAR  72-
      SUBROUTINE MARCH(PHIB,MATN,FXS,AXS,VOL,MASS,MASSP,ALAM,LD,LCN,      MARC   2
     1                LFN,JML,I0,I1,I2,M2)                                MARC   3
      DIMENSION  PHIB(1), MATN(1), FXS(JML,1),AXS(JML,1),VOL(1),          MARC   4
     1          MASS(JML,1),MASSP(JML,1),ALAM(1), LD(1), LCN(JML,1),      MARC   5
     2          LFN(JML,1),I0(1), I1(1), I2(1), M2(1)                     MARC   6
       INCLUDE 'ABC.FOR'

C     THIS SUBROUTINE COMPUTES THE TIME DEPENDENT ISOTOPIC CONCENTRATIONMARC   8
      TEMP = DELT * 24. * 3600. / 10.                                     MARC   9
      TEMP1 = .0                                                          MARC  10
      DO 5 KZ = 1,IZM                                                     MARC  11
      PHIB(KZ) = PHIB(KZ) * 10.**(-24)                                    MARC  12
      DO 5 KN = 1,NCON                                                    MARC  13
      LN = MATN(KN)                                                       MARC  14
5     TEMP1 = TEMP1 + FXS(KN,KZ)*PHIB(KZ)*MASS(LN,KZ)*VOL(KZ)             MARC  15
      DO 200 KT = 1,10                                                    MARC  16
      TEMP3 = .0                                                          MARC  17
      DO 20 KZ = 1,IZM                                                    MARC  18
      DO 20 KN = 1,NCON                                                   MARC  19
      LN = MATN(KN)                                                       MARC  20
20    MASSP(LN,KZ) = MASS(LN,KZ)                                          MARC  21
      DO 100 KZ = 1,IZM                                                   MARC  22
      DO 50 KKK = 1,5                                                     MARC  23
      DO 50 KN = 1,NCON                                                   MARC  24
      LN = MATN(KN)                                                       MARC  25
CCC  WARNING WARNING                                                      HRA2
      TEMP2=-(MASS(LN,KZ)+MASSP(LN,KZ))*(ALAM(LN)+AXS(KN,KZ)*PHIB(KZ))    MARC  26
CC   ONLY USE ABOVE WHEN SIGMA ABS DOES INCLUDE SIGMA FIS                 HRA2
C      TEMP2=-(MASS(LN,KZ)+MASSP(LN,KZ))*(ALAM(LN)+                        HRA2
C     1        (AXS(KN,KZ)+FXS(KN,KZ))*PHIB(KZ))                           HRA2  2
6
      IF (LD(KN)) 30, 30, 28                                              MARC  27
28    KK = LD(KN)                                                         MARC  28
      KK = MATN(KK)                                                       MARC  29
      TEMP2 = TEMP2 + ALAM(KK)*(MASS(KK,KZ) + MASSP(KK,KZ))               MARC  30
```

```
30    DO 32 K = 1,2                                                    MARC  31
      KK = LCN(KN,K)                                                   MARC  32
      KL = MATN(KK)                                                    MARC  33
      IF (KK) 32,32,31                                                 MARC  34
CCC  WARNING WARNING                                                   HRA2
31    TEMP2 = TEMP2 + (AXS(KK,KZ) - FXS(KK,KZ))*PHIB(KZ)*              MARC  35
CC   ONLY USE ABOVE WHEN SIGMA ABS DOES INCLUDE SIGMA FIS              HRA2
C31    TEMP2 = TEMP2 + (AXS(KK,KZ) )*PHIB(KZ)*                          HRA2  3
5
     1 (MASS(KL,KZ) + MASSP(KL,KZ))                                    MARC  36
32    CONTINUE                                                         MARC  37
      DO 36 K = 1,7                                                    MARC  38
      KK = LFN(KN,K)                                                   MARC  39
      KL = MATN(KK)                                                    MARC  40
      IF (KK) 36,36,34                                                 MARC  41
CCC  THE BELOW IS FOR YIELDS OF FISSION PRODUCT POISONS                HRA2
C34            YIELD = 1.0
C              IF(KL.EQ.1)THEN
C                  IF(LN.EQ.7)YIELD=0.061
C                  IF(LN.EQ.8)YIELD=0.003
C                  IF(LN.EQ.9)YIELD=0.0113
C                  IF(LN.EQ.11)YIELD=1.0
C              ENDIF
C              IF(KL.EQ.6)THEN
C                  IF(LN.EQ.7)YIELD=0.055
C                  IF(LN.EQ.8)YIELD=0.000
C                  IF(LN.EQ.9)YIELD=0.019
C                  IF(LN.EQ.11)YIELD=1.0
C              ENDIF
C34    TEMP2=TEMP2+YIELD+FXS(KK,KZ)*PHIB(KZ)*(MASS(KL,KZ)+MASSP(KL,KZ))  HRA2  4
2
CCC   THE ABOVE IS FOR YIELDS OF FISSION PRODUCT POISONS
34    TEMP2 = TEMP2 + FXS(KK,KZ)*PHIB(KZ)*(MASS(KL,KZ)+MASSP(KL,KZ))   MARC  42
36    CONTINUE                                                         MARC  43
50    MASS(LN,KZ) = MASSP(LN,KZ) + .5*TEMP*TEMP2                       MARC  44
      DO 100 KN = 1,NCON                                               MARC  45
      LN = MATN(KN)                                                    MARC  46
100   TEMP3 = TEMP3 + FXS(KN,KZ)*PHIB(KZ)*MASS(LN,KZ)*VOL(KZ)          MARC  47
```

