ANL-7406

ANL-7406

TESS, A One-dimensional S_n Transport-theory Code for the CDC-3600

Ronald W. Goin and J. P. Plummer

U of C-AUA-USAEC

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

Prepared for the U.S. ATOMIC ENERGY COMMISSION
under contract W-31-109-Eng-38

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
Iowa State University
The University of Iowa
Kansas State University
The University of Kansas
Loyola University
Marquette University
Michigan State University
The University of Michigan
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame
The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
The University of Texas at Austin
Washington University
Wayne State University
The University of Wisconsin

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.95

ANL-7406
Mathematics and
Computers

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

TESS, A One-dimensional S_n Transport-theory Code for the CDC-3600

by

Ronald W. Goin and J. P. Plummer

Applied Physics Division

December 1971

TABLE OF CONTENTS

	Page
ABSTRACT	7
CODE SUMMARY	7
I. INTRODUCTION	9
II. THEORY	13
A. Derivation of Transport Equation for Slab and Spherical Geometries	13
B. Numerical Approximations Used in TESS Program	15
C. Derivations of Difference Equations	17
1. Slab	17
2. Sphere	17
3. Boundary Conditions	18
D. Method of Solution Used in Each Energy Group	20
E. Energy-group Coupling	22
F. Integrated Balance Formulas	24
G. Integration Formulas	25
H. Cross-section Homogenization	27
I. Buckling	30
III. OUTER-ITERATION AND CONVERGENCE CYCLE	31
A. Source Iterations	31
B. Search Iterations	32
IV. INPUT	34
A. Alphanumeric Card	34
B. Card Format for Fixed-point Data	34
C. Card Format for Floating-point Data	35
D. Optional Cross-section Tape Information	36
E. Optional Reaction-rate Material Names and Cross Sections	37
F. Input Instructions for Fixed-point Data	38

Page

G. Input Instructions for Floating-point Data 41

H. Additional Notes on Input 42

1. Δx Criterion 42
2. Additional Boundary-condition Information 43
3. XLIBIT Output-card Format 44

I. Error Indicators 44

V. PROGRAM INFORMATION 47

A. General Information and List of Subroutines and Overlays . 47

B. Tape Assignments 48

C. Input-deck Order 49

VI. OUTPUT DATA 49

VII. ESTIMATE OF RUNNING TIME 52

APPENDIXES

A. Floating-point Addresses of Cross-section Matrices 53

1. First-word Addresses 53
2. First-word Addresses of STR Scattering Transfer Matrix 53

B. TRIM Angles for Use in TESS 54

C. Choice of Angles for Use in TESS 57

1. Discussion 57
2. Results 58

D. Angular Convergence: TESS versus SNARG 62

E. Listing of TESS 63

F. Sample Problem 142

1. Listing of Sample-problem Input Deck 142
2. Sample-problem Output 143

ACKNOWLEDGMENTS 151

REFERENCES 152

LIST OF FIGURES

No.	Title	Page
1.	Zero Power Plutonium Reactor (ZPPR)	9
2.	Typical ZPPR Matrix Tube, Drawer, and Assorted Constituent Plates	10
3.	Top View of Inner-core Drawer Loading	10
4.	Calculational Model for Transverse Cell	11
5.	Calculational Model for Vertical Cell	11
6.	Iteration Time in TESS vs S_n Angular Order	52
7.	Matrix Computation Time in TESS vs S_n Angular Order	52
C.1.	Calculational Model	57
C.2.	Degree of Convergence vs Angular Order for Various Choices of Source and Nonsource Region Thicknesses	59
D.1.	Unit Cell with Mirror Boundary Conditions	62
D.2.	TESS and SNARG Convergence, as a Function of Angular Order	62

LIST OF TABLES

No.	Title	Page
I.	Maximum Recommended Mesh-interval Size	43
B.1.	Cosines of TRIM Angles	54
C.1.	Cosines Obtained by Method I	58
C.2.	Cosines Obtained by Method II	58
C.3.	Combination Cosines	58

TESS, A One-dimensional S_n Transport-theory Code for the CDC-3600

by

Ronald W. Goin and J. P. Plummer

ABSTRACT

TESS is a code, written in FORTRAN, which provides the multigroup real and adjoint solutions to the transport equation in the S_n approximation for one-dimensional slab and spherical geometry. A direct method of solution for each outer iteration is used for maximum efficiency in slowly converging problems. A double S_n formulation in slab geometry, which allows the code to treat discontinuities more efficiently in the angular flux at interfaces, yields more accurate results with fewer angles. TESS contains a sophisticated cross-section homogenization routine, which permits cross-section collapse in both space and energy by six different prescriptions--three by real flux weighting, and three by flux and adjoint weighting. In addition, two different methods of providing for cell leakage make the code convenient for fast-reactor, critical-facility heterogeneity studies. Highly generalized boundary-condition capabilities make TESS quite suitable for photon-transport problems as well. Integrals of flux and adjoint for perturbation analysis can be calculated. Reaction rates may be computed for specified isotopes as a function of space.

CODE SUMMARY

1. Program title: TESS.

2. Computer for which designed: CDC-3600 with 50K words of available core (64K including the monitoring system) and 10 tapes (not including the system tape).

   Other computers upon which it is operable: With minor conversion effort, any computer with sufficient core and auxiliary storage (since the code is written in FORTRAN).

3. Nature of physical problem solved: Solves transport equation in slab and spherical geometry using double-S_n approximation for slab.

4. Method of solution: Uses direct solution for angular fluxes for each outer iteration.

5. Restrictions on the complexity of the problem: 150 mesh points, 26 groups, 20 angular intervals, 12 isotropic downscatter groups, one P_1 downscatter group, 40 regions, 25 materials, no restriction on angles except that they be symmetric about $\pi/2$ ($\mu = 0$). No upscatter is allowed.

6. Typical running time: Approximately 5×10^{-4} sec/(point-group-angular order squared) per iteration. This is a very rough estimate.

7. Unusual features of the program: Automatically computes both flux and adjoint and performs integrals necessary for perturbation analysis. Does bilinear as well as real flux-weighted cross-section homogenization over both space and energy. Computes reaction rates as a function of space for specified isotopes.

8. Related and auxiliary programs: TESS is an extension and modification of the MIST code. One of the main modifications in TESS is the conversion of the coefficient matrices from three-dimensional to one-dimensional vectors to increase both the maximum permissible problem size and the speed of the code. Other modifications are the adjoint calculation, the lifetime calculation, the calculation of flux and adjoint integrals for perturbation analysis, the cross-section homogenization routine, and the addition of the spherical solution.

9. Status: Operating.

10. References: IDO-16856, MIST (Multigroup Internuclear Slab Transport), by G. E. Putnam and D. M. Shapiro (May 10, 1963).

11. Machine requirements: 50K available core storage, 10 tape units, including I/O but not including monitor.

12. Programming language used: 3600 FORTRAN, 100%.

13. Operating system under which program is executed: SCOPE 6.2114.

14. Other programming or operating information or restrictions: The "Buffer In" and "Buffer Out" statements are used and would have to be changed to be compatible with FORTRAN IV. The time remaining at any

stage of the computation is determined by an A = TIMELEFT(A) statement; the program is made up of five overlays, one of which contains three segments.

15. Material available: Source decks, writeup, listings, and overlay tape.

I. INTRODUCTION

One problem associated with the analysis of plate-type fast reactors, such as ZPR-3, -6, -9, and ZPPR,[1] is to determine the effect of the heterogeneity introduced by using plates of various materials to represent materials in a power reactor. This problem arises when, for example, plates of graphite, canned sodium, plutonium metal, and depleted uranium, each perhaps 1/8 or 1/4 in. thick, are stacked together in stainless steel drawers to simulate a carbide-fueled, sodium-cooled power reactor. Another example is the use of sodium carbonate, iron oxide, depleted uranium oxide, and plutonium metal plates to simulate an oxide-fueled power reactor.

A picture of ZPPR is shown in Fig. 1. The reactor consists of one fixed and one movable half, each an array of square matrix tubes into which are inserted drawers containing the materials to be used in the reactor. Figure 2 shows loaded drawers as used in ZPPR Assembly 2. Figure 3 is a top view of the core drawer loading. For a detailed description of ZPPR, see Ref. 1.

Fig. 1. Zero Power Plutonium Reactor (ZPPR). ANL Neg. No. 103-A11302.

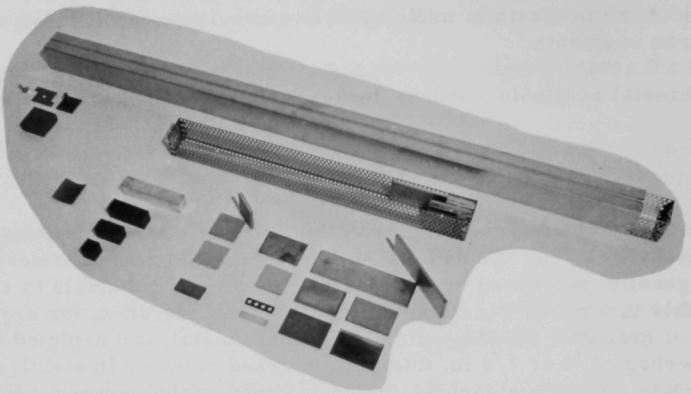


Fig. 2. Typical ZPPR Matrix Tube, Drawer, and Assorted Constituent Plates. ANL Neg. No. 103-A11524.

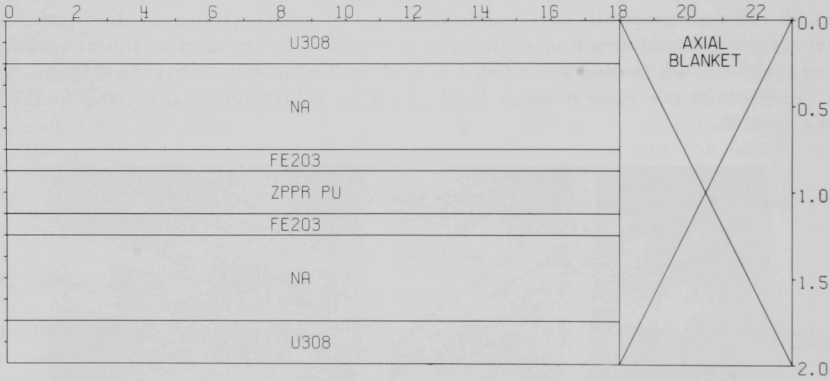


Fig. 3. Top View of Inner-core Drawer Loading

In thermal-reactor physics calculations, it has been standard procedure to determine the flux depression through a fuel pin or plate in the thermal-energy group by use of transport-theory cell calculations, and apply flux and volume weighting to either the atomic densities or the macroscopic cross sections to obtain homogenized thermal-group cross sections. A more complex situation occurs in fast reactors because the flux in a particular material zone may peak at some energies and be depressed at other energies. The spatial shape of the adjoint is dependent upon the composition of the various material zones. In the analysis of reactors such as these, the effect of the heterogeneities on the spectrum

(both the real spectrum and the adjoint spectrum) is needed. This might suggest that the individual material macroscopic cross sections for each energy group should be weighted bilinearly (both flux and adjoint) for homogenization. The question of whether to use bilinear weighting, or flux weighting only, probably depends on what quantities one is trying to calculate. For most purposes, either scheme should be quite adequate.

The procedure that might be followed for an assembly with drawer loadings as shown in Fig. 3 follows: Begin with a cell calculation in the transverse direction across a drawer cell. The calculational model is shown in Fig. 4. Both flux and adjoint calculations would be done with this model; then the individual material cross sections would be homogenized by either flux or bilinear weighting.

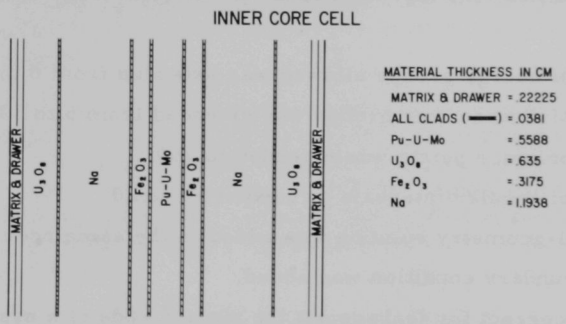


Fig. 4

Calculational Model for Transverse Cell

The resulting homogenized macroscopic cross sections would be used in a vertical cell calculation to account for the drawer and matrix-tube top and bottom. The model for this calculation is shown in Fig. 5.

The cross sections of the two materials in the vertical cell would then be flux or bilinearly weighted to give homogenized macroscopic cross sections. These homogenized cross sections would be used in a cylindrical or spherical core calculation to determine spatially dependent flux and adjoint spectra, reaction rates, prompt-neutron lifetime, etc.

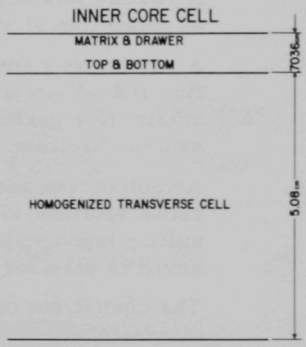


Fig. 5. Calculational Model for Vertical Cell

The transverse and vertical cells shown in Figs. 4 and 5 are the unit cells in the periodic structure of the inner core zone of ZPPR, Assembly 2. The TESS cell calculations could be done with the cells represented exactly as shown, using the periodic-boundary-condition option. In practice, however, use would be

made of the fact that both horizontal and vertical cells have mirror planes of symmetry. Thus, half of each cell would be represented in the input, and the mirror-boundary-condition option (MFR = 0 and the appropriate number of elements in the ALPHA array set to 1.0) would be selected.

To assist in the analysis of these plate-type systems, the TESS code has been programmed in FORTRAN for the CDC-3600 with 64K storage. TESS is based upon a direct method of solution (one requiring no "inner iterations") of the double S_n formulation in slab geometry developed by Putnam and programmed as the MIST code.[2] Because it is a direct solution, this formulation converges in about the same number of iterations as diffusion-theory calculations.

The main modifications that were made to the MIST code that resulted in TESS are:

The number of energy groups allowed was extended from 6 to 26.

The number of downscatter groups was extended from 5 to 12.

The number of space points was extended to 150.

The number of angular intervals was extended to 20.

The spherical-geometry solution was added in the same code.

A periodic boundary condition was added.

Two ways to correct for leakage out the top and ends of a system (corresponding to DB^2 in diffusion theory) were added.

An option was added to compute both flux and adjoint for a particular case automatically.

A routine was added to compute reaction rates of specified isotopes as a function of space.

A routine was added to compute integrals of the product of angular flux and adjoint and integrals of the product of scalar flux and adjoint (for perturbation analysis), and to compute the prompt-neutron lifetime.

A routine was added to flux- and adjoint-weight the macroscopic cross sections to homogenize over both space and energy. The resulting homogenized cross sections can be punched on cards and/or saved in core for succeeding problems.

The coefficient matrix was changed from three-dimensional to one-dimensional vectors for more efficient handling.

The main advantages of TESS over existing S_n codes for ZPR- and ZPPR-type reactor analyses are:

(1) TESS has maximum change-case flexibility. In one pass on the computer, one can, for example, do the following: (a) Obtain flux- and adjoint-weighted cross sections across the plates, assuming the plates are infinitely long and high, then correct for matrix tubes and drawers by a second calculation. (If desired, a calculation along the length of the plates could be done to account for end effects.) (b) Follow the same procedure to obtain homogenized cross sections in the blanket drawer cells and, if desired, the core drawer cells at the core edge. (c) Perform a calculation describing the entire reactor in either slab or spherical geometry using the homogenized cross sections obtained in the preceding cases to obtain k_{eff}, fluxes and adjoints, prompt-neutron lifetime, etc.

(2) The flux and adjoint solutions can be obtained in one pass. In most existing S_n codes, two separate runs are required to obtain the flux and adjoint solutions, and, if both are wanted, some card or tape handling is necessary.

(3) A double S_n approximation is used in slab geometry, and the angular points can be completely arbitrary as long as they are symmetric about $\mu = 0$. For example, in a slab cell calculation, where the gradient of the angular flux is greatest at $\pi/2$, the angles can be concentrated about $\pi/2$ to result in the most accurate description of the angular distribution with a minimum of angles. See Appendix C for recommendations on the choice of angles.

(4) Computer time is saved. In most cases, TESS is at least as fast as other S_n codes, and for problems in which a large fraction of neutron collisions result in nondegrading scatter, where transport effects are important, TESS can be considerably faster. See Appendix D for a comparison of the rate of convergence as a function of angular order for TESS and SNARG.

II. THEORY

A. Derivation of Transport Equation for Slab and Spherical Geometries

The Boltzmann transport equation balancing leakage, collisions, scatterings, and fixed sources in the energy independent case is[3]

$$\operatorname{div} \overline{N}(\overline{r},\overline{\Omega}) + \Sigma(\overline{r},\overline{\Omega})N(\overline{r},\overline{\Omega}) = \int_{\Omega'} N(\overline{r},\overline{\Omega}')\Sigma_s(\overline{r},\overline{\Omega},\overline{\Omega}')\, d\overline{\Omega}' + q(\overline{r},\overline{\Omega}), \qquad (1)$$

where $\overline{N}(\overline{r},\overline{\Omega}) = N(\overline{r},\overline{\Omega})\overline{\Omega}$ is the vector flux defined as the number of particles traveling in a direction $\overline{\Omega}$ crossing a unit area normal to $\overline{\Omega}$ per unit time per unit solid angle, $\Sigma(\overline{r},\overline{\Omega})$ is the total cross section for removal of neutrons from $d\overline{\Omega}$ about $\overline{\Omega}$, and $\Sigma_s(\overline{r},\overline{\Omega},\overline{\Omega}')$ is the scattering cross section for scattering of neutrons in $d\overline{\Omega}'$ about Ω' into $d\overline{\Omega}$ about $\overline{\Omega}$. The $q(\overline{r},\overline{\Omega})$ represents any source of particles per unit volume per unit solid angle per unit

time emitted in $d\bar{r}$ about $\bar{r}$ and $d\bar{\Omega}$ about $\bar{\Omega}$. Consider the restriction of Eq. 1 to one-dimensional slab or spherical geometry, where the flux and source are functions only of a position coordinate x and some angle θ between a unit vector in the positive x direction and $\bar{\Omega}$. In this case, $N(r,\bar{\Omega}) \rightarrow N(x,\mu)$ and $q(r,\bar{\Omega}) \rightarrow q(x,\mu)$, where $\mu = \cos\theta = (\bar{x} \cdot \bar{\Omega})/x$. The first term in Eq. 1 can be reduced by a well-known identity,

$$\begin{aligned} \operatorname{div}[N(x,\bar{\Omega})\bar{\Omega}] &= \operatorname{grad} N(x,\bar{\Omega}) \cdot \bar{\Omega} + N(x,\bar{\Omega}) \operatorname{div} \bar{\Omega} \\ &= \operatorname{grad} N(x,\bar{\Omega}) \cdot \bar{\Omega}. \end{aligned}$$

First consider spherical coordinates:

$$\begin{aligned} \operatorname{div}[N(\bar{x},\bar{\Omega})\bar{\Omega}] &= \frac{\partial N(x,\mu)}{\partial x}\mu + \frac{\partial N(x,\mu)}{\partial \mu}\bar{\Omega} \cdot \operatorname{grad} \mu \\ &= \frac{\partial N(x,\mu)}{\partial x}\mu + \frac{1}{x}\frac{\partial N(x,\mu)}{\partial \mu}(\eta^2 + \xi^2), \end{aligned}$$

where η and ξ, in addition to μ, are the direction cosines of $\bar{\Omega}$ in our spherical coordinates, thus satisfying $\eta^2 + \xi^2 + \mu^2 = 1$. Therefore,

$$\operatorname{div}[N(\bar{x},\bar{\Omega})\bar{\Omega}] = \frac{\partial N(x,\mu)}{\partial x}\mu + \frac{(1-\mu^2)}{x}\frac{\partial N(x,\mu)}{\partial \mu}. \qquad \underline{\text{spherical geometry}}$$

In slab geometry, since μ is defined with respect to a fixed direction in space, grad $\mu = 0$ and the result is

$$\operatorname{div}[N(\bar{x},\bar{\Omega})\bar{\Omega}] = \mu\frac{\partial N(x,\mu)}{\partial x}. \qquad \underline{\text{slab geometry}}$$

Setting $d\Omega = d\varphi\, d\mu$ in Eq. 1 leads to the result

$$\mu\frac{\partial N(x,\mu)}{\partial x} + \delta\left[\frac{(1-\mu^2)}{x}\frac{\partial N(x,\mu)}{\partial x}\right] + \Sigma(x)N(x,\mu) =$$

$$\int_0^{2\pi}\int_{-1}^{1} N(x,\mu')\Sigma_s(x,\Omega,\Omega')\, d\mu'\, d\varphi + q(x,\mu), \tag{2}$$

where

$$\begin{aligned} \delta &= 0 \text{ in slab geometry} \\ &= 1 \text{ in spherical geometry.} \end{aligned}$$

Now it is assumed that the scattering cross section is a function only of the angle θ_0 between $\bar{\Omega}$ and $\bar{\Omega}'$, so that if $\mu_0 = \cos\theta_0$, then

$$\Sigma_s(x,\mu_0) = \Sigma_s(x,\bar{\Omega},\bar{\Omega}').$$

Also, the representation of $\Sigma_s(x,\mu_0)$ in spherical harmonics is

$$\Sigma_s(x,\mu_0) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \Sigma_{s\ell}(x) P_\ell(\mu_0), \tag{3}$$

where

$$\Sigma_{s\ell}(x) = \int_0^{2\pi} \int_{-1}^{1} \Sigma_s(x,\mu_0) P_\ell(\mu_0)\, d\mu_0\, d\varphi;$$

and if one uses the addition theorem for spherical harmonics,

$$P_\ell(\mu_0) = P_\ell(\mu) P_\ell(\mu') + 2 \sum_{m=1}^{\ell} \frac{(\ell-m)!}{(\ell+m)!} P_\ell^m(\mu') \cos[m(\varphi-\varphi')] \tag{4}$$

(where the P_ℓ^m's are the associated Legendre polynomials).

These relationships can be used directly to reduce the basic Boltzmann equation to

$$\mu \frac{\partial N(x,\mu)}{\partial x} + \delta\left[\frac{1-\mu^2}{x} \frac{\partial N(x,\mu)}{\partial \mu}\right] + \Sigma(x) N(x,\mu) =$$

$$\sum_{\ell=0}^{\infty} \frac{2\ell+1}{2} \Sigma_{s\ell}(x) P_\ell(\mu) \int_{-1}^{1} P_\ell(\mu') N(x,\mu')\, d\mu' + q(x,\mu). \tag{5}$$

The terms involving m in Eq. 4 vanish with integration over φ'. Equation 5 is the basic equation used for each neutron group in the TESS program.

B. Numerical Approximations Used in TESS Program

In the TESS program the sum on the right-hand side of Eq. 5 is restricted to $\ell = 1$, and the $q(x,\mu)$ is assumed to be isotropic and therefore independent of μ. The only other approximation made is described as follows: $N(x,\mu)$ is assumed to be linear with respect to x and μ in each interval, which results from the independent partitioning of the x and the μ spaces. This approximation is simply that in the x interval (x_i, x_{i+1}) and the μ interval (μ_j, μ_{j+1}), $N(x,\mu)$ is represented by the expression

$$N(x,\mu) = N(x_i\mu_j) + \frac{\partial N(x_i\mu_j)}{\partial x}(x-x_i) + \frac{\partial N(x_i\mu_j)}{\partial \mu}(\mu-\mu_j) + \frac{\partial^2 N(x_i\mu_j)}{\partial x \partial \mu}(x-x_i)(\mu-\mu_j), \tag{6}$$

where

$$\frac{\partial N(x_i, \mu_j)}{\partial x} = \frac{N(x_{i+1}, \mu_j) - N(x_i, \mu_j)}{x_{i+1} - x_i},$$

$$\frac{\partial N}{\partial \mu}(x_i, \mu_j) = \frac{N(x_i, \mu_{j+1}) - N(x_i, \mu_j)}{\mu_{j+1} - \mu_j},$$

and

$$\frac{\partial^2 N(x_i, \mu_j)}{\partial x \partial \mu} = \frac{N(x_{i+1}, \mu_{j+1}) - N(x_{i+1}, \mu_j) - N(x_i, \mu_{j+1}) + N(x_i \mu_j)}{(x_{i+1} - x_i)(\mu_{j+1} - \mu_j)}.$$

Each half of the μ space is partitioned into $n/2$ intervals, if n is the angular order. Thus there are $n + 1$ discrete values of μ in spherical geometry and $n + 2$ values in slab geometry, to allow the two values of $\mu = 0$ that enable TESS to treat discontinuities in the angular flux at slab interfaces. Furthermore, the code assumes that the negative and positive halves of μ space are partitioned symmetrically. The code also assumes that the μ's are ordered from -1 to +1; i.e., $\mu_1 = -1$. If we define $J \equiv n + 2$ in slab geometry, and $J \equiv n + 1$ in spherical geometry, then the symmetry requirement on the μ_j is simply $\mu_j = -\mu_{J+1-j}$.

The partitioning of the range of x proceeds by regions inasmuch as each region of a problem is partitioned into a number of equal intervals (one or more). The word "region" here and elsewhere in the program description always refers to some defined portion of the range of x in which both the width of the x interval and all the material properties (cross sections, etc.) are constant.

This partitioning of the range of x defines the mesh points x_i, and in the TESS code:

$$x_1 \geq 0$$

and

$$x_1 < x_i < x_{MAX} \text{ for } 1 < i < MAX,$$

where x_{MAX} is the upper bound of the range of x, and MAX is the number of mesh points.

The TESS program solves for all the values of $N(x_i, \mu_j)$ for $1 \leq i \leq MAX$ and $1 \leq j \leq J$.

C. Derivations of Difference Equations

The basic balance equation (Eq. 5) is used to derive the necessary difference equations by making use of the numerical approximations and the partitioning descriptions set forth in Sect. II.B above.

With the approximation (Eq. 6), Eq. 5 can be multiplied by $4\pi x^2$ (in spherical geometry) and integrated in an interval (x_i, x_{i+1}) with respect to x, and the scattering integral on the right from $\mu = -1$ to $+1$ can be represented in terms of values of $N(x_{i+1}, \mu_j)$ and $N(x_i, \mu_j)$ with all values of j occurring. Without writing the resulting equation, we shall term it Eq. 5'. Equation 5' can then be used to derive directly the final difference equations from which are obtained:

Sphere	Slab
$N(x_i, \mu_j)$ for $1 \le j \le \frac{J+1}{2}$	$N(x_i, \mu_j)$ for $1 \le j \le J/2$
and	and
$N(x_{i+1}, \mu_j)$ for $\left(\frac{J+1}{2} + 1\right) \le j \le J$.	$N(x_{i+1}, \mu_j)$ for $(J/2+1) \le j \le J$.

1. Slab

The equations for $N(x_i, \mu_1)$ and $N(x_{i+1}, \mu_J)$ are determined by simply setting μ equal to -1 and +1, respectively, in Eq. 5'.

The $J/2 - 1$ equations for $N(x_{i+1}, \mu_j)$ for $J/2 + 1 \le j \le J$ are determined by integrating Eq. 5' in the corresponding interval (μ_j, μ_{j+1}).

Similarly, the $J/2 - 1$ equations for $N(x_i, \mu_j)$ for $1 < j \le J/2$ are determined by integrating Eq. 5' in the corresponding interval (μ_{j-1}, μ_j).

In this manner, $J \times (MAX - 1)$ difference equations are obtained from Eq. 5' and the remaining J equations necessary for solution for all $N(x_i, \mu_j)$ are obtained from equations that state the boundary conditions.

2. Sphere

The equation for $N(x_i, \mu_1)$ is determined by setting μ equal to -1 in Eq. 5'.

The remaining $J - 1$ equations for $N(x_i, \mu_j)$ for $1 < j \le (J+1)/2$ and $N(x_{i+1}, \mu_j)$ for $(J+1)/2 + 1 \le j \le J$ are determined by integrating Eq. 5' in the corresponding interval (μ_{j-1}, μ_j).

In this manner, $J \times (MAX - 1)$ difference equations are obtained from Eq. 5' and the remaining J equations necessary for solution for all $N(x_i, \mu_j)$ are obtained from equations that state the boundary conditions.

Note that the adjoint solution will not in general be identical to the real solution in spherical geometry since the adjoint solution starts with $\mu = +1$ and proceeds in the direction opposite that of the real solution. As the width of the angular interval approaches zero, the two solutions approach the same value. This difference between the real and adjoint solutions can serve as an indication of the adequacy of the angular description.

3. Boundary Conditions

The boundary condition at either boundary for each neutron group is expressed by the equation

$$\mu N_g(x_s, \mu) = \mu \Bigg[A_{s,g} N(x_s, -\mu)(1 - \alpha) + 2B_{s,g} \int_{\Delta\mu_s} N_g(x_s, \mu')\mu' \, d\mu'$$

$$+ \sum_{\ell=0}^{9} S_{\ell,s,g} P_\ell(\mu) + N'_g(x_s, \mu) + \alpha A_{s,g} N(x_{s'}, -\mu) \Bigg]. \qquad (7)$$

In this equation,

α = 1 for periodic boundary condition, 0 otherwise,

g = group number,

s = 1 for a left boundary, = 2 for a right boundary,

and

s' = 2 for a left boundary, = 1 for a right boundary.