```
      IF(TEMP3)  200,200,110                                            MARC  48
110   DO 120 KZ = 1,IZM                                                 MARC  49
120   PHIB(KZ) = PHIB(KZ) * TEMP1/TEMP3                                 MARC  50
200   CONTINUE                                                          MARC  51
      DO 500 KZ = 1,IZM                                                 MARC  52
500   PHIB(KZ) = PHIB(KZ)*10.**(24)                                     MARC  53
      DO 540  KZ=1,IZM                                                  MARC  54
      DO 540  M=1,M01                                                   MARC  55
      IF(IO(M) - M2(KZ))  540,520,540                                   MARC  56
520   DO 530  KN=1,NCON                                                 MARC  57
      LN = MATN(KN)                                                     MARC  58
      IF(LN - I1(M))  530,525,530                                       MARC  59
525   I2(M) = MASS(LN,KZ)                                               MARC  60
530   CONTINUE                                                          MARC  61
540   CONTINUE                                                          MARC  62
      RETURN                                                            MARC  63
      END                                                               MARC  64
      SUBROUTINE REAG2(HOLL,ARRAY,NCOUNT)                               REAG2001
CCC   REPLACE ALL FORMAT STATEMENTS WITH * ON A VAX                     HRA2
      COMMON NSORCE,NINP, NOUT
      DIMENSION ARRAY (1), HOL(80),HE(40),LE(40)
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      J=0
1     READ (NINP, 10) (HOL(I),I=1,80)
10    FORMAT (80A1)
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF(I.LE.80) GO TO 50
      L=L-1
      GO TO 100
50    IF (HOL(I).EQ.' ') GO TO 60
      IF (HOL(I).EQ.'T') GO TO 55
```

```
      IF (HOL(I).EQ.'C') GO TO 55
      IF (HOL(I).EQ.'F') GO TO 55
      IF (HOL(I).EQ.'R') GO TO 55
      IF (HOL(I).EQ.'I') GO TO 55
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
55    IF (LE(L).GT.0) L=L+1
      LE(L)=1
      HE(L)(1:1)= HOL(I)
      IF (HOL(I).EQ.'R' .OR. HOL(I).EQ. 'I') GO TO 30
      GO TO 100
60    IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
100   LL=L
      L=0
110   L=L+1
      IF (L .GT. LL) GO TO 1
      IF (HE(L)(1:1).EQ.'T') GO TO 150
      IF (HE(L)(1:1).EQ.'C') GO TO 115
      IF (HE(L)(1:1).EQ. 'F') GO TO 120
      IF (HE(L)(1:1).EQ. 'R') GO TO 130
      IF (HE(L)(1:1).EQ.'I') GO TO 140
      J=J+1
      READ (HE(L)(1:LE(L)),112) ARRAY(J)
C     DECODE (LE(L),112,HE(L))  ARRAY(J)
112   FORMAT (E20.2)
      GO TO 110
C     CYCLE
115   READ (HE(L+1)(1:LE(L+1)),132) J1
C115  DECODE (LE(L+1),132,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),132) J2
C     DECODE (LE(L+2),132,HE(L+2)) J2
      J0 = J
      DO 119 K1=1,J1
      DO 119 K2=1,J2
      J=J+1
119   ARRAY(J)=ARRAY(J0 - J2 + K2)
```

```
      L=L+2
      GO TO 110
C     FILL
120   DO 125 JJ=J+1,NCOUNT
125   ARRAY(JJ)=ARRAY(J)
      J=NCOUNT
      GO TO 150
C     REPEAT
130   READ (HE(L+1)(1:LE(L+1)),132) J1
C130  DECODE (LE(L+1),132,HE(L+1)) J1
132   FORMAT (I20)
      READ (HE(L+2)(1:LE(L+2)),112) T1
C     DECODE (LE(L+2),112,HE(L+2)) T1
      DO 135 JJ=J+1,J+J1
135   ARRAY(JJ)=T1
      J=J+J1
      L=L+2
      GO TO 110
C     INTERPOLATE
140   READ (HE(L+1)(1:LE(L+1)),132) J1
C140  DECODE (LE(L+1),132,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),112) ARRAY(J+J1+1)
C     DECODE (LE(L+2),112,HE(L+2)) ARRAY(J+J1+1)
      T1= (ARRAY(J+J1+1) - ARRAY(J))/(J1+1)
      DO 145 JJ=J+1,J+J1
145   ARRAY(JJ)= ARRAY(JJ-1) + T1
      J=J +J1+1
      L=L+2
      GO TO 110
150   IF (HOLL .EQ. '   INP') GO TO 155
      IF (HOLL .EQ. '  INPB') GO TO 155
      IF (HOLL .EQ. '  S860') GO TO 155
      WRITE (NOUT,160) HOLL,J,(ARRAY(I),I=1,J)
155   IF (J-NCOUNT) 170,180,170
160   FORMAT (6X,A6,I6/(6E12.5))
170   CALL ERRO2 ( ' REAG2',170,1)
180   RETURN
      END
```