The variable μ is understood to be the cosine of an angle in the angular halfspace that is directed into the boundary surface. Therefore, for $N_g(x_1, \mu)$, $0 \le \mu \le 1$; and for $N_g(x_2, \mu)$, $-1 \le \mu \le 0$. The symbol $\Delta\mu_s$ denotes the μ halfspace for fluxes emerging from the surface. The physical meaning of each term on the right-hand side of Eq. 7 is as follows:

a. If $A_{s,g}$ is some number in the interval (0, 1), then the fraction $A_{s,g}$ of any flux emerging from the boundary surface will be reflected back into the surface as if reflected by a perfect mirror. This $A_{s,g}$ term in the equation is, for example, used in specifying a perfect symmetry condition at a boundary by setting $A_{s,g} = 1$. The number $A_{s,g}$ is termed the mirror albedo coefficient.

b. If $B_{s,g}$ is some number in the interval (0, 1), then the fraction $B_{s,g}$ of any flux emerging from the boundary surface will be reflected back into the surface isotropically. For example, the current reentering the boundary surface will be exactly $B_{s,g}$ times the current leaving the surface, but the angular-flux distribution reentering the surface is isotropic. In the radiative transport field, such a boundary condition describes a Lambert surface. The number $B_{s,g}$ is termed the isotropic albedo coefficient.

c. The term

$$\sum_{\ell=0}^{9} S_{\ell,s,g} P_\ell(\mu)$$

is simply a Legendre-polynomial representation of an axially symmetric (no φ dependence) source at boundary s for group g. TESS is dimensioned to handle no more than a ninth-degree Legendre polynomial. For convenient reference, if $S_{\ell,s,g} = 0$ for $1 \leq \ell$, an isotropic entrant flux $N(x, \mu) = S_{0,s,g}$ will result, and the entrant current in this case will be

$$\int_0^{2\pi} d\varphi \int_{\Delta\mu_s} S_{0,s,g}\mu \, d\mu = \pi S_{0,s,g}.$$

The user should be reminded that if an anisotropic source is specified (i.e., $S_{\ell,s,g} \neq 0$ for some $\ell \geq 1$), care should be taken to ensure that the Legendre-polynomial coefficients for odd values of ℓ are of the correct sign. The source as expressed by the Legendre polynomial defines a flux distribution for both halfspaces of μ, and the flux distribution of an anisotropic source containing nonzero coefficients for odd ℓ values is not the same in both μ halfspaces; i.e., it is not symmetric about $\mu = 0$. The user should check to make sure that his polynomial produces the desired flux distribution in the halfspace $\underline{\mu = 0 \text{ to } 1}$ when the source is applied at the <u>left boundary</u> and that the polynomial produces the desired flux distribution in the halfspace $\underline{\mu = -1 \text{ to } 0}$ when the source is applied at the <u>right boundary</u>.

d. The term $N_g^{\prime}(x_s, \mu)$ in the boundary condition Eq. 7 allows one to apply a known entrant-flux distribution as a source at a boundary. The TESS program interprets such an input flux distribution in the same manner as it interprets those fluxes it manufactures; that is, the flux distribution $N_g^{\prime}(x_s, \mu)$ will be assumed linear in each interval of the μ halfspace. (These intervals are defined, of course, by the μ space partitioning in which each halfspace is divided into n/2 intervals.) The user specifies a boundary condition of this kind by defining exactly [(J + 1)/2, sphere; or J/2, slab] point values of the desired flux distribution at each boundary for

each group. In this way, output fluxes from a TESS solution can be directly input to another TESS problem if both solutions have the same number of angular intervals.

The necessary additional J difference equations that must be derived from the boundary conditions (for each group) are derived from Eq. 7 in exactly the same manner as were difference equations derived from Eq. 5'. Specifically, in spherical geometry at a right boundary ($s = 2$), the equation for $N_g(x_{MAX}, \mu_1)$ is derived simply by setting $\mu = -1$. Difference equations for $N(x_{MAX}, \mu_j)$ for $1 < j \leq (J+1)/2$ are derived by integrating Eq. 7 with respect to μ in the interval (μ_{j-1}, μ_j). Difference equations are derived from Eq. 7 for a left boundary by integrating with respect to μ in the interval (μ_{j-1}, μ_j) for $(J+1)/2 + 1 \leq j \leq J$. Before the difference equations at the boundaries are derived from Eq. 7, the integral over $\Delta\mu_s$ that appears on the right-hand side is expressed in terms of $N_g(x_s, \mu_k)$, where k represents the indices for the fluxes emerging from the surface.

In slab geometry, at a left boundary ($s = 1$), the equation for $N_g(x_1, \mu_j)$ is derived simply by setting $\mu = +1$. Difference equations for $N(x_1, \mu_j)$ for $J/2 < j < J$ are derived by integrating Eq. 7 with respect to μ in the interval (μ_j, μ_{j+1}). Difference equations are derived from Eq. 7 for a right boundary in a strictly analogous manner, except that one equation is obtained by setting $\mu = -1$ and the others are obtained by integrating with respect to μ in the interval (μ_{j-1}, μ_j). Before the difference equations at the boundaries are derived from Eq. 7, the integral over $\Delta\mu_s$ that appears on the right-hand side is expressed in terms of $N_g(x_s, \mu_k)$ where k represents the indices for the fluxes emerging from the surface.

D. Method of Solution Used in Each Energy Group

In Sect. II.C above, the difference equations derived for each energy group from the basic balance equation and the boundary conditions were described. These difference equations are $J \times MAX$ in number, for the solutions $N(x_i, \mu_j)$ must be obtained at every point x_i of the x space and at every point μ_j of the μ space. As previously defined, the maximum i is MAX and the maximum j is J, hence the need for $J \times MAX$ equations.

If the terms obtained from $q(x, \mu)$ in Eq. 5 (when the difference equations are derived) remain on the right-hand side and if the terms obtained from the scattering integral are taken to the left-hand side, the entire set of difference equations can be conveniently represented in matrix form as $\underline{M}\underline{N} = \underline{Q}$, where $\underline{M}$ is a $J \times MAX$ by $J \times MAX$ matrix of coefficients, $\underline{N}$ is a flux vector with $J \times MAX$ components, and $\underline{Q}$ is a source vector with $J \times MAX$ components.

The first J rows of $\underline{M}$ are the coefficients of the difference equations derived for $x_i = x_1$ and $j = J$ through 1, respectively. The next set of J rows corresponds to x_2, etc.

The solution to this set of equations is accomplished in three steps.

1. By simple Gaussian elimination, the subdiagonal elements of $\underline{M}$ are reduced to zero in the first column, then the second, the third, etc. This is most conveniently accomplished in the following manner: Let $\underline{L}_1'$ be a unit $J \times MAX$ by $J \times MAX$ matrix, except for nonzero elements in the first column, and let $\underline{L}_1'$ be such that $\underline{L}_1'\underline{M} = \underline{M}_1'$, where $\underline{M}_1'$ has zero subdiagonal elements in the first column. Let $\underline{L}_2'$ be a unit $J \times MAX$ by $J \times MAX$ matrix, except for nonzero elements in the second column, and let $\underline{L}_2'$ be such that $\underline{L}_2'\underline{M}_1' = \underline{M}_2'$, where $\underline{M}_2'$ has zero subdiagonal elements in the first and second columns. In a similar manner, one can define $\underline{L}_i'$ and $\underline{M}_i'$ for $1 \leq i \leq MAX - 1$. (Each $\underline{L}_i'$ has at most $J + 1$ nonzero subdiagonal elements in spherical geometry and $(3/2)J - 1$ in slab or $3J - 1$ in slab with periodic boundary conditions, for that is the maximum number of subdiagonal elements that can appear in any column of M.) Moreover, one can define

$$\underline{L} = (\underline{L}'_{MAX-1}\underline{L}'_{MAX-2} \cdots \underline{L}_3'\underline{L}_2'\underline{L}_1')$$

and

$$\underline{U} = \underline{M}'_{MAX-1}.$$

Hence, $\underline{L}$ is a lower-diagonal matrix with a diagonal of unit elements and $\underline{U}$ is an upper-diagonal matrix with nonzero diagonal elements (and no more than $J+1$ nonzero superdiagonal elements in any column in spherical geometry; and $(3/2)J - 1$ in slab and $3J - 1$ in slab with periodic boundary conditions).

Overall storage in the TESS program is conserved by storing each subdiagonal column of an $\underline{L}_i'$ sequentially in a vector 3000 words long, and storing vectors on tape in as many 3000-word (or less) records as necessary. Likewise, the $\underline{U}$ matrices are stored row-wise backward in vector form on tape in 3000-word (or less) records.

After this matrix manipulation is accomplished, the original matrix equation can be expressed as $\underline{L}^{-1}\underline{L}\underline{M}\underline{N} = \underline{L}^{-1}\underline{U}\underline{N} = \underline{Q}$.

2. The second step is simply to premultiply the $\underline{Q}$ vector by $\underline{L}$, for it is obvious from the matrix equation that $\underline{L}\underline{L}^{-1}\underline{U}\underline{N} = \underline{U}\underline{N} = \underline{L}\underline{Q} = \underline{P}$.

3. The third step accomplishes the solution for $\underline{N}$ (all components $N(x_i, \mu_j)$ by a back solution using the matrix form $\underline{U}\underline{N} = \underline{P}$). This is possible because the last row of this matrix equation is a function of only a single component of the $\underline{N}$ vector; namely, $N(x_{MAX}, \mu_J)$.

Because the matrix $\underline{U}$ is an upper-diagonal matrix, each row can be successively used to compute a component of the $\underline{N}$ vector in terms of components evaluated by using the rows below, until, finally, the entire solution for the $\underline{N}$ vector is accomplished.

In these three steps the solution is accomplished for the given value of the $\underline{Q}$ vector, which of course depends upon the fixed volume sources, boundary sources, group-to-group scattered sources, and fission sources for a given group. If there are no fission sources in a TESS problem (and no upscattering is allowed), there is no iteration required to obtain the solution for any number of groups. (The maximum number of groups allowable in the program is 26.) If fission sources are allowed, the iterations required are only the outer iterations (common to the usual diffusion-theory solutions) that are necessary to converge the source distribution which determines, finally, the $\underline{Q}$ vector.

Moreover, the $\underline{L}$ and $\underline{U}$ matrices need be formed only once for each group of a TESS problem. Therefore, step 1 is performed for each group before the first iteration only. The outer iterations after the first proceed much more rapidly because only steps 2 and 3 are needed to compute the new $\underline{N}$ vector.

E. Energy-group Coupling

The TESS program, as already stated, allows up to a first-order Legendre polynomial representation of the scattering function for scattering within a given energy group, and down one group. For $\ell = 0$ scattering, 12 downscatter groups are allowed.

Energy groups are also coupled by fissions which produce source neutrons simultaneously in a number of groups.

Both forms of coupling are described by the equation for the source term $q(x, \mu)$ in Eq. 3. The source is assumed to be isotropic, except for the $\ell = 1$ scattering-in and down-one source, and in any group g the isotropic component source is given by

$$q_g(x) = \frac{1}{4\pi}\left[\sum_{m=1}^{g-1} \Sigma^0_{m\to g}(x)F_m(x) + \frac{\chi_g}{\lambda'}\sum_{m=1}^{G} \nu_m\Sigma_{f,m}(x)F'_m(x) + S_g(x)\right]$$

$$\cdot \frac{\text{particles}}{\text{cm}^3 \text{ sec steradian}} \cdot \qquad (8)$$

By definition,

$$q_g(x,\mu) = q_g(x) + \frac{1}{4\pi}\sum_{m=1}^{g-1} 3\Sigma^1_{m\to g}(x)\mu J(x),$$

$$F_m(x) = \int_0^{2\pi}\int_{-1}^{1} N(x,\mu)\, d\mu\, d\varphi \; \frac{\text{particles}}{\text{cm}^2\ \text{sec}},$$

where $F_m(x)$ is the usual scalar flux, and

$$J_m(x) = \int_0^{2\pi}\int_{-1}^{1} \mu N(x,\mu)\, d\mu\, d\varphi = \text{scalar current.}$$

$F'_m(x) = F_m(x)$ as calculated in the previous outer iteration. (In a first iteration, a source guess is used directly rather than a flux guess.)

$\Sigma^0_{m\to g}(x)(\text{cm}^{-1})$ is the $\ell = 0$ cross section for the scattering of neutrons from group m to g. (As noted, $m < g$, and $\Sigma_{m\to g}(x)$ is constant with respect to x in each region.)

$\Sigma^1_{m\to g}(x)(\text{cm}^{-1})$ is the $\ell = 1$ cross section for scattering neutrons from m to g.

χ_g is the fraction of fission neutrons released in group g. $\left(\text{Normally, } \sum_{g=1}^{G} \chi_g = 1.\right)$

ν_m is the number of neutrons produced per fission by neutrons absorbed in group m.

$\nu_m\Sigma_{f,m}(x)(\text{cm}^{-1})$ is the fission cross section for neutrons in group m multiplied by the number of neutrons per fission.

G is the total number of groups. $G \le 26$ in the TESS program.

λ' is equal to unity in those problems that are not eigenvalue problems. During the outer iterations of an eigenvalue problem, it is the computed eigenvalue from the last iteration.

$S_g(x)$ (particles/cm^3 sec) is the fixed volume source for group g. It remains constant throughout any problem, but it can be given pointwise as a function of x.

The fixed source $S_g(x)$ is assumed linear in x in any interval (x_i, x_{i+1}), the cross sections are constant in each region, and $F_g(x)$ is linear in (x_i, x_{i+1}), so that $q(x)$ is also linear in the same interval. Hence, assumptions with

regard to the source generated by downscattering, fissioning, and fixed volume sources are consistent with the assumptions made in deriving the difference equations from Eq. 3. The TESS program properly allows for discontinuity (or double values) of the sources defined by Eq. 8 at all region boundaries in slab geometry.

F. Integrated Balance Formulas

TESS, as output, computes balance data for each group and each region according to the following formulas for spherical geometry:

$$\text{Net leakage} = 4\pi x_R^2 \left[2\pi \int_{-1}^{1} N_g(x_R, \mu)\mu \, d\mu \right] - 4\pi x_L^2 \left[2\pi \int_{-1}^{1} N_g(x_L, \mu)\mu \, d\mu \right]$$

where x_L is the value of x at the left boundary of the region and x_R is the value of x at the right boundary,

$$\text{Absorptions} = 4\pi \int_{x_L}^{x_R} \left(\Sigma_{t,g} - \Sigma_{s0,g} - \sum_{m=g+1}^{G} \Sigma_{g\to m} \right) x^2 F_g(x) \, dx,$$

$$\text{Fixed-source production} = 4\pi \int_{x_L}^{x_R} S_g(x) x^2 \, dx,$$

and

$$\text{Fission-neutron production} = \frac{4\pi}{\lambda} \int_{x_L}^{x_R} \nu_g \Sigma_{f,g} F_g(x) x^2 \, dx.$$

Similarly, in slab geometry,

$$\text{Net leakage} = 2\pi \left[- \int_{-1}^{1} N_g(x_L, \mu)\mu \, d\mu + \int_{-1}^{1} N_g(x_R, \mu)\mu \, d\mu \right],$$

$$\text{Absorptions} = \int_{x_L}^{x_R} \left(\Sigma_{t,g} - \Sigma_{s0,g} - \sum_{m=g+1}^{G} \Sigma_{g\to m} \right) F_g(x) \, dx,$$

$$\text{Fixed-source production} = \int_{x_L}^{x_R} S_g(x) \, dx,$$

and

$$\text{Fission-neutron production} = \frac{1}{\lambda}\int_{x_L}^{x_R} \nu_g \Sigma_{f,g} F_g(x)\, dx.$$

In these final balance formulas, the $F_m(x)$ and the λ used are those of the same iteration that is being edited. Therefore, the balance will be imperfect to the degree that there is lack of convergence in the outer iterations when:

a. $\lambda = 1$ and the problem is not an eigenvalue problem, but there are fissions (some $\Sigma_{f,m}$ is nonzero).

b. The problem is an eigenvalue problem and the total fission source production is correct, but (because of lack of convergence in the scalar-flux distributions in space and energy) the production in a given region and group will not equal the losses.