```
      SUBROUTINE RREAG2(HOLL,A,NCOUNT)                                  REAG2001
CCC   REPLACE ALL FORMAT STATEMENTS WITH * ON A VAX                     HRA2
      COMMON NSORCE,NINP, NOUT
      DIMENSION ARRAY (1), HOL(80),HE(40),LE(40)
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      J=0
1     READ (NINP, 10) (HOL(I),I=1,80)
10    FORMAT (80A1)
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF(I.LE.80) GO TO 50
      L=L-1
      GO TO 100
50    IF (HOL(I).EQ.' ') GO TO 60
      IF (HOL(I).EQ.'T') GO TO 55
      IF (HOL(I).EQ.'C') GO TO 55
      IF (HOL(I).EQ.'F') GO TO 55
      IF (HOL(I).EQ.'R') GO TO 55
      IF (HOL(I).EQ.'I') GO TO 55
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
55    IF (LE(L).GT.0) L=L+1
      LE(L)=1
      HE(L)(1:1)= HOL(I)
      IF (HOL(I).EQ.'R' .OR. HOL(I).EQ. 'I') GO TO 30
      GO TO 100
60    IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
100   LL=L
      L=0
110   L=L+1
```

```
      IF (L .GT. LL) GO TO 1
      IF (HE(L)(1:1).EQ.'T') GO TO 150
      IF (HE(L)(1:1).EQ.'C') GO TO 115
      IF (HE(L)(1:1).EQ. 'F') GO TO 120
      IF (HE(L)(1:1).EQ. 'R') GO TO 130
      IF (HE(L)(1:1).EQ.'I') GO TO 140
      J=J+1
      READ (HE(L)(1:LE(L)),112) A
C     DECODE (LE(L),112,HE(L))  A
112   FORMAT (E20.2)
      GO TO 110
C     CYCLE
115   READ (HE(L+1)(1:LE(L+1)),132) J1
C115  DECODE (LE(L+1),132,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),132) J2
C     DECODE (LE(L+2),132,HE(L+2)) J2
      J0 = J
      DO 119 K1=1,J1
      DO 119 K2=1,J2
      J=J+1
119   ARRAY(J)=ARRAY(J0 - J2 + K2)
      L=L+2
      GO TO 110
C     FILL
120   DO 125 JJ=J+1,NCOUNT
125   ARRAY(JJ)=ARRAY(J)
      J=NCOUNT
      GO TO 150
C     REPEAT
130   READ (HE(L+1)(1:LE(L+1)),132) J1
C130  DECODE (LE(L+1),132,HE(L+1)) J1
132   FORMAT (I20)
      READ (HE(L+2)(1:LE(L+2)),112) T1
C     DECODE (LE(L+2),112,HE(L+2)) T1
      DO 135 JJ=J+1,J+J1
135   ARRAY(JJ)=T1
      J=J+J1
      L=L+2
```

```
      GO TO 110
C     INTERPOLATE
140   READ (HE(L+1)(1:LE(L+1)),132) J1
C140  DECODE (LE(L+1),132,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),112) ARRAY(J+J1+1)
C     DECODE (LE(L+2),112,HE(L+2)) ARRAY(J+J1+1)
      T1= (ARRAY(J+J1+1) - ARRAY(J))/(J1+1)
      DO 145 JJ=J+1,J+J1
145   ARRAY(JJ)= ARRAY(JJ-1) + T1
      J=J +J1+1
      L=L+2
      GO TO 110
150   IF (HOLL .EQ. '   INP') GO TO 155
      IF (HOLL .EQ. '  INPB') GO TO 155
      IF (HOLL .EQ. '  S860') GO TO 155
      WRITE (NOUT,160) HOLL,J,(ARRAY(I),I=1,J)
155   IF (J-NCOUNT) 170,180,170
160   FORMAT (6X,A6,I6/(6E12.5))
170   CALL ERRO2 ( ' REAG2',170,1)
180   RETURN
      END
      SUBROUTINE REAI2(HOLL,IARRAY,NCOUNT)                          REAI2001
      COMMON NSORCE, NINP, NOUT
      DIMENSION IARRAY (1), HOL(80),HE(40),LE(40)
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      J=0
1     READ (NINP, 10) (HOL(I),I=1,80)
10    FORMAT (80A1)
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF (HOL(I) .EQ. '/') GO TO 45
      IF(I.LE.80) GO TO 50
```

```
45    L=L-1
      GO TO 100
50    IF (HOL(I).EQ.' ') GO TO 60
      IF (HOL(I).EQ.'T') GO TO 55
      IF (HOL(I).EQ.'F') GO TO 55
      IF (HOL(I).EQ.'R') GO TO 55
      IF (HOL(I).EQ.'I') GO TO 55
      IF (HOL(I) .EQ. 'C') GO TO 55
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
55    IF (LE(L).GT.0) L=L+1
      LE(L)=1
      HE(L)(1:1)= HOL(I)
      IF (HOL(I) .EQ. 'C') GO TO 30
      IF (HOL(I).EQ.'R' .OR. HOL(I).EQ. 'I') GO TO 30
      GO TO 100
60    IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
100   LL=L
      L=0
110   L=L+1
      IF (L .GT. LL) GO TO 1
      IF (HE(L)(1:1).EQ.'T') GO TO 150
      IF (HE(L)(1:1).EQ. 'C') GO TO 115
      IF (HE(L)(1:1).EQ. 'F') GO TO 120
      IF (HE(L)(1:1).EQ. 'R') GO TO 130
      IF (HE(L)(1:1).EQ.'I') GO TO 140
      J=J+1
      READ (HE(L)(1:LE(L)),112) IARRAY(J)
C     DECODE (LE(L),112,HE(L))  IARRAY(J)
112   FORMAT (I20)
      GO TO 110
C     CYCLE
115   READ (HE(L+1)(1:LE(L+1)),112) J1
C115  DECODE (LE(L+1),112,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),112) J2
C     DECODE (LE(L+2),112,HE(L+2)) J2
```

```
      J0 = J
      DO 119 K1=1,J1
      DO 119 K2=1,J2
      J=J+1
119   IARRAY(J)=IARRAY(J0 - J2 + K2)
      L=L+2
      GO TO 110
C     FILL
120   DO 125 JJ=J+1,NCOUNT
125   IARRAY(JJ)=IARRAY(J)
      J=NCOUNT
      GO TO 150
C     REPEAT
130   READ (HE(L+1)(1:LE(L+1)),112) J1
C130  DECODE (LE(L+1),112,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),112) I1
C     DECODE (LE(L+2),112,HE(L+2)) I1
      DO 135 JJ=J+1,J+J1
135   IARRAY(JJ)=I1
      J=J+J1
      L=L+2
      GO TO 110
C     INTERPOLATE
140   READ (HE(L+1)(1:LE(L+1)),112) J1
C140  DECODE (LE(L+1),112,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),112) IARRAY(J+J1+1)
C     DECODE (LE(L+2),112,HE(L+2)) IARRAY(J+J1+1)
      I1= (IARRAY(J+J1+1) - IARRAY(J))/(J1+1)
      DO 145 JJ=J+1,J+J1
145   IARRAY(JJ)= IARRAY(JJ-1) + I1
      J=J +J1+1
      L=L+2
      GO TO 110
150   IF (HOLL.EQ.'   INP') GO TO 155
      IF (HOLL .EQ. '  INPB') GO TO 155
      IF (HOLL .EQ. '  S860') GO TO 155
      WRITE (NOUT,160) HOLL,J,(IARRAY(I),I=1,J)
155   IF (J-NCOUNT) 170,180,170
```

```
160   FORMAT (6X,A6,I6/(10I6))
170   CALL ERRO2 (' REAI2',170,1)
180   RETURN
      END
      SUBROUTINE REARI2(HOLL,MIARRAY,NCOUNT)                          REAI2001
      COMMON NSORCE, NINP, NOUT
      DIMENSION MIARRAY (1), HOL(80),HE(40),LE(40)
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      J=0
1     READ (NINP, 10) (HOL(I),I=1,80)
10    FORMAT (80A1)
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF (HOL(I) .EQ. '/') GO TO 45
      IF(I.LE.80) GO TO 50
45    L=L-1
      GO TO 100
50    IF (HOL(I).EQ.' ') GO TO 60
      IF (HOL(I).EQ.'T') GO TO 55
      IF (HOL(I).EQ.'F') GO TO 55
      IF (HOL(I).EQ.'R') GO TO 55
      IF (HOL(I).EQ.'I') GO TO 55
      IF (HOL(I) .EQ. 'C') GO TO 55
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
55    IF (LE(L).GT.0) L=L+1
      LE(L)=1
      HE(L)(1:1)= HOL(I)
      IF (HOL(I) .EQ. 'C') GO TO 30
      IF (HOL(I).EQ.'R' .OR. HOL(I).EQ. 'I') GO TO 30
      GO TO 100
```

```
60    IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
100   LL=L
      L=0
110   L=L+1
      IF (L .GT. LL) GO TO 1
      IF (HE(L)(1:1).EQ.'T') GO TO 150
      IF (HE(L)(1:1).EQ. 'C') GO TO 115
      IF (HE(L)(1:1).EQ. 'F') GO TO 120
      IF (HE(L)(1:1).EQ. 'R') GO TO 130
      IF (HE(L)(1:1).EQ.'I') GO TO 140
      J=J+1
      READ (HE(L)(1:LE(L)),112) MIARRAY(J)
C     DECODE (LE(L),112,HE(L))  RIARRAY(J)
112   FORMAT (I20)
      GO TO 110
C     CYCLE
115   READ (HE(L+1)(1:LE(L+1)),112) J1
C115  DECODE (LE(L+1),112,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),112) J2
C     DECODE (LE(L+2),112,HE(L+2)) J2
      J0 = J
      DO 119 K1=1,J1
      DO 119 K2=1,J2
      J=J+1
119   MIARRAY(J)=MIARRAY(J0 - J2 + K2)
      L=L+2
      GO TO 110
C     FILL
120   DO 125 JJ=J+1,NCOUNT
125   MIARRAY(JJ)=MIARRAY(J)
      J=NCOUNT
      GO TO 150
C     REPEAT
130   READ (HE(L+1)(1:LE(L+1)),112) J1
C130  DECODE (LE(L+1),112,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),112) I1
C     DECODE (LE(L+2),112,HE(L+2)) I1
```

```
      DO 135 JJ=J+1,J+J1
135   MIARRAY(JJ)=I1
      J=J+J1
      L=L+2
      GO TO 110
C     INTERPOLATE
140   READ (HE(L+1)(1:LE(L+1)),112) J1
C140  DECODE (LE(L+1),112,HE(L+1)) J1
      READ (HE(L+2)(1:LE(L+2)),112) MIARRAY(J+J1+1)
C     DECODE (LE(L+2),112,HE(L+2)) MIARRAY(J+J1+1)
      I1= (MIARRAY(J+J1+1) - MIARRAY(J))/(J1+1)
      DO 145 JJ=J+1,J+J1
145   MIARRAY(JJ)= MIARRAY(JJ-1) + I1
      J=J +J1+1
      L=L+2
      GO TO 110
150   IF (HOLL.EQ.'   INP') GO TO 155
      IF (HOLL .EQ. '  INPB') GO TO 155
      IF (HOLL .EQ. '  S860') GO TO 155
      WRITE (NOUT,160) HOLL,J,(MIARRAY(I),I=1,J)
155   IF (J-NCOUNT) 170,180,170
160   FORMAT (6X,A6,I6/(10I6))
170   CALL ERRO2 (' REAI2',170,1)
180   RETURN
      END
      SUBROUTINE S860 (C,CO,JTL,JGM,JMT,JML,ATW,HOLN,ALAM)
       INCLUDE 'ABC.FOR'