G. Integration Formulas

Since the angular flux and angular adjoints are available from a particular problem (when so requested), there is no reason why the integral of the product should not be computed "exactly." The product integral would be "exact" to the extent that the angular flux and adjoint can be represented--as TESS in fact does--by bilinear functions of space and angle. Let f and g be the angular flux and adjoint, respectively, and each is assumed to be bilinear in r and μ, that is,

$$f(r,\mu) = f(r_i,\mu_j) + \frac{f(r_{i+1},\mu_j) - f(r_i,\mu_j)}{r_{i+1} - r_i}(r - r_i) + \frac{f(r_i,\mu_{j+1}) - f(r_i,\mu_j)}{\mu_{j+1} - \mu_j}(\mu - \mu_j)$$

$$+ \frac{f(r_{i+1},\mu_{j+1}) - f(r_{i+1},\mu_j) - f(r_i,\mu_{j+1}) + f(r_i,\mu_j)}{(r_{i+1} - r_i)(\mu_{j+1} - \mu_j)}(r - r_i)(\mu - \mu_j),$$

$$r_i \le r \le r_{i+1} \text{ and } \mu_j \le \mu \le \mu_{j+1}, \tag{9}$$

where the energy dependence has been omitted.

The product $f(r,\mu)g(r,\mu)$ is quadratic in r and μ and is fairly lengthy. However, the integral of the product over both angle and space can be written fairly simply. It is

$$\int_0^R \int_0^{2\pi} f(r,\mu)g(r,\mu)\, d\mu\, dv = \sum_{i=1}^{I-1} \frac{\Delta r_i}{3} \Bigg\{ \alpha 1i \sum_{j=1}^{n/2} \Big[b1j(f_{i,j}g_{i,j} + f_{i,J1-j+1}g_{i,J1-j+1})$$

$$+ \frac{b2j}{2}(f_{i,j+1}g_{i,j} + f_{i,j}g_{i,j+1} + f_{i,J1-j}g_{i,J1-j+1} + f_{i,J1-j+1}g_{i,J1-j}) + b3j(f_{i,j+1}g_{i,j+1} + f_{i,J1-j}g_{i,J1-j})\Big]$$

$$+ \frac{\alpha 2i}{2} \sum_{j=1}^{n/2} \Big[b1j(f_{i+1,j}g_{i,j} + f_{i+1,J1-j+1}g_{i,J1-j+1})$$

$$+ \frac{b2j}{2}(f_{i+1,j+1}g_{i,j} + f_{i+1,j}g_{i,j+1} + f_{i+1,J1-j}g_{i,J1-j+1} + f_{i+1,J1-j+1}g_{i,J1-j})$$

$$+ b3j(f_{i+1,j+1}g_{i,j+1} + f_{i+1,J1-j}g_{i,J1-j})\Big] + \frac{\alpha 2i}{2} \sum_{j=1}^{n/2} \Big[b1j(f_{i,j}g_{i+1,j} + f_{i,J1-j+1}g_{i+1,J1-j+1})$$

$$+ \frac{b2j}{2}(f_{i,j+1}g_{i+1,j} + f_{i,j}g_{i+1,j+1} + f_{i,J1-j}g_{i+1,J1-j+1} + f_{i,J1-j+1}g_{i+1,J1-j})$$

$$+ b3j(f_{i,j+1}g_{i+1,j+1} + f_{i,J1-j}g_{i+1,J1-j})\Big] + \alpha 3i \sum_{j=1}^{n/2} \Big[b1j(f_{i+1,j}g_{i+1,j} + f_{i+1,J1-j+1}g_{i+1,J1-j+1})$$

$$+ \frac{b2i}{2}(f_{i+1,j+1}g_{i+1,j} + f_{i+1,j}g_{i+1,j+1} + f_{i+1,J1-j}g_{i+1,J1-j+1} + f_{i+1,J1-j+1}g_{i+1,J1-j})\Big]\Bigg\}, \tag{10}$$

where the r and μ dependence are implicit in the subscripts, and

$$b1j = b2j = b3j = \frac{2\pi}{3}\Delta\mu_j,$$

$$\Delta\mu_j = \mu_{j+1} - \mu_j,$$

n = number of angular intervals,

$J1$ = number of angular points $\begin{pmatrix} = n + 2 \text{ for slab} \\ = n + 1 \text{ for sphere} \end{pmatrix}$,

I = number of space points over which the integration is being performed,

$$\Delta r_i = r_{i+1} - r_i,$$

and

	Slab	Sphere	Cylinder
$\alpha 1i$	1.0	$\frac{2\pi}{5}(r_{i+1}^2 + 3r_{i+1}r_i + 6r_i^2)$	$\frac{\pi}{2}(r_{i+1} + 3r_i)$
$\alpha 2i$	1.0	$\frac{2\pi}{5}(3r_{i+1}^2 + 4r_{i+1}r_i + 3r_i^2)$	$\pi(r_{i+1} + r_i)$
$\alpha 3i$	1.0	$\frac{2\pi}{5}(6r_{i+1}^2 + 3r_{i+1}r_i + r_i^2)$	$\frac{\pi}{2}(3r_{i+1} + r_i)$

Integration of the scalar product is similar and is obtained by substituting the appropriate scalar products in place of the summations over angle in Eq. 10.

In cylindrical geometry, the coefficients for Eq. 10 are

$$b1j = \frac{\Delta\varphi_j(1 + 2\cos^2\varphi_{j+1}) - (\cos\varphi_{j+1})(3\sin\varphi_{j+1} - 4\sin j) - \cos\varphi_j\sin\varphi_j}{(\cos\varphi_{j+1} - \cos\varphi_j)^2},$$

$$b2j = \frac{-\Delta\varphi_j(1 + 2\cos\varphi_{j+1}\cos\varphi_j) + (\cos\varphi_{j+1})(\sin\varphi_{j+1} - 2\sin\varphi_j) + (\cos\varphi_j)(2\sin\varphi_{j+1} - \sin\varphi_j)}{(\cos\varphi_{j+1} - \cos\varphi_j)^2},$$

and

$$b3j = \frac{\Delta\varphi_j(1 + 2\cos^2\varphi_j) - \cos\varphi_j(4\sin\varphi_{j+1} - 3\sin\varphi_j) + \cos\varphi_{j+1}\sin\varphi_{j+1}}{(\cos\varphi_{j+1} - \cos\varphi_j)^2},$$

where φ is the azimuthal angle.

Note that the current version of TESS does not have a cylindrical-geometry option.

H. Cross-section Homogenization

The flux and bilinear weighting schemes incorporated in TESS are based on the theoretical development by Nicholson.[5] The theory is predicated upon the idea that the system in question, either slab or spherical, is made up of repeating cells which are themselves of slab or spherical geometry. The cross sections of the cell composition may then be spatially averaged over one cell, or collapsed over energy, or both. This can be done using either bilinear or real-flux weighting. The resulting homogenized cross sections would then typically be used in an ensuing homogeneous calculation over the actual dimensions of the system. There are actually three different bilinear weighting options in TESS as well as three real-flux options. One of each trio is more correct for cases where the entire system is a sphere (or a cylinder that has roughly equal height and diameter); a second is better for a slab; and the third is preferable for any system in which the leakage is predominantly parallel to the plates.

Following are the expressions used for bilinear weighting of the total cross section, the isotropic scattering cross section, the fission cross section (averaged together with the fission spectrum χ and the number of neutrons per fission ν), and the first-order anisotropic scattering cross section, respectively:

$$\langle \Sigma \rangle_G = \frac{1}{4\pi V \eta_G} \sum_{g \text{ in } G} \iint d\Omega \, dV \, N_g^{c\dagger}(x, \Omega) \Sigma_g(x) N_g^c(x, \Omega)$$

$$- \frac{1}{4\pi V \eta_G} \sum_{g \text{ in } G} \iint d\Omega \, dV \, N_g^c(x, \Omega) \mu \frac{dN_g^{c\dagger}(x, \Omega)}{dx}, \tag{11}$$

$$\langle \Sigma^{is} \rangle_{JG} = \frac{1}{16\pi^2 V \eta_G} \sum_{g \text{ in } G} \sum_{j \text{ in } J} \iiint dV \, d\Omega \, d\Omega' \, N_g^{c\dagger}(x, \Omega) \Sigma_{jg}^{is}(x) N_j^c(x, \Omega'), \tag{12}$$

$$\langle \chi \nu \Sigma_f \rangle_{JG} = \frac{1}{16\pi^2 V \eta_G} \sum_{g \text{ in } G} \sum_{j \text{ in } J} \iiint dV \, d\Omega \, d\Omega' \, N_g^{c\dagger}(x, \Omega) \chi_g \nu_j \Sigma_{fj}(x) N_j^c(x, \Omega'), \tag{13}$$

and

$$\langle \Sigma^1 \rangle_{JG} = \frac{1}{4\pi^2 V \eta_G} \sum_{g \text{ in } G} \sum_{j \text{ in } J} \int dV \, \Sigma_{jg}^1(x) \int d\Omega \, |\mu| N_g^{c\dagger}(x, \Omega)$$

$$\int d\Omega' \, |\mu'| N_j^c(x, \Omega'), \tag{14}$$

where

$$\eta_G = \frac{1}{2\pi V} \sum_{g \text{ in } G} \iint d\Omega \, dV \, N_g^{c\dagger}(x, \Omega) |\mu| N_g^c(x, \Omega). \tag{15}$$

The angular flux $N_g^c(x, \Omega)$ and adjoint $N_g^{c\dagger}(x, \Omega)$ must be normalized as follows:

$$\sum_{g \text{ in } G} \int dV \int d\Omega \, N_g^c(x, \Omega) = 1; \tag{16}$$

$$\sum_{g \text{ in } G} \int dV \int d\Omega \, N_g^{c\dagger}(x, \Omega) = 1. \tag{17}$$

In these expressions, g is an energy group in the Gth homogenized energy group, V is the volume of the cell (actually just the thickness since the code is one-dimensional), and μ is the direction cosine of the angular flux. The integrals over volume are integrals over the thickness of the cell; most of the weighting options in TESS work only in slab geometry.

The homogenized cross sections, as given by Eqs. 11-15 are for bilinear weighting appropriate to a case in which the final homogeneous system is in slab geometry. One gets this weighting by setting the output option parameter NOT equal to 8. If the system in which the cell-homogenized cross sections are to be used is "pseudospherical," Nicholson's theory predicts that one gets slightly better results if $|\mu|$ is replaced in Eq. 15 by 1/2. This occurs for NOT = 5. However, if the system is such that the neutron leakage occurs primarily in a direction parallel to the plate structure, the recommended option is NOT = 10. In this mode, the only change is that in Eq. 15 for η_G, $|\mu|$ is replaced by $(2/\pi)\sqrt{1-\mu^2}$.

Similarly, the flux weighting formulas follow from Eqs. 11-15 merely by setting $N_g^{c\dagger}$ equal to unity, and in addition (in Eq. 15) only $|\mu| = 1/2$ for pseudospherical final geometry or $|\mu| \rightarrow (2/\pi)\sqrt{1-\mu^2}$ for leakage parallel to the plates. Note that for any of the flux-weighting-only options, the second term in Eq. 11 for the total cross section vanishes.

The integrations necessary in calculating the homogenized cross sections follow the assumptions of Sect. II.G above, that both the flux and the adjoint are bilinear functions of space and angle. Thus any product of a flux and an adjoint, each of which is of no higher order than linear in space or angle, would be integrated exactly.

The fission matrix given by Eq. 13 is in fact a full matrix, dependent upon both j and G in a nonseparable manner. From this, TESS calculates two more conventional quantities,

$$\nu_J \Sigma_{fJ} = \sum_G \langle \chi \nu \Sigma_f \rangle_{JG}$$

and

$$\chi_{JG} = \frac{\langle \chi \nu \Sigma_f \rangle_{JG}}{\nu_J \Sigma_{fJ}} .$$

On the assumption that the fission spectrum depends but weakly on the energy of the neutron causing the fission, TESS is programmed to handle a fission-spectrum <u>vector</u> only. Nevertheless, the bilinear weighting prescriptions in TESS will produce a χ-matrix in which the columns (labeled by J) will not be identical, although in practice they are very nearly so. Since TESS is incapable of using a χ-matrix, for each homogenized region the χ-vector for the energy group with the greatest production of fission neutrons is chosen. In case the homogenized cross sections are to be used in a code that can handle an energy-dependent fission spectrum, TESS punches out both the χ-matrix and the selected χ-vector, in XLIBIT format only (MIK = 2). Regardless of the output-punch option, the energy-dependent [illegible] ys included in the printed output.

I. Buckling

TESS has provisions for treating leakage as either a "source" modification or a "sink" modification. In the former, the source term for each energy group (take group g, for example), including the in-group scattering source, is multiplied by the nonleakage factor

$$\frac{\Sigma_{tg}}{\Sigma_{tg} + (DB^2)_g},$$

where Σ_{tg} is the total macroscopic cross section for group g, and $(DB^2)_g$ is the leakage appropriate to that group. The source modification can be used only in slab geometry, and DB^2 must be input one value per group. There is no provision for DB^2 to be region dependent. This leakage treatment was devised by Arne P. Olson.

The sink modification can be used in either slab or spherical geometry and treats DB^2 as a fictitious absorption cross section,

$$\Sigma_t' = \Sigma_t + \frac{B^2}{3\Sigma_t}.$$

This increase to the total cross section is made in every group. Here, B^2 rather than DB^2 is the input quantity, and may be a single value, group dependent, region dependent, or both group and region dependent. After the flux-iteration sequence has converged, the buckling correction is subtracted from Σ_t', returning the mixture total cross sections to their original values. The buckling correction is made at the mixture level, so material (or group and material)-dependent bucklings must be entered in addresses corresponding to the appropriate mixture numbers, not isotope numbers.

Investigation is going on at the time of the writing of this report concerning the relative merits of the source and sink treatments of the buckling. The only result so far is that for a theoretical slab reactor with leakage perpendicular to the plates, the cell calculation using the source method gave a k_{eff} that was somewhat closer to the exact value for the system. For most purposes, either method is probably quite accurate, considering the uncertainty in the buckling values one uses.