      DIMENSION C(JTL,JGM,JMT), CO(JTL,JMT), ATW(1), HOLN(JML,1),ALAM(1)
      DIMENSION XSD(50,50),XSR(50,50),XD(50),XR(50)                     HRA2
C     THIS SUBROUTINE READS CROSS SECTIONS, PERFORMS ADJOINT            S860   6
C     REVERSALS IF REQUIRED, AND WRITES CROSS SECTION TAPE              S860   7
      WRITE(NOUT,5)  (ID(I), I=1,20)                                    S860   8
5     FORMAT(1H1,20A4 ///)                                              S860   9
10    WRITE (NOUT, 20 )                                                 S860  11
20    FORMAT (55H CROSS SECTIONS ARE READ-IN FOR THE FOLLOWING MATERIALSS860  12
     1/)                                                                S860  13
      DO 50 I=1,ML
```

```
      IF (MTP) 40,40,30
30    READ (15) (HOLN(I,K),K=1,2), ATW(I)
      READ(15)  ((C(L,IIG,I), L=1,ITL), IIG=1,IGM)                    S860  19
      GO TO 48
40    READ (NINP,42) (HOLN(I,K),K=1,2)
42    FORMAT (2A4)
      CALL REAG2 ('  S860', ATW(I), 1)
      DO 45  IIG=1,IGM                                                S860  21
      CALL REAG2 ('  S860',C(1,IIG,I), ITL)
45    CONTINUE
48    WRITE (NOUT,55) I, (HOLN(I,K),K=1,2)
50    CONTINUE
55    FORMAT (I3, 6X, 2A4)
C     CHECK ON CROSS SECTION CONSISTENCY AND ORDER                    S860  26
      IF(MCR)  70,70,90                                               S860  27
70    REWIND 15                                                       S860  28
90    CONTINUE                                                        S860  29
      DO 140 J=1,ML                                                   S860  30
      DO 140  I=1,IGM                                                 S860  31
      G = C(2,I,J) + C(5,I,J)                                         S860  32
      DO 110  K = 1, NXCM                                             S860  33
      KK = I + K                                                      S860  34
      M = 5 + K                                                       S860  35
      IF(KK - IGM)  100, 100, 110                                     S860  36
100   G = G + C(M,KK,J)                                               S860  37
110   CONTINUE                                                        S860  38
      XSR(I,J)=G-C(IHS,I,J)                                           HRA2
      XSD(I,J)=G-C(IHT-2,I,J)-C(IHS,I,J)                              HRA2
      IF(C(4,I,J).EQ. 0.0) GO TO 130
      G=ABS((G-C(4,I,J))/C(4,I,J))                                    S860  39
      IF(G-.0001) 140,130,130                                         S860  40
130   WRITE(NOUT,135) J,I,G                                           S860  41
135   FORMAT (' CHECK MATERIAL ',I2, 5X, ' GROUP ',I2,G10.4)
140   CONTINUE                                                        S860  43
C                                                                     S860  44
C     A02=0/1=FLUX CALCULATION/ADJOINT CALCULATION                    S860  45
160   IF(A02)  170, 280, 170                                          S860  46
170   DO 190  IIG=1,IGM                                               S860  47
```

```
      IGBAR=IGM-IIG+1                                              S860  48
      DO 180 M=1,ML
      DO 180  L = IHT-3,IHS                                        HRA2  50
      TEMP=C(L,IIG,M)                                              S860  51
      C(L,IIG,M)=C(L,IGBAR,M)                                      S860  52
180   C(L,IGBAR,M)=TEMP                                            S860  53
      IF (IGBAR - IIG -1) 200, 200, 190                            S860  54
190   CONTINUE                                                     S860  55
200   CONTINUE                                                     S860  56
      KK = ITL - IHS                                               S860  57
      IF (KK) 280, 280, 210                                        S860  58
210   CONTINUE                                                     S860  59
      DO 240 M=1,ML
      DO 240  IIG = 1,IGM                                          S860  61
      IGBAR = IGM - IIG + 1                                        S860  62
      DO 240  L = 1,KK                                             S860  63
      IF (L - IIG) 220, 240, 240                                   S860  64
220   I = L + IHS                                                  S860  65
      ITEMP = IGBAR + L                                            S860  66
      IF (IIG - ITEMP) 230, 230, 240                               S860  67
230   TEMP = C(I, IIG, M)                                          S860  68
      C(I,IIG,M) = C(I,ITEMP,M)                                    S860  69
      C(I,ITEMP,M) = TEMP                                          S860  70
240   CONTINUE                                                     S860  71
C     WRITE CROSS SECTION TAPE                                     S860  72
280   DO 300  IIG=1,IGM                                            S860  73
      DO 295 M=1,MT
      IF(M .LE. ML) GO TO 288
      DO 284 L=1,ITL
  284 CO(L,M)=0.0
      GO TO 295
  288 CONTINUE
      DO 290  L=1,ITL                                              S860  75
290   CO(L,M)=C(L,IIG,M)                                           S860  76
      XR(M)=XSR(IIG,M)                                             HRA2
      XD(M)=XSD(IIG,M)                                             HRA2
  295 CONTINUE
      WRITE (11) (XR(M),M=1,MT)                                    HRA2
```

```
      WRITE (12) (XD(M),M=1,MT)                                 HRA2
      DO 88 M=1,MT                                              HRA2
88    CONTINUE                                                  HRA2
300   WRITE (NCR1)    ((CO(L,M),L=1,ITL),M=1,MT)                S860  77
      REWIND NCR1                                               S860  78
      REWIND 11                                                 HRA2
      REWIND 12                                                 HRA2
      RETURN                                                    S860  79
      END                                                       S860  80
      SUBROUTINE S862(NO,RF,ZF)                                 S862   2
       INCLUDE 'ABC.FOR'

      DIMENSION  NO(1), RF(1), ZF(1)                            S862   4
C     THIS SUBROUTINE READS THE INPUT FLUXES AND PREPARES A FLUX TAPE   S862   5
      WRITE(NOUT,5)                                             S862   6
5     FORMAT(1H1)                                               S862   7
C     M07=0/1=NO FLUX INPUT/FLUX FROM TAPE 14
      DO 1000  IIG = 1, IGM                                     S862  14
      IF (M07 .GT. 0) GO TO 120
      DO 59 I=1,IM
      DO 59 J=1,JM
      ITEMP= (J-1)*IM + I
59    NO(ITEMP) = 1.0
      GO TO 200
120   READ(14) (NO(I), I=1, IMJM)                               S862  34
200   WRITE(NFLUX1)  (NO(I), I=1, IMJM)                         S862  35
1000  CONTINUE                                                  S862  36
      REWIND 14
      REWIND NFLUX1
      RETURN                                                    S862  40
      END                                                       S862  41
      SUBROUTINE S864 (S2)
       INCLUDE 'ABC.FOR'

      DIMENSION S2(1)
C     THIS SUBROUTINE READS THE EXTERNAL SOURCE AND PREPARES A SOURCE TAS864   6
      DO 50 IIG = 1,IGM                                         S864  32
      CALL REAG2 ('    SO',S2,IMJM)                             HVX
```

```
      WRITE (NSORCE) (S2(I), I = 1,IMJM)                              S864  59
50    CONTINUE                                                        S864  60
      REWIND NSORCE                                                   S864  61
      RETURN                                                          S864  62
      END                                                             S864  63
      SUBROUTINE S8830                                                S883   2
       INCLUDE 'ABC.FOR'
      INTEGER*2 IT1,IT2,IT3,IT4
C     MONITOR PRINT
C    CHANGE GETTIM TO CLOCK(IT1,IT2,IT3) ON A VAX                     HRA2
      CALL GETTIM (IT1,IT2,IT3,IT4)
      TI = FLOAT (3600*IT1 + 60*IT2 + IT3)/60.
      KPAGE = KPAGE + 1                                               S883   7
      IF(KPAGE - 40)  220, 160, 160                                   S883   8
160   KPAGE = 0                                                       S883   9
210   WRITE(NOUT, 213)                                                S883  10
213   FORMAT (1H1,'   TIME       OUTER      IN. IT.      EIGENVALUE
     1    EIGENVALUE      LAMBDA')
      WRITE(NOUT, 215)                                                S883  13
215   FORMAT ('  (MINUTES)  ITERATIONS   PER LOOP      SLOPE'/)
220   WRITE(NOUT, 225)    TI, P02, LC, EQ,  EV, ALA                   S883  16
225   FORMAT (2X, F6.2, 7X, I4, 7X, I4, 6X, 3(E14.7))
230   P02=P02+1                                                       S883  18
      LC=0                                                            S883  19
      IF(P02-D05)430,430,330                                          S883  20
330   NGOTO = 1                                                       S883  21
      GO TO 630                                                       S883  22
430   NGOTO = 4                                                       S883  23
630   RETURN                                                          S883  24
      END                                                             S883  25
      SUBROUTINE S8847                                                S884   2
       INCLUDE 'ABC.FOR'

C     THIS SUBROUTINE PRINTS THE FINAL NEUTRON BALANCE TABLE
      E2(IGP) = .0                                                    S884   4
      E3(IGP) = .0                                                    S884   5
      E4(IGP) = .0                                                    S884   6
      E5(IGP) = .0                                                    S884   7
```

```
      E6(IGP) = .0                                                      S884   8
      E7(IGP) = .0                                                      S884   9
      E8(IGP) = .0                                                      S884  10
      E9(IGP) = .0                                                      S884  11
      DO 10 I = 1,IGM                                                   S884  12
      E2(IGP) = E2(IGP) + E2(I)                                         S884  13
      E3(IGP) = E3(IGP) + E3(I)                                         S884  14
      E4(IGP) = E4(IGP) + E4(I)                                         S884  15
      E5(IGP) = E5(IGP) + E5(I)                                         S884  16
      E6(IGP) = E6(IGP) + E6(I)                                         S884  17
      E7(IGP) = E7(IGP) + E7(I)                                         S884  18
      E8(IGP) = E8(IGP) + E8(I)                                         S884  19
   10 E9(IGP) = E9(IGP) + E9(I)                                         S884  20
      WRITE(NOUT,20)                                                    S884  21
   20 FORMAT (1H1, 28H FINAL NEUTRON BALANCE TABLE///                   S884  22
     1 ' GROUP  FISSION  IN-SCAT  OUT-SCAT  ABSORB '
     2,'  L.L.    R.L.    T.L.    B.L. '//)
      DO 30 I = 1,IGM                                                   S884  26
   25 FORMAT (I4,3X, 1P8E9.2)
   30 WRITE(NOUT,25) I,E1(I),E2(I),E3(I),E4(I),E5(I),E6(I),E7(I),       S884  28
     1 E8(I)
      WRITE(NOUT,35)                                                    S884  30
   35 FORMAT (1H )                                                      S884  31
      I = IGM + 1                                                       S884  32
      WRITE(NOUT,25) I,E1(I),E2(I),E3(I),E4(I),E5(I),E6(I),E7(I),       S884  33
     1 E8(I)
      XK=E1(I)/(E4(I)+E9(I))
      WRITE(NOUT,70) XK
   70 FORMAT (1H0/5X, 'NEUTRON MULTIPLICATION CONSTANT = ',F10.6)
      RETURN                                                            S884  35
      END                                                               S884  36
      SUBROUTINE S8850(F2,N2,R1,Z1,R4,Z4,V7,JIM,JJM,FN2,                S885   2
     1  CO,NO,MO,M2,FO,JTL,JMT)
        INCLUDE 'ABC.FOR'