III. OUTER-ITERATION AND CONVERGENCE CYCLE

A. Source Iterations

Three classifications of problems may be run with the program:

Type		
A	Fissions	No fixed or boundary sources
B	No fissions	Fixed or boundary sources
C	Fissions	Fixed or boundary sources

Problem types A and C require outer iterations. Problem type B requires no outer iterations.

The fission density at a point is defined as

$$FD_i = \sum_{m=1}^{G} (F_{m,i}\nu_m\Sigma_{f,m,i}) \tag{18}$$

for the real solution; and, for the adjoint solution,

$$FD_i^{\dagger} = \sum_{m=1}^{G} (F^{\dagger}_{m,i}\chi_{m,i}), \tag{19}$$

where the i subscript denotes values at the point x_i.

In the problem type A, at the beginning of each iteration, the fission density is normalized so that its integral over the fissionable volume is some input value, FAC. The outer iteration begins with the solution for the group 1 fluxes at all spatial points. The code then solves for all flux values in the second group, using the recently obtained values for the first group flux for the scattering-in source. Finally, the last group fluxes are computed. The adjoint solution starts with the last group and works up to the first. Then new values of the fission density are computed, and the eigenvalue is defined as

$$\lambda = \frac{\int FD\, dV}{FAC}.$$

The fission density is then renormalized by dividing each new value by λ.

At this point a test for convergence is made. The problem may be made to converge pointwise on fission density or on the problem eigenvalue. If eigenvalue convergence is desired, the test is as follows:

$$\frac{\lambda_k - \lambda_{k-1}}{\lambda_k} \leq \epsilon, \quad \text{(For problem type C, } \lambda \text{ represents the total integrated source from fissions.)} \tag{20}$$

where k indicates the present iteration and ϵ is an input quantity (EPS1). If pointwise convergence is desired, each point value of the normalized fission density for a given iteration is divided by the respective normalized point for the previous iteration. Let E_{max} be the maximum value of this ratio and E_{min} the minimum value of this ratio. Then the test is

$$\frac{E_{max} - E_{min}}{E_{max}} \leq \epsilon. \tag{21}$$

Fission-density extrapolation at each point x_i may be applied after the third iteration, by one of two possible procedures. A linear extrapolation of the following form can be used

$$FD'_{i,k} = FD_{i,k}(1 + \theta) - \theta FD_{i,k-1},$$

where k indicates the iteration index and θ is an input value. Experience has shown, however, that a bad choice of θ can be worse than no extrapolation, especially for large systems. Therefore, a Chebyshev polynomial extrapolation procedure, which determines optimum extrapolation parameters, is used, unless the code user desires something else. This code uses the Chebyshev extrapolation subroutine developed for one-dimensional diffusion-theory codes by Putnam[6] and is based upon the procedure developed for PDQ-5.[7]

B. Search Iterations

The search process consists of the search routine modifying either a concentration (JSP = 1) or a zone thickness (JSP = 2), followed by a series of outer iterations to converge the flux or fission density or both. Any series of outers can be terminated in three different ways: (1) The convergence criterion EPS1 is satisfied, (2) the limit on the number of outers for a given search cycle, KIT1, is reached, or (3) the limit on the total number of outers for the problem, ITOUT, is reached. In the event of (3), the problem is terminated; in the event of either (1) or (2), the search routine is reentered unless the current value of the eigenvalue differs from SEN by less than EPS3. In the latter case, if (1) is true, the problem is finished; if (2) is true, the limit KIT1 is ignored and the outers continue until either the problem is converged or $|\lambda - SEN|$ is no longer less than EPS3, which causes immediate transfer to the search routine.

After the search routine has used the second guess (SGES) input by the user, subsequent guesses for the search parameter are generated using quadratic interpolation based on the three most recent values of the search parameter. If interpolation (or extrapolation) sends the parameter to zero or negative, an alternate guess is generated by dividing the smallest of the values used in the interpolation by 5.0.

The user should be cautioned at this point that transport-theory searches are expensive. It is usually advisable to run perhaps three change-case problems that will bracket the dimension or concentration desired, interpolate to obtain the parameter desired, and then run a fourth case to confirm the choice. If the search option is used, care should be exercised to allow a sufficient number of source iterations so that the eigenvalue is reasonably converged before the parameters are changed. Otherwise, bad guesses can be generated, thus requiring much excess computer time.

For the dimension search, the input includes the region number of the region whose thickness is to be varied (KREG). For the concentration search, three quantities should be entered:

1. An index to indicate the material concentration to be varied (NSOS).
2. An index to indicate the material concentration to be used as "filler" or "diluent" (NFOS).
3. A search ratio (RR).

Let N_1 and N_2 be the volume fractions or atom densities of the search material and the filler material, respectively, according to the initial guess--i.e., as input in the CONC vector. Let N_1' be the second guess for the search parameter (SGES). The code finds the corresponding second guess for the filler concentration or volume fraction, N_2', by solving

$$N_2' = N_2 - RR \times (N_1' - N_1). \tag{22}$$

Therefore when the N's are volume fractions and the total volume is to be kept constant (i.e., $N_1 + N_2 = N_1' + N_2' =$ constant), RR = 1. However, when the N's are atom densities, the value of RR that preserves the total volume is given by solving

$$RR = \frac{\rho_2 A_1}{\rho_1 A_2}, \tag{23}$$

where ρ_1 and ρ_2 are material densities and A_1 and A_2 are atomic weights.

The same procedure is repeated to generate the new value of filler atom density or volume fraction each time the search routine is entered.

IV. INPUT

There is a minimum of three main sections of input for each problem:

1. Alphanumeric card.
2. Fixed-point data.
3. Floating-point data.
4. Optionally, cross-section tape element names.
5. Optionally, reaction-rate element names and cross sections.

The input format has been constructed for a maximum change-case and serial-case capability, a minimum number of input cards, and a thorough input-error check. (A change case is a problem that is the same, except for minor changes, as the one preceding it in a set of problems run serially on the computer.) Each section of input will now be described.

A. Alphanumeric Card (change-case option)

The alphanumeric card may contain any desired alphanumeric information in columns 1-72 and a two-digit test in columns 73-74. A one in column 1 will start all printouts at the top of a new page.

The two-digit entry in columns 73-74 is used to tell the program whether a change-case problem follows. If columns 73-74 contain 88, any homogenized cross sections computed in this problem will be stored for the next problem. If columns 73-74 contain any other two digits, then any homogenized cross sections generated in the present problem will not be saved in memory for succeeding change-case problems.

If columns 73-74 contain 99, all the input area of memory will be zeroed out before reading the input for the present case.

The alphanumeric card must always be physically the first card in every case.

B. Card Format for Fixed-point Data

The card format for the fixed-point data is as follows:

Card columns 1-2	Number of pieces of data on this card (right-adjusted*): $1 \leq$ No. ≤ 20.
Card columns 3-6	Nonzero: This is the last fixed-point data card.
	Zero (or blank): This is not the last fixed-point data card.

*Right-adjusted means that the last significant digit of a number is positioned in the card columns at the extreme right of the field.

Card columns 7-12	Address of first piece of data on this card (right-adjusted).
Card columns 13-15 16-18 19-21, etc.	Up to 20 pieces of integral data (each right-adjusted) in I3 format.

Section IV.F below contains the addresses, identification, and explanation of the fixed-point data.

C. Card Format for Floating-point Data

The card format for the floating-point data is as follows:

Card columns 1-2	Number of pieces of data on this card (right-adjusted): $1 \leq$ No. ≤ 5.
Card columns 3-6	Nonzero: This is the last floating-point data card. Zero (or blank): This is not the last floating-point data card.
Card columns 7-12	Address of first piece of data on the card (right-adjusted).
Card columns 13-24 25-36 37-48 49-60 61-72	Up to five pieces of floating-point data in E12.6 format.

Note: The floating-point data format may be illustrated by the following examples, all acceptable, for the number 3.1415927:

+	3	1	4	1	5	9	2	7	-	0	1
3	.	1	4	1	5	9	2	7	+	0	0
3	.	1	4	1	5	9	3	E	+	0	0
3	.	1	4	1	5	9	2	7	0	+	0
3	1	.	4	1	5	9	2	7	0	-	1
.	0	3	1	4	1	5	9	2	7	+	2
(b)	3	.	1	4	1	5	9	2	7	+	0
3	1	4	1	5	9	2	7	E	-	0	1
3	.	1	4	1	5	9	2	7	0	(b)	(b)

(b) indicates blank

Section IV.G below contains the addresses, identification, and explanation of the floating-point data.

D. Optional Cross-section Tape Information

If MMIX (fixed-point address 16) is nonzero, the program expects additional information after the last floating-point data card. In general, this is only required for the first case of a series unless MMIX is set nonzero in a change case, because the program sets MMIX = 0 during the first problem.

There will be one or two cards depending upon the value of MMIX. The card formats are as follows:

a. Card 1

Card columns 1-5	Alphanumeric cross-section set identification--must be the same as on the tape.
Card columns 7-12 13-18 19-24, etc.	Up to 11 alphanumeric material names--must be the same as on the tape.

b. Card 2. If more than 11 elements are requested, the second card is required.

Card columns 1-6 7-12, etc.	Alphanumeric material names.

The code limitation of no more than 20 isotopes from tape is not as restrictive as it might seem. One can form macroscopic mixtures and then read more cross sections on a succeeding change case, thus utilizing repeatedly memory locations that do not contain mixture cross sections that must be saved for the neutronics calculation.

Note: There must be a one-to-one correspondence between this list of materials requested from tape and the NTMIX vector (fixed-point addresses 166-185).

This program will read only cross-section tapes that are in the XLIBIT format.[12] If the tape contains scattering cross sections for $\ell > 1$, the total $\ell > 1$ values are subtracted from Σ_T and $\Sigma_{s,g\rightarrow g}$ ($\ell = 0$) for a pseudotransport correction.

The code cannot handle energy-dependent fission spectra, and if the tape contains it, the job will be rejected. Also, if the tape contains downscatter information for more than 12 groups, or for more than one P_1 downscatter group, this will be ignored, and a message will be printed out indicating the materials having too many downscatter groups.

E. Optional Reaction-rate Material Names and Cross Sections

If, and only if, NRATE (fixed-point address 225) is nonzero, the program expects additional information after the cross-section tape information. This additional information consists of two parts; the first gives the names of isotopes for which reaction rates are desired (NRATE names), and the second part is cross sections.

The reaction-rate isotope names are input on one or more cards in (12A6) format. If an isotope name corresponds with a tape name, the capture and fission cross sections from the tape will be read in for that isotope. Note that the cross-section tape is read after the input deck.

If NRATE is negative, the reaction-rate cross sections are read in floating-point format (see Sect. IV.C), with the last card having a nonzero value in columns 3-6. The addresses are as follows:

Address	
1-26	Fission cross section for isotope 1, groups 1-26
27-52	Fission cross section for isotope 2, groups 1-26
⋮	
625-650	Fission cross section for isotope 25, groups 1-26
651-676	Capture cross section for isotope 1, groups 1-26
677-702	Capture cross section for isotope 2, groups 1-26
⋮	
1275-1300	Capture cross section for isotope 25, groups 1-26

If NRATE is not negative, the second part (cross sections) referred to above does not appear.

If reaction rates are desired for isotopes for which the cross sections are read from a tape, the reaction rates must be requested in the same problem in which the cross sections are read from the tape. (This is true even if the reaction rates are desired only in a later change-case problem.) If this is done, and NRATE is not changed in subsequent change-case problems, the reaction rates for the isotopes requested will be calculated in the initial problem and in all following change-case problems. The program will not calculate reaction rates if NRATE is zero in the problem in which the cross sections are read from a tape, then is set to some number greater than zero in a later change-case problem.

F. Input Instructions for Fixed-point Data

Fixed-point Address	Name (dimension)	Explanation	
1	MAX	Number of space points	$3 \le MAX \le 150$
2	JMAX	Number of regions	$1 \le JMAX \le 40$
3	NGR	Number of groups	$1 \le NGR \le 26$
4	N	Number of angular intervals	N = 2,4,6,8,10,12,14,16,18,20
5	NDS	Number of ℓ = 0 downscatter groups	$0 \le NDS \le 12$
6	NPS	Number of ℓ = 1 downscatter groups	$0 \le NPS \le 1$
7	NOT	Output control = 0, eigenvalue, scalar fluxes, fission density.	
		= 1, output for NOT = 0 plus neutron balance.	
		= 2, output for NOT = 1 plus angular fluxes.	
		= 3, output for NOT = 2 plus iteration fission densities.	
		= 5, output for NOT = 2 plus perturbation integrals, both flux and adjoint calculations are done automatically, homogenized cross sections are bilinearly weighted without "μ-weighting."	
		= 6, output for NOT = 5, except that no adjoint calculation is done and the homogenized cross sections are flux weighted (without "μ-weighting") only.	
		= 7, same as NOT = 6, except that the homogenized cross sections are flux weighted <u>with</u> "μ-weighting."	
		= 8, same as NOT = 5, except that the homogenized cross sections are bilinearly weighted <u>with</u> "μ-weighting."	
		= 9, same as NOT = 6, except that the homogenized cross sections are flux-weighted with "$\sqrt{1-\mu^2}$ weighting."	
		= 10, same as NOT = 5, except that the homogenized cross sections are bilinearly weighted with "$\sqrt{1-\mu^2}$ weighting."	
		= 11, = 12, = 13, = 14, = 15, = 16	same as NOT = 6, except that no flux or adjoint calculation is done. The fluxes and adjoints are assumed to be stored in memory from an earlier problem.
8	ITOUT	Maximum number of outer iterations; initially set to 50 unless read in. If ITOUT = 40, one gets in addition a great deal of extra printout from the homogenization routine. If ITOUT = 0, no outer iterations are performed.	
9	LCO	Fission-density convergency option:	
		= 0, pointwise convergence	
		= 1, eigenvalue convergence	
10	MIK	Cross-section punch option:	
		= 0, no punch	
		= 1, punch in TESS format	
		= 2, punch in XLIBIT format (see Sect. IV.H.3)	
		= 3, punch both TESS format and XLIBIT format	

Fixed-point Address	Name (dimension)	Explanation
11	LPG*	Fission-density guess option (see Sect. III).
		Problem type A = 0, flat guess, integral normalized to FAC.
		= 1, input guess, integral normalized to FAC.
		= 2, guess from previous problem, integral normalized to FAC.
		Problem type B = ignored.
		Problem type C = 0, zero guess.
		= 1, input guess, not renormalized.
		= 2, guess from previous problem, not renormalized.

*Note: Search problems will leave LPG = 1, so change-case problems following a search problem should usually have LPG as input. For an adjoint calculation, the code internally gives itself a flat guess, ignoring LPG, because a flat guess is about as good as can be done for an adjoint calculation. Thus, setting LPG = 2 is advantageous when a real-flux calculation follows a similar real-flux calculation having the same number of mesh points. Whenever NOT > 10, POWR1 is zeroed; therefore, if a real-flux calculation follows, LPG should not be 2.

12	IDP	Input-print option
		= 0, print input.
		= 1, do not print input.
13	MUTEST	Angular-interval option
		= 1, equal intervals in cos θ.
		= 2, equal intervals in θ.
		= 3, read in.
14	JSP	Search option = 0, no search.
		= 1, concentration search.
		= 2, zone thickness search.
15	NMIX	Number of words needed to specify mixtures $0 \le NMIX \le 40$ (see MIX and CONC).
16	MMIX	Number of elements desired from cross-section tape.
		$0 \le MMIX \le 20$. This is set to zero by the program after the cross sections for the first problem have been read in, but can be changed by subsequent change-case input, if desired.
17	KREG	Region number for zone thickness search (JSP = 2).
18	NSOS	Position in mixture vector (MIX) of search concentration (JSP = 1).
19	NFOS	Position in mixture vector (MIX) of diluent concentration (JSP = 1). If NFOS = 0, no diluent.
20	KIT1	Index to determine maximum number of outer iterations per search iteration.
21	IBUK	Buckling-input option
		= 0, no buckling input.
		= 1, buckling material- and group-independent (one value).
		= 2, buckling material-dependent (up to 25 values).
		= 3, buckling group-dependent (up to 26 values).
		= 4, buckling group- and material-dependent.
		= 5, group-dependent DB^2 values input, modified source leakage treatment used. (IG = 1, only.)

Fixed-point Address	Name (dimension)	Explanation
22	MADJ	Adjoint Option = 0, real-flux calculation only. = 1, adjoint calculation only. = 2, both flux and adjoint calculation. (If NOT = 5, 8, or 10, MADJ is set = 2 internally.)
23	MFR	Periodic-boundary-condition option = 0, no. = 1, yes.
24	IG	Geometry indicator = 1, slab. = 3, sphere. (If IG is negative, the output will include a checkout dump of tremendous proportions.)
25	IEXOP	Fission-density extrapolation option = 0, Chebyshev polynomial extrapolation. = 1, no extrapolation. = 2, linear extrapolation.
26-65	II(40)	Upper-point index for each region (JMAX values) $1 < II(J) \leq MAX,\ J = 1,\ JMAX$ $II(J) < II(J+1)$
66-105	MIR(40)	Material number in each region (JMAX values) $1 \leq MIR(J) \leq 25,\ J = 1,\ JMAX$
106-145	MIX(40)	Material numbers to specify mixtures (NMIX values) (see CONC).
166-185	NTMIX(20)	Material numbers corresponding to tape-element names in a one-to-one correspondence (MMIX values).
186	NHGP	Number of homogenized groups (only applies if NOT ≥ 5). $0 \leq NHGP \leq 26$
187	NHREG	Number of homogenized regions (only applies if NOT ≥ 5). $0 \leq NHREG \leq 6$
188-197	LHREG(10)	Last problem region in each homogenized region (NHREG values)
198-223	LHGP(26)	Last problem group in each homogenized group (NHGP values)
224	LFREG	Material number of first homogenized region to designate addresses for storing and punching homogenized cross sections; $1 \leq LFREG \leq 25$.
225	NRATE	Number of materials for which reaction rates are desired. If negative, some cross sections will be read in with the input data (see Sect. IV.E).
250	ISKIP	Problem Skip Option; if ISKIP ≠ 0, the problem will be skipped; however, all input printing and checking and cross-section mixing are done.

G. Input Instructions for Floating-point Data

Floating-point Address	Name (dimension)	Explanation
1	EPS1	Convergence criterion on fission density.
2	Not used	
3	EPS3	Convergence criterion for search calculations.
4	FAC	Fission-density normalization factor. Set = 1.0 if not read in.
5	THETA	Power-extrapolation factor (only applies if IEXOP = 2).
6	SEN	Desired value of eigenvalue for search problems.
7	SGES	Second guess for search parameter (ΔR or CONC).
8	RR	Search ratio, for concentration searches to preserve the sum of volume fractions of fuel and diluent (see Sect. III.B).
9	XIN	Radius at left (inner) boundary; assumes XIN = 0 for sphere. XIN must be ≥ 0.
10-49	DELR(40)	Value of ΔR for each region (JMAX values).
50-699	SIGT(26,25)	Total (or transport) cross section by group and material (see Appendix A).
700-1349	SIGS(26,25)	Zero-moment nondegrading scatter cross section (see Appendix A).
1350-1999	SIGS1(26,25)	First-moment nondegrading scatter cross section (see Appendix A).
2000-2649	VUSIG(26,25)	Fission cross section multiplied by number of neutrons emitted per fission (see Appendix A).
2650-3299	CHI(26,25)	Relative portion of fission spectrum emitted in each group for each fissioning material (see Appendix A).
3300-9149	STR(234,25)	Zero-moment group-transfer cross sections (maximum of 12 downscatter groups) (see Appendix A).
9150-9774	STR1(25,25)	First-moment group-transfer cross sections (maximum of one downscatter group) (see Appendix A).
9775-10424	VINV(26,25)	Inverse velocity for each group and material (see Appendix A).
10425-10476	ALPHA(2,26)	Mirror-albedo coefficients by boundary and group (initially set = 1.0, 1.0 for slab or to 1.0, 0.0 for sphere by the code). 10425, 10426; group 1 (left, right). ... 10475, 10476; group 26 (left, right).
10477-10528	BETA(2,26)	Isotropic-albedo coefficients by boundary and group. BETA = 1.0 causes every neutron striking the boundary surface to be reflected back isotropically. A value of BETA = χ will in general cause the reflected current to be equal to χ times the outgoing current, and the reflected flux is isotropic.
10529-11048	GAMMA(10,2,26)	Legendre-polynomial coefficients for a diffuse boundary source, by degree (0-9), boundary, and group. To obtain the same angularly shaped source with respect to directions into the medium at both boundaries, the odd coefficients should be of opposite signs.

Floating-point Address	Name (dimension)	Explanation
11049-11620	DELTA(22,26)	Fixed-boundary fluxes by angular point and group. 11049: fixed right-boundary flux, group 1, μ = -1. . . . 11049 + N/2: fixed right-boundary flux, group 1, μ = 0. 11050 + N/2: fixed left-boundary flux, group 1, μ = 0. . . . 11049 + N + 1: fixed left-boundary flux, group 1, μ = +1.
11621-11642	EMU(22)	Values of cosines of the angles for each angular point starting with -1.0. In slab geometry there must be N + 2 values (two μ = 0 values), and in spherical geometry, N + 1 values (one μ = 0).
11643-11682	CONC(40)	Material atomic densities or volume fractions for cross-section mixing, by position (NMIX values).
	Example:	Form three mixtures: MAT 1 = X_1 * MAT 2 + X_2 * MAT 3, MAT 6 = X_3 * MAT 10 + X_4 * MAT 15 + X_5 * MAT 20, and MAT 25 = X_6 * MAT 1 + X_7 * MAT 5. Then NMIX = 10, and the MIX and CONC vectors would contain the following values:

Position	1	2	3	4	5	6	7	8	9	10
MIX	1	2	3	6	10	15	20	25	1	5
CONC	0	X_1	X_2	0	X_3	X_4	X_5	0	X_6	X_7

11683-11871	POWR1(189)	Fission density guess by source point (2 at each interface).
11872-12521	BUCK(26,25)	Optional Buckling Specification. IBUK = 1: Enter one value of B^2 in 11872. IBUK = 2: Enter one value for each material, starting in 11872. IBUK = 3: Enter one value for each group, starting in 11872. IBUK = 4: Enter by material, and for each material, by group. Material 1, group 1, goes in 11872; material 2, group 1, in 11898; etc., down to material 25, group 1, in address 12496. IBUK = 5: Enter one value of DB^2 (not B^2) for each group (NGR values), starting in 11872.
12522	SVM(189,26)	Fixed volume source by point and group (two at each interface).

H. Additional Notes on Input

1. Δx Criterion (DELR, floating-point address 10)

Each region must have a positive Δx specified. A negative or zero value causes an error indication to be printed and the job to be terminated.

The criterion for choosing the proper Δx_i values in each region depends upon the ratio of the nondegrading scattering cross section to the total cross section and on the order of the approximation. The physical exponential attenuation of the angular flux over each Δx interval is approximated according to the linearity assumption by a function of the form

$$e^{-x} \simeq \frac{1 - x/2}{1 + x/2}.$$

The Δx's in Table I have been chosen so that the angular flux neighboring the $\mu = 0$ flux does not go negative due to this approximation. The criterion is derived using this flux since it has the greatest amount of material to traverse per interval. Negative fluxes might still occur due to anisotropy in the angular distribution coupled with the linearity in μ that is assumed. In the latter case, a higher angular approximation should be tried.

TABLE I. Maximum Recommended Mesh-interval Size[a]
(Δx in mean free paths)

	Σ_{s0}/Σ_t				
n	1	0.8	0.5	0.1	0
2	2/3	5/9	4/9	20/57	1/3
4	2/9	5/24	4/21	20/117	1/6
6	2/15	5/39	4/33	20/177	1/9
8	2/21	5/54	4/45	20/237	1/12
10	2/27	5/69	4/57	20/297	1/15
12	2/33	5/84	4/69	20/357	1/18
14	2/39	5/99	4/81	20/417	1/21
16	2/45	5/114	4/93	20/477	1/24
18	2/51	5/129	4/105	20/537	1/27
20	2/57	5/144	4/117	20/597	1/30

[a]This table is an expansion of a similar table in Ref. 2.

The values in Table I may be considered as "very safe" and usually may be increased at least by a factor of two in regions where the transport is unimportant and/or there are large fixed sources.

2. Additional Boundary-condition Information

The user may specify any number for either the mirror or isotropic albedo coefficients; however, if the sum $A_{s,g} + B_{s,g}$ at any surface in any group is greater than 1.0, then the returning current will be larger than the exit current. Such a situation is usually physically unrealistic, but in certain problems a surface albedo greater than one is a valid representation. For example, a thin film of ^{239}Pu at a boundary of symmetry might be very well represented by setting $A_{s,g}$ and $B_{s,g}$ to nonzero values with $A_{s,g} + B_{s,g}$ greater than one. In these cases in which a boundary surface actually serves as a multiplying type of source, the program will still accept problems of type A, B, or C according to Sect. III.A without detecting a boundary source as such. Therefore, care should be taken to avoid trying to find solutions to problems with no steady-state or persisting solution. If the periodic boundary condition is specified (MFR = 1), the albedo coefficients determine the incoming flux at one boundary from the outgoing flux at the opposite boundary.

3. XLIBIT[12] Output-card Format

Because only the cross sections needed in the TESS code are available for homogenization--namely, Σ_t, $\nu\Sigma_f$, and the scattering cross sections--the XLIBIT type-1 card punched by TESS contains the following:

The Σ_{cap} field = 0,

the $\Sigma_{fiss,i}$ field = $\Sigma_{abs,i} = \Sigma_{t,i} - \sum_{j=0}^{NDS} \Sigma_{s,i\rightarrow i+j}$,

and

the ν field = $\nu\Sigma_{fiss}/\Sigma_{abs}$.

The code punches both kinds of XLIBIT type-2 cards; i.e., it punches out a fission-spectrum fraction matrix (which is a separate fission spectrum for each energy group) as well as a group-independent fission-spectrum fraction vector. TESS can use only the latter, but for the purposes of codes that can use it, the matrix is also punched out. Thus the user should discard one or the other from the deck of punched cards. Any homogenization option using flux weighting only usually yields an energy-independent χ-vector, although only for NOT = 6 will this vector be the same as the input χ. An exception would be a case in which one spatially collapses over a cell containing at least two regions with different χ-vectors. Any bilinear weighting option typically results in a χ-matrix.

I. Error Indicators

The following error indications are printed off-line when an input error or inconsistency is detected. When such an error is detected, the problem and all succeeding change cases are skipped over. The following is a list of error printouts and the errors that cause them.

1. ERROR, ADDRESS FORMAT FOR FIXED POINT DATA
   a. Number of data words on card (columns 1-2) is zero, negative, or greater than 20.
   b. Fixed-point address (columns 4-7) zero, negative, or greater than 250.
2. ERROR, ADDRESS FORMAT FOR FLOATING POINT DATA
   a. Number of data words on card (columns 1-2) is zero, negative, or greater than five.

b. Floating-point address (columns 4-7) zero, negative, or greater than 17480.

(For both of the above error indications, the contents of the card that is in error are printed on-line and off-line.)

3. ERROR, NUMBER OF POINTS TOO LARGE

   MAX > 150

4. ERROR, NUMBER OF POINTS TOO SMALL

   MAX < 3

5. ERROR, NUMBER OF REGIONS TOO LARGE

   JMAX > 40

6. ERROR, NUMBER OF REGIONS IS ZERO

   JMAX ≤ 0

7. ERROR, NUMBER OF GROUPS TOO LARGE

   NGR > 26

8. ERROR, NUMBER OF GROUPS IS ZERO

   NGR ≤ 0

9. ERROR, NUMBER OF DOWN- OR UP-SCATTERS TOO LARGE

   NDS > NGR, NPS > NGR

   NDS > 12, NPS > 1

10. ERROR, ZERO OR NEGATIVE MATERIAL NUMBER

    MIR (I) ≤ 0

11. ERROR, UPPER REGION BOUNDARY POINT NON-INCREASING OR II(1) IS LESS THAN OR EQUAL TO 1.

    a. II(J) ≤ II(J - 1)

    b. II(1) ≤ 1

12. ERROR, ZERO DELTA R

    DELR(I) ≤ 0

13. ERROR, NO NON-ZERO SIGMA TOTAL

    SIGT = 0 for all regions and groups

14. ERROR, INCONSISTENT ANGULAR INPUT

    a. N > 20

    b. N < 2

    c. N odd

d. $\mu_i \neq -\mu_{J-i+1}$, $J = \begin{cases} N+2, \text{ slab} \\ N+1, \text{ sphere} \end{cases}$

e. $\mu_i = \mu_j$ for $i \neq j$

15. MATERIAL NUMBER GREATER THAN 25

    MIR (I) > 25

16. ERROR, GEOMETRY SPECIFIED INCORRECTLY

    $|IG| \neq 1$ or $|IG| \neq 3$

17. CANT FIND SET ID

    The cross-section set name does not correspond to any set name on the cross-section tape.

18. THIS MAT NOT IN CROSS SECTION SET - (material name)

    An input isotope name does not correspond with any on this tape.

19. GROUP DEPENDENT FISSION SPECTRUM

20. MIXING VECTOR MUST HAVE AT LEAST TWO ELEMENTS

    NMIX = 1 or NMIX < 0

21. INCONSISTENT CONCENTRATION VECTOR

    CONC(1) $\neq$ 0

    CONC(NMIX) = 0

    CONC(I) = CONC(I+1) = 0

22. MATERIAL NUMBER GREATER THAN 25 IN MIXTURE VECTOR

    MIX (I) > 25

23. ERROR, NO SOURCE IN INHOMOGENEOUS CASE

    ITOUT = 0, no fixed or boundary sources.