      DIMENSION F2(JIM,JJM), N2(JIM,JJM), R1(1), Z1(1), R4(1), Z4(1),   S885   5
     1          FLUX(6), FN2(1), CO(JTL,JMT), NO(JIM,JJM), MO(JIM,JJM), S885   6
     2          M2(1), FO(JIM,JJM),V7(1)                                S885   7
```

```
C     S8850   FINAL PRINT
      ICARD = 1                                                          S885   9
      CALL S8830                                                         S885  11
      CALL S8847                                                         S885  13
      IF (NPRT-1) 90,90,15
15    J=IP                                                               S885  14
      IF(IP-JP) 30, 30, 20                                               S885  15
20    J=JP                                                               S885  16
30    WRITE (NOUT, 40 ) (I,R1(I),R4(I),Z1(I),Z4(I),I=1,J)                S885  17
40    FORMAT (1H1, 16X, 'RADII', 9X, 'AVG RADII', 11X, 'AXII',
     1 14X, 'AVG AXII'//(I4, 4F18.4))
      J=J+1                                                              S885  20
      IF(IP-JP)  50, 90, 70                                              S885  21
50    WRITE (NOUT, 60 ) (I,Z1(I),Z4(I),I=J,JP)                           S885  22
60    FORMAT(I4,36X,2F18.4)                                              S885  23
      GO TO 90                                                           S885  24
70    WRITE (NOUT, 80 ) (I,R1(I),R4(I),I=J,IP)                           S885  25
80    FORMAT(I4,2F18.4)                                                  S885  26
90    CONTINUE                                                           S885  27
      DO 100   I=1, IM                                                   S885  28
      DO 100   J=1,JM                                                    S885  29
      NO(I,J) = 0.0                                                      S885  30
100   F2(I,J) = 0.0                                                      S885  31
      DO 220   IIG=1, IGM                                                S885  32
      IF (NPRT .GT.2) WRITE (NOUT, 110) IIG
110   FORMAT(1H1, 20X,14HFLUX FOR GROUP,I3)                              S885  34
      READ (NFLUX1)((N2(I,J),I=1,IM),J=1,JM)                             S885  35
      READ(NCR1)((CO(II,J), II = 1, ITL), J = 1, MT)                     S885  36
      DO 120   I=1, IM                                                   S885  37
      DO 120   J=1, JM                                                   S885  38
      NO(I,J) = NO(I,J) + N2(I,J)                                        S885  39
      ITEMP = MO(I,J)                                                    S885  40
      ITEMP = M2(ITEMP)                                                  S885  41
120   F2(I,J) = F2(I,J) + CO(IHT-3,ITEMP)*N2(I,J)*1000.*TSD              HRA2  42
      IF(NPUN)   210, 210, 205                                           S885  43
205   WRITE(16) ((N2(I,J),I=1,IM),J=1,JM)                                S885  63
210   IF (NPRT .GT.2) CALL PRT (IM,JM,N2,Z4,NOUT)
220   CONTINUE                                                           S885  65
```

```
      IF (NPRT .LE. 0) GO TO 250
      WRITE(NOUT, 230 )                                              S885  67
230   FORMAT(1H1//, 19X,11H TOTAL FLUX//)                            S885  68
      CALL PRT(IM,JM,NO,Z4,NOUT)                                     S885  69
      WRITE(NOUT, 240 )                                              S885  71
240   FORMAT(1H1//, 19X, 26HPOWER DENSITY  (MWT/LITER))              S885  72
      CALL PRT(IM,JM,F2,Z4,NOUT)                                     S885  73
250   CONTINUE
      IF(NPUN - 1)  270, 260, 260                                    S885  82
  260 END FILE 16                                                    S885  83
      WRITE(NOUT,265)                                                S885  84
 265  FORMAT(1H0,50X,'***** FLUXES, ETC. DUMPED TO TAPE *****')
270   REWIND NCR1                                                    S885  86
      REWIND NFLUX1                                                  S885  87
      RETURN                                                         S885  88
      END                                                            S885  89
      SUBROUTINE TCHEK(LGH,JUMP)                                     TCHE   2
      INTEGER*2 IT1,IT2,IT3,IT4
CC CHANGE GETTIM TO CLOCK(IT1,IT2,IT3) ON A VAX                      HRA2
      CALL GETTIM (IT1,IT2,IT3,IT4)
      ISEC = 3600*IT1 + 60*IT2 + IT3
      IF(ISEC.GT.60*MAXT.AND.MAXT.GT.0) JUMP=1
      RETURN                                                         TCHE   6
      END                                                            TCHE   7
CCC THIS SUBROUTINE MUST BE USED ON THE VAX
CCC       SUBROUTINE CLOCK (IT1,IT2,IT3)
C       INTEGER*2 IT1,IT2,IT3
C       IT1=0
C       IT2=0
C       IT3=0
C       RETURN
C       END
      SUBROUTINE IFLUXL (N2, CO, VO, CXS, NO, M2, JTL,JIM,JJM, CXR, CXT HRA2   2
     1  ,XR,XD)                                                      HRA2
      INCLUDE 'ABC.FOR'