24. ERROR, ALL FISSION X-SECTS ZERO

    ITOUT > 0, no fission cross sections

25. ERROR, NO NON-ZERO CHI

    No fission-spectrum input for problem containing fission cross section.

26. SEARCH FAILED, TWO CONSECUTIVE SEARCH GUESSES ARE EQUAL, CHECK SECOND GUESS

27. FOUND WRONG SET

    The code cannot find the desired cross-section set on the tape.

28. EOF OR PARITY ERROR ON LIB TAPE

V. PROGRAM INFORMATION

A. General Information and List of Subroutines and Overlays

TESS is written in FORTRAN and assembled under the CDC-3600 SCOPE 6.2114 monitor system. Because of core storage limitations, the code is divided logically into five overlays. (The overlay feature of the SCOPE monitor allows segments of programs to be stacked on tape permanently and to be called from the tape into core storage when needed. Overlays transfer data by means of common storage.) The general logical purpose of each overlay and the subroutines used in each are as follows:

Overlay 1

Reads, checks, and prints input. Initializes angular and geometric data, mixes cross sections, and reads cross-section tape. Initializes fission source.

Subroutines

INPR	Prints input data.
MIXX	Reads cross-section tape and mixes cross sections.
SCHECK	Initializes fission source.
SEARCH	Changes search parameter.

Overlay 2

Sets up the original matrix M for each group and writes the L and U matrices on tape.

Subroutines

OPINT SETUP ALQS AQSJL	Calculate auxiliary coefficients for use in computing the elements of the original matrix M for each group.

Segment 1

MTXSET	Calculates boundary-condition matrix elements and boundary sources; calculates the matrix elements of, and operates on, the matrix M to form matrices L and U, for slab geometry.

Segment 2

MTXSES Same as MTXSET, except for sphere.

Segment 3

REVERSE Reverses the coefficient matrix for the adjoint calculation.

Overlay 3

Obtains angular and scalar flux solution for each group. Obtains eigenvalue and fission source. Checks for fission density and/or flux and/or search convergence.

Subroutines

SOURCE Multiplies source vector by L matrix, and uses U matrix to back-solve for angular fluxes.

CONV. Calculates fission source and checks for fission-density convergence.

EXTRAP Fission-density extrapolation routine

Overlay 4

Prints output data; computes and prints reaction rates.

Overlay 5

Computes integrals of flux and adjoint for perturbation analysis and for cross-section homogenization; computes prompt-neutron lifetime.

Subroutines

HOMOG Computes space-and-energy homogenized cross sections.

XLPCH Punches homogenized cross sections in XLIBIT format, if desired.

B. Tape Assignments

	FORTRAN IV Logical Number
All BCD printed OUTPUT	61
All BIN and BCD Input	60
Punched card output	62

Intermediate binary	1, 2, 3, 4, 8
Program (overlay) tape	29
Cross-section tape	11

C. Input-deck Order

The deck setup is as follows:

a. Job card

b. Accounting card

c. 7/9 EQUIP, 29 = (TESS 71 OVLY, 2), RO, SV

d.* 7/9 EQUIP, 11 = (cross-section tape label), RO, SV

e. 7/9 LOADMAIN, 29, 200, 99999, 7

f. Problem data

VI. OUTPUT DATA

The general complexity of the output for each problem is governed by the two fixed point input words NOT and IDP.

IDP (Fixed-point Address 12) governs the option of printing the input to the problem off-line. The input print consists of the following data:

1. General fixed-point data, such as number of groups, regions, points, and the various options.

2. The floating-point words EPS1, XIN, THETA, FAC, SGES, SEN, and RR as defined previously.

3. General region data, consisting of region number, material number in each region, the maximum index value of a point in the region, the mesh-point spacing, and the outer (right) value of the radius.

4. The μ_j values for each value of j.

5. The mixture data including material numbers and concentrations (if any).

*Only needed if cross sections are read from tape.

6. The fixed-volume-source input (if any).

7. The boundary-condition specifications for each group.

8. The cross-section data for each region.

9. The fission-spectrum data.

NOT (Fixed-point Address 7) governs the complexity of the output print according to the following table:

Value of NOT

NOT = 0

1. The final eigenvalue for problem type A (or the final integrated fission source for problem type C).
2. The radius, group scalar flux, and fission-neutron density at each point, including two values for the fission density at each interface.

NOT = 1

1. Output for NOT = 0.
2. The neutron balance by region for each group, the system total summed over all groups. The neutron-balance data include integrated (with respect to the spatial variable) flux, integrated fixed-volume source, integrated absorption, integrated fissions, and net leakage.

NOT = 2

1. Output for NOT = 0.
2. Output for NOT = 1.
3. Auxiliary output by group and space point, with the scalar fluxes, hemispherical currents to the left, hemispherical currents to the right, and the angular fluxes listed opposite each value of the spatial variable.

NOT = 3

1. Output for NOT = 0.
2. Output for NOT = 1.
3. Output for NOT = 2.
4. The spatially dependent fission density for each iteration for problem types A and C.

NOT = 5

1. Output for NOT = 0.
2. Output for NOT = 1.
3. Output for NOT = 2.

4. Integrals of flux and adjoint for perturbation analysis, along with prompt-neutron lifetime (if inverse velocities are input), and homogenized cross sections, bilinearly weighted without μ-weighting. This weighting is considered preferable to NOT = 8 or 10 if the system in which the homogenized cross sections are to be used is spherical* rather than slab (see Sect. II.H). However, the homogenization procedure is only for a slab geometry cell calculation, so NOT = 5 should not be selected unless IG = 1.

NOT = 6 — Same as NOT = 5, except that the homogenized cross sections are flux weighted only. (No adjoint calculation is done, regardless of the value of MADJ.) This is flux weighting without μ-weighting and is considered preferable to NOT = 7 or 9 if the system in which the homogenized cross sections are to be used is a sphere.* This is the only cross-section homogenization option that uses only the scalar fluxes (and not the angular fluxes); it is also the only option that treats cells of spherical geometry as well as slab geometry.

NOT = 7 — Same as NOT = 6, except that the cross-section homogenization is flux weighting with μ-weighting; considered preferable to NOT = 6 or 9 if the system in which the homogenized cross sections are to be used is a slab. Should not be selected unless IG = 1.

NOT = 8 — Same as NOT = 5, except that the homogenized cross sections are bilinearly weighted with μ-weighting; considered preferable to NOT = 5 or 10 if the system in which the cross sections are to be used is a slab. Select only with IG = 1.

NOT = 9 — Same as NOT = 6, except that the homogenized cross sections are flux weighted with $\sqrt{1 - \mu^2}$ weighting; considered preferable to NOT = 6 or 7 for systems with leakage predominantly parallel to the plates. Select only with IG = 1.

NOT = 10 — Same as NOT = 5, except that the homogenized cross sections are bilinearly weighted with $\sqrt{1 - \mu^2}$ weighting; considered preferable to NOT = 5 or 8 for systems with leakage predominantly parallel to the plates. Select only with IG = 1.

*Includes cylinders with roughly equal height and diameter.

NOT = 11-16 Same as NOT - 6, except that no flux or adjoint calculation is done. These values are assumed to be stored in memory from an earlier problem. Also, the angular flux print is suppressed.

If NOT is negative, a decimal punch-out of the converged fission density is obtained to be used in subsequent cases.

The final item of output in each case is the computer time required.

VII. ESTIMATE OF RUNNING TIME

The running time for a particular problem is almost completely spent doing two things: (1) setting up the matrix, and (2) iterating. Estimating the time involved in either phase is far from an exact science. Figures 6 and 7 are intended to serve only as a rough guide. Often the user will be better off running a sample problem and estimating from that, using the following approximate guidelines: Running time increases linearly with the number of groups, linearly with the number of spatial points, and as the square of the angular order. Regarding the use of Fig. 6, the minimum number of iterations in TESS is 2. The maximum number for convergence is strongly problem-dependent, but for convergence criterion EPS1 equal to 10^{-4}, the code seldom requires more than 15 iterations. If one is also doing an adjoint calculation as part of the problem, the matrix setup time for the adjoint will be considerably less than for the real flux calculation, which is always done first. However, if the problem is solely an adjoint calculation, the time to set up the matrix is the same as for a real flux calculation. Of course, because the adjoint is usually a flatter function of space, the adjoint calculation requires somewhat fewer iterations than the corresponding real flux problem, assuming that both are initiated with a flat source guess.

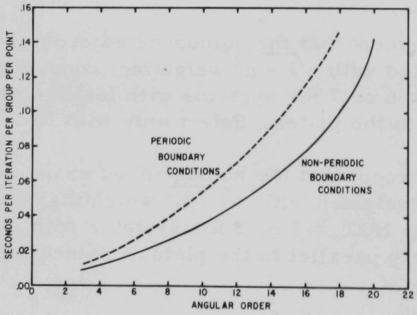


Fig. 6. Iteration Time in TESS vs S_n Angular Order

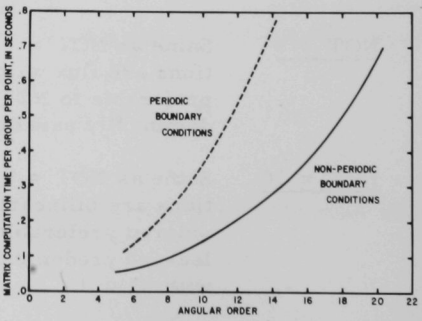


Fig. 7. Matrix Computation Time in TESS vs S_n Angular Order

APPENDIX A

Floating-point Addresses of Cross-section Matrices

1. First-word Addresses

Material Number	SIGT	SIGS	SIGS1	VUSIG	CHI	STR	STR1	VINV
1	50	700	1350	2000	2650	3300	9150	9775
2	76	726	1376	2026	2676	3534	9175	9801
3	102	752	1402	2052	2702	3768	9200	9827
4	128	778	1428	2078	2728	4002	9225	9853
5	154	804	1454	2104	2754	4236	9250	9879
6	180	830	1480	2130	2780	4470	9275	9905
7	206	856	1506	2156	2806	4704	9300	9931
8	232	882	1532	2182	2832	4938	9325	9957
9	258	908	1558	2208	2858	5172	9350	9983
10	284	934	1584	2234	2884	5406	9375	10009
11	310	960	1610	2260	2910	5640	9400	10035
12	336	986	1636	2286	2936	5874	9425	10061
13	362	1012	1662	2312	2962	6108	9450	10087
14	388	1038	1688	2338	2988	6342	9475	10113
15	414	1064	1714	2364	3014	6576	9500	10139
16	440	1090	1740	2390	3040	6810	9525	10165
17	466	1116	1766	2416	3066	7044	9550	10191
18	492	1142	1792	2442	3092	7278	9575	10217
19	518	1168	1818	2468	3118	7512	9600	10243
20	544	1194	1844	2494	3144	7746	9625	10269
21	570	1220	1870	2520	3170	7980	9650	10295
22	596	1246	1896	2546	3196	8214	9675	10321
23	622	1272	1922	2572	3222	8448	9700	10347
24	648	1298	1948	2598	3248	8682	9725	10373
25	674	1324	1974	2624	3274	8916	9750	10399

2. First-word Addresses of STR Scattering Transfer Matrix

Material Number	Down 1	Down 2	Down 3	Down 4	Down 5	Down 6	Down 7	Down 8	Down 9	Down 10	Down 11	Down 12
1	3300	3325	3349	3372	3394	3415	3435	3454	3472	3489	3505	3520
2	3534	3559	3583	3606	3628	3649	3669	3688	3706	3723	3739	3754
3	3768	3793	3817	3840	3862	3883	3903	3922	3940	3957	3973	3988
4	4002	4027	4051	4074	4096	4117	4137	4156	4174	4191	4207	4222
5	4236	4261	4285	4308	4330	4351	4371	4390	4408	4425	4441	4456
6	4470	4495	4519	4542	4564	4585	4605	4624	4642	4659	4675	4690
7	4704	4729	4753	4776	4798	4819	4839	4858	4876	4893	4909	4924
8	4938	4963	4987	5010	5032	5053	5073	5092	5110	5127	5143	5158
9	5172	5197	5221	5244	5266	5287	5307	5326	5344	5361	5377	5392
10	5406	5431	5455	5478	5500	5521	5541	5560	5578	5595	5611	5626
11	5640	5665	5689	5712	5734	5755	5775	5794	5812	5829	5845	5860
12	5874	5899	5923	5946	5968	5989	6009	6028	6046	6063	6079	6094
13	6108	6133	6157	6180	6202	6223	6243	6262	6280	6297	6313	6328
14	6342	6367	6391	6414	6436	6457	6477	6496	6514	6531	6547	6562
15	6576	6601	6625	6648	6670	6691	6711	6730	6748	6765	6781	6796
16	6810	6835	6859	6882	6904	6925	6945	6964	6982	6999	7015	7030
17	7040	7069	7093	7116	7138	7159	7179	7198	7216	7233	7249	7264
18	7278	7303	7327	7350	7372	7393	7413	7432	7450	7467	7483	7498
19	7512	7537	7561	7584	7606	7627	7647	7666	7684	7701	7717	7732
20	7746	7771	7795	7818	7840	7861	7881	7900	7918	7935	7951	7966
21	7980	8005	8029	8052	8074	8095	8115	8134	8152	8169	8185	8200
22	8214	8239	8263	8286	8308	8329	8349	8368	8386	8403	8419	8434
23	8448	8473	8497	8520	8542	8563	8583	8602	8620	8637	8653	8668
24	8682	8707	8731	8754	8776	8797	8817	8836	8854	8871	8887	8902
25	8916	8941	8965	8988	9010	9031	9051	9070	9088	9105	9121	9136

APPENDIX B

TRIM Angles for Use in TESS

Meneghetti[8,9] has derived optimum angles for thin binary slab cells with source and nonsource regions of various thickness. These are tabulated here for use in TESS.

Following the notation of Meneghetti, A and B are the thicknesses of the two cell regions in mean free paths, and

$$\alpha = \frac{AB}{A + B}.$$

Meneghetti recommends[9] if the use of TRIM angles is indicated, choose from among the listed n-approximations, corresponding to the α nearest to $AB/(A+B)$, an n-approximation that has a minimum of at least one value of $\mu < AB/(A+B)$.

Table B.1 lists the TRIM angles for various values of α.