      DIMENSION  N2(1), CO(JTL,1), VO(1),CXS(JIM,JJM,3),MO(1), M2(1),   IFLU   4
     1          CXR(1), CXT(1)                                         IFLU   5
```

```
      DIMENSION XD(50),XR(50)
C     THIS SUBROUTINE NORMALIZES FLUXES BEFORE EACH INNER ITERATION
C     ABSORPTION AND OUT-SCATTER
      E3(IGV) = 0.0
      E4(IGV) = 0.0
      DO 10  I=1, IMJM
      TEMP = VO(I)*N2(I)
      ITEMP = MO(I)
      ITEMP = M2(ITEMP)
      E3(IGV) = E3(IGV) + (XD(ITEMP))*TEMP
10    E4(IGV) = E4(IGV) + CO(IHT-2,ITEMP)*TEMP
C     LEFT LEAKAGE
      IF(B01)  20, 20, 40
20    E5(IGV) = 0.0
      DO 30  KJ = 1, JM
      I = (KJ - 1)*IM + 1
30    E5(IGV) = E5(IGV) + CXS(1,KJ,1)*N2(I)
      GO TO 50
40    E5(IGV) = .0
C     RIGHT LEAKAGE
50    IF(B02)  60, 60, 80
60    E6(IGV) = 0.0
      DO 70  KJ = 1, JM
      I = KJ*IM
70    E6(IGV) = E6(IGV) + CXR(KJ)*N2(I)
      GO TO 90
80    E6(IGV) = 0.0
C     TOP LEAKAGE
90    IF(B03-1)  120, 140, 100
100   E7(IGV) =.0
      DO 110   KI = 1,IM
      I = IMJM - IM + KI
110   E7(IGV) = E7(IGV) + CXS(KI,1,2)*(N2(I) - N2(KI))
      E8(IGV) = - E7(IGV)
      GO TO 190
120   E7(IGV) = 0.0
      DO 130  KI = 1, IM
      I = IMJM - IM + KI
```

```
TLIST2.FOR           Wednesday, February 21, 1990 12:31 pm            Page 81

130   E7(IGV) = E7(IGV) + CXT(KI)*N2(I)                               IFLU  43
      GO TO 150                                                       IFLU  44
140   E7(IGV) = 0.0                                                   IFLU  45
C     BOTTOM LEAKAGE                                                  IFLU  46
150   IF(B04)  160, 160, 180                                          IFLU  47
160   E8(IGV) = 0.0                                                   IFLU  48
      DO 170 KI = 1, IM                                               IFLU  49
170   E8(IGV) = E8(IGV) + CXS(KI,1,2)*N2(KI)                          IFLU  50
      GO TO 190                                                       IFLU  51
180   E8(IGV) = 0.0                                                   IFLU  52
190   E9(IGV) = E5(IGV) + E6(IGV) + E7(IGV) + E8(IGV)                 IFLU  53
      RETURN                                                          IFLU  64
      END                                                             IFLU  65

      PROGRAM ABC.FOR
      COMMON NSORCE
      COMMON NINP,NOUT,NCR1,NFLUX1,NSCRAT,ALA,B07,
     1CNT,CVT,DAY,DELT,E0(51),E1(51),E2(51),E3(51),E4(51),E5(51),
     2E6(51),E7(51),E8(51),E9(51),
     3E01,E02,E03,EQ,EVP,EVPP,FEF,GBAR,GLH,IGEP,IGP,IGV,IHS,IHT,II,
     4IMJM,IP,ITEMP,ITEMP1,ITEMP2,ITL,IZP,JP,K07,KPAGE,LAP,LAPP,LAR,
     5LC,ML,NCON,NGOTO,NPRT,ORFP
      COMMON P02,PBAR,SBAR,SK7,T06,T7,T11,TEMP,TEMP1,TEMP2,TEMP3,TEMP4,
     7TI,TSD,V11
      COMMON ID(20),MAXT,A02,I04,S02,IGM,NXCM,MCR,MTP,M07,D05,G07,S04,
     1NPUN,IGE,IM,JM,IZM,MT,M01,B01,B02,B03,B04,IZ,JZ,EV,EVM,S03,BUCK,
     2LAL,LAH,EPS,EPSA,G06,POD,ORF,S01,NACT,NFD
      COMMON LATW,LHOLN,LALAM,LCO,LN0,LN2,LA0,LA1,LF0,LF2,LI0,LI1,LI2,
     1LI3,LK6,LK7,LM0,LM2,LR0,LR1,LR2,LR3,LR4,LR5,LS2,LV0,LV7,LZ0,LZ1,
     2LZ2,LZ3,LZ4,LZ5,LCXS,LVOL,LMASS,LMATN,LNBR,LLD,LLCN,LLFN,LPHIB,
C     3LAXS,LFXS,LMASSP,LCXR,LCXT,LHA,LPA,LT6,LT8,lntwon,bigr
     3LAXS,LFXS,LMASSP,LCXR,LCXT,LHA,LPA,lact,lacpos,bigr
      INTEGER A02,B01,B02,B03,B04,B07,CNT,CVT,D05,G07,P02,S02,S04,T06,
     1R2,Z2,GLH
      REAL I2,I3,K6,K7,LAH,LAL,LAP,LAPP,LAR,N0,N2,MASS,MASSP
```