TABLE B.1. Cosines of TRIM Angles

α	n	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7
	4	0.0	0.0261	1.0				
	6	0.0	0.0098	0.0675	1.0			
0.005	8	0.0	0.0057	0.0262	0.1099	1.0		
	10	0.0	0.0039	0.0146	0.0461	0.1489	1.0	
	12	0.0	0.0030	0.0097	0.0261	0.0675	0.1841	1.0
	4	0.0	0.0317	1.0				
	6	0.0	0.0125	0.0776	1.0			
0.0075	8	0.0	0.0074	0.0317	0.1226	1.0		
	10	0.0	0.0052	0.0183	0.0542	0.1634	1.0	
	12	0.0	0.0040	0.0125	0.0317	0.0775	0.1994	1.0
	4	0.0	0.0364	1.0				
	6	0.0	0.0149	0.0856	1.0			
0.01	8	0.0	0.0090	0.0364	0.1327	1.0		
	10	0.0	0.0064	0.0215	0.0608	0.1746	1.0	
	12	0.0	0.0050	0.0149	0.0364	0.0856	0.2114	1.0
	4	0.0	0.0505	1.0				
	6	0.0	0.0224	0.1084	1.0			
0.02	8	0.0	0.0141	0.0505	0.1604	1.0		
	10	0.0	0.0103	0.0314	0.0798	0.2050	1.0	
	12	0.0	0.0081	0.0225	0.0505	0.1084	0.2432	1.0

TABLE B.1 (Contd.)

α	n	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7
	4	0.0	0.0609	1.0				
	6	0.0	0.0284	0.1244	1.0			
0.03	8	0.0	0.0182	0.0608	0.1790	1.0		
	10	0.0	0.0134	0.0389	0.0935	0.2252	1.0	
	12	0.0	0.0106	0.0283	0.0608	0.1243	0.2644	1.0
	4	0.0	0.0693	1.0				
	6	0.0	0.0333	0.1369	1.0			
0.04	8	0.0	0.0217	0.0693	0.1936	1.0		
	10	0.0	0.0161	0.0452	0.1043	0.2407	1.0	
	12	0.0	0.0128	0.0333	0.0693	0.1369	0.2802	1.0
	4	0.0	0.0767	1.0				
	6	0.0	0.0377	0.1474	1.0			
0.05	8	0.0	0.0249	0.0767	0.2058	1.0		
	10	0.0	0.0186	0.0508	0.1137	0.2537	1.0	
	12	0.0	0.0148	0.0377	0.0766	0.1474	0.2934	1.0
	4	0.0	0.0831	1.0				
	6	0.0	0.0417	0.1566	1.0			
0.06	8	0.0	0.0277	0.0831	0.2160	1.0		
	10	0.0	0.0208	0.0557	0.1217	0.2645	1.0	
	12	0.0	0.0167	0.0417	0.0831	0.1566	0.3047	1.0
	4	0.0	0.0889	1.0				
	6	0.0	0.0453	0.1647	1.0			
0.07	8	0.0	0.0303	0.0889	0.2252	1.0		
	10	0.0	0.0229	0.0602	0.1291	0.2744	1.0	
	12	0.0	0.0183	0.0452	0.0888	0.1645	0.3143	1.0
	4	0.0	0.0942	1.0				
	6	0.0	0.0486	0.1720	1.0			
0.08	8	0.0	0.0328	0.0942	0.2334	1.0		
	10	0.0	0.0247	0.0642	0.1354	0.2827	1.0	
	12	0.0	0.0199	0.0486	0.0941	0.1719	0.3231	1.0
	4	0.0	0.0991	1.0				
	6	0.0	0.0517	0.1787	1.0			
0.09	8	0.0	0.0350	0.0990	0.2408	1.0		
	10	0.0	0.0265	0.0680	0.1413	0.2902	1.0	
	12	0.0	0.0214	0.0517	0.0990	0.1787	0.3310	1.0
	4	0.0	0.1036	1.0				
	6	0.0	0.0546	0.1848	1.0			
0.10	8	0.0	0.0372	0.1037	0.2476	1.0		
	10	0.0	0.0282	0.0716	0.1469	0.2974	1.0	
	12	0.0	0.0228	0.0547	0.1037	0.1849	0.3382	1.0

TABLE B.1 (Contd.)

α	n	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7
	4	0.0	0.1226	1.0				
	6	0.0	0.0670	0.2099	1.0			
0.15	8	0.0	0.0463	0.1226	0.2752	1.0		
	10	0.0	0.0355	0.0866	0.1697	0.3259	1.0	
	12	0.0	0.0288	0.0670	0.1225	0.2099	0.3669	1.0
	4	0.0	0.1375	1.0				
	6	0.0	0.0769	0.2292	1.0			
0.20	8	0.0	0.0537	0.1374	0.2961	1.0		
	10	0.0	0.0414	0.0986	0.1875	0.3475	1.0	
	12	0.0	0.0337	0.0769	0.1374	0.2292	0.3881	1.0
	4	0.0	0.1498	1.0				
	6	0.0	0.0853	0.2450	1.0			
0.25	8	0.0	0.0599	0.1497	0.3129	1.0		
	10	0.0	0.0463	0.1084	0.2018	0.3644	1.0	
	12	0.0	0.0378	0.0853	0.1498	0.2450	0.4055	1.0
	4	0.0	0.1603	1.0				
	6	0.0	0.0924	0.2583	1.0			
0.30	8	0.0	0.0654	0.1604	0.3272	1.0		
	10	0.0	0.0507	0.1170	0.2142	0.3789	1.0	
	12	0.0	0.0414	0.0924	0.1603	0.2583	0.4197	1.0
	4	0.0	0.1696	1.0				
	6	0.0	0.0988	0.2698	1.0			
0.35	8	0.0	0.0701	0.1696	0.3394	1.0		
	10	0.0	0.0544	0.1245	0.2248	0.3911	1.0	
	12	0.0	0.0446	0.0988	0.1696	0.2698	0.4319	1.0
	4	0.0	0.1776	1.0				
	6	0.0	0.1043	0.2799	1.0			
0.40	8	0.0	0.0743	0.1776	0.3498	1.0		
	10	0.0	0.0578	0.1312	0.2343	0.4019	1.0	
	12	0.0	0.0474	0.1043	0.1776	0.2799	0.4426	1.0
	4	0.0	0.1849	1.0				
	6	0.0	0.1094	0.2890	1.0			
0.45	8	0.0	0.0781	0.1849	0.3593	1.0		
	10	0.0	0.0609	0.1372	0.2427	0.4116	1.0	
	12	0.0	0.0499	0.1093	0.1849	0.2889	0.4521	1.0
	4	0.0	0.1915	1.0				
	6	0.0	0.1139	0.2970	1.0			
0.50	8	0.0	0.0815	0.1915	0.3679	1.0		
	10	0.0	0.0637	0.1426	0.2503	0.4202	1.0	
	12	0.0	0.0522	0.1139	0.1915	0.2970	0.4607	1.0

APPENDIX C

Choice of Angles for Use in TESS

1. Discussion

The choice of optimum angles (and their corresponding weights, if a quadrature scheme is used) for a transport code depends, of course, on the problem to be solved. This should be kept in mind when evaluating the content of this appendix. The calculational model used here for comparison of convergence obtained with different choices of angles is one of the models presented in Ref. 8. Briefly, it is a binary-cell, one-energy-group problem with a unit source (but no fission) in one region, which is A mean free paths thick, and no source in the second region, which is B mean free paths thick. The calculation describes half of each region with a reflective boundary condition at both boundaries. (See Fig. C.1.)

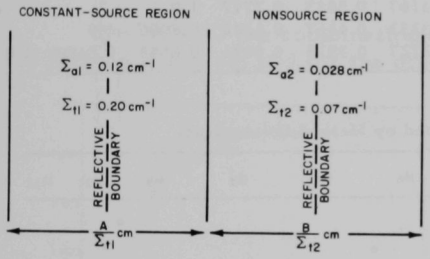


Fig. C.1. Calculational Model

Three methods of choosing angles were used for this comparison study. The Meneghetti TRIM angles of Appendix B were one choice. They are derived to be optimum angles for thin binary slab cells. It is well known that in slab geometry the angular flux at an interface can have a very large gradient as the angle approaches $\pi/2$. Thus, the TRIM angles between 0 and 90° are concentrated near 90°. However, even for fairly thin binary slab cells, the flux at the center of a region tends to be much less anisotropic than at the interface. This suggests that some sort of compromise should be made with the angles a bit more evenly spaced (in terms of cos θ). To this end, the following two methods were devised (n is the order of the S_n calculation and $\mu_i = \cos\theta_i$):

Method I

$$\mu_1 = 0.0; \quad \mu_{i+1} = \sum_{k=1}^{i} k \Big/ \sum_{j=1}^{n/2} j. \tag{C.1}$$

Method II

$$\mu_1 = 0.0; \quad \mu_{i+1} = \sum_{k=1}^{i} 2^{k-1} \Big/ \sum_{j=1}^{n/2} 2^{j-1}. \tag{C.2}$$

Method II results in a finer μ spacing close to $\mu = 0$ compared to Method I, yet gives much more evenly spaced values of μ than TRIM. Tables C.1 and C.2 show the positive values of μ obtained by these methods.

TABLE C.1. Cosines Obtained by Method I (see text)

n	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}	μ_{11}
2	0.0	1.0									
4	0.0	0.3333	1.0								
6	0.0	0.1667	0.5	1.0							
8	0.0	0.1	0.3	0.6	1.0						
10	0.0	0.0667	0.2	0.4	0.6667	1.0					
12	0.0	0.0476	0.1429	0.2857	0.4762	0.7143	1.0				
14	0.0	0.0357	0.1071	0.2143	0.3571	0.5357	0.7500	1.0			
16	0.0	0.0278	0.0833	0.1667	0.2778	0.4167	0.5833	0.7777	1.0		
18	0.0	0.0222	0.0667	0.1333	0.2222	0.3333	0.4667	0.6222	0.8000	1.0	
20	0.0	0.0182	0.0545	0.1091	0.1818	0.2727	0.3818	0.5091	0.6545	0.8182	1.0

TABLE C.2. Cosines Obtained by Method II (see text)

n	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}	μ_{11}
2	0.0	1.0									
4	0.0	0.3333	1.0								
6	0.0	0.1429	0.4286	1.0							
8	0.0	0.0667	0.2000	0.4667	1.0						
10	0.0	0.0323	0.0967	0.2258	0.4839	1.0					
12	0.0	0.0159	0.0476	0.1111	0.2381	0.4921	1.0				
14	0.0	0.0079	0.0236	0.0551	0.1181	0.2441	0.4961	1.0			
16	0.0	0.0039	0.0118	0.0275	0.0588	0.1216	0.2471	0.4980	1.0		
18	0.0	0.0020	0.0059	0.0137	0.0294	0.0607	0.1233	0.2485	0.4990	1.0	
20	0.0	0.0010	0.0029	0.0068	0.0147	0.0303	0.0616	0.1241	0.2493	0.5	1.0

One other combination of angles was formed for an S-10 calculation by using the angles obtained by Method II for S-8 and adding one cosine a factor of 10 smaller than the smallest nonzero value in the S-8 set. These are shown in Table C.3.

TABLE C.3. Combination Cosines

n	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
10	0.0	0.0067	0.0667	0.2	0.5	1.0

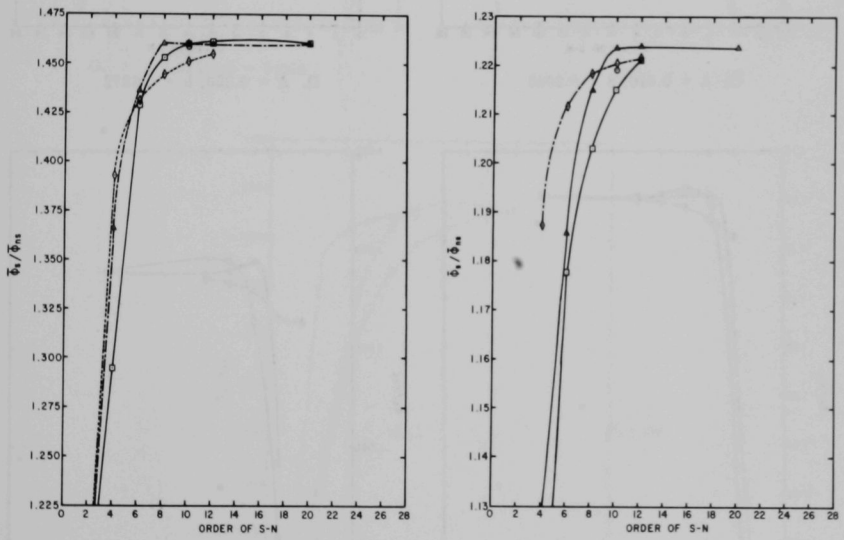
2. Results

Even though these cases have no fission, the results are indicative of the solutions that would be obtained in realistic slab-geometry multigroup cell calculations with fission in one region. The comparison

parameter used is the ratio of average flux in the source region to the average flux in the nonsource region. Presumably, if this ratio is correct, that is, if increasing the number of angles (or space points) does not change the ratio, then other quantities such as eigenvalue and reaction rates are correct. The ratio of $\overline{\varphi}_s$ to $\overline{\varphi}_{ns}$ is plotted for each case considered, as a function of the number of angular intervals in μ-space, in Fig. C.2.

For A << B (Figs. C.2A-C) and A >> B (Figs. C.2G-I), the Method II angles give the "correct" solution with fewest angles, even for the thinnest case, where AB/(A+B) = 0.015. Also, for A < B, Method II angles appear to give a more nearly correct solution sooner than Method I. In most of the cases, the TRIM angles are better for small values of n (4 and 6), but they do not converge as rapidly. This indicates that it is important, even for very thin cells, to have some fairly large values of μ.

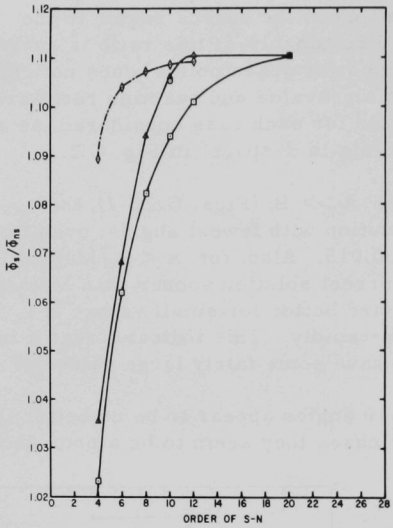
For A < B, the combination S-10 angles appear to be no better than the Method II angles, and for the other cases they seem to be a poor choice.



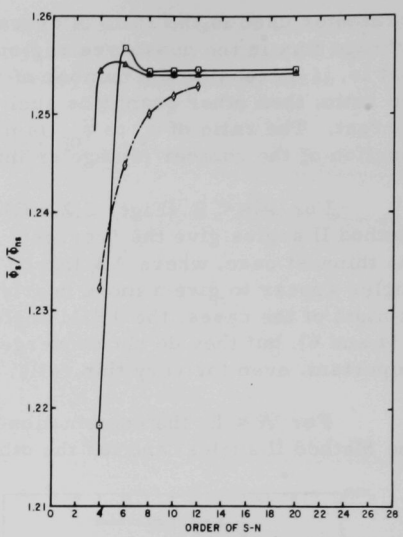
A. A = 0.064, B = 0.8192

B. A = 0.032, B = 0.4096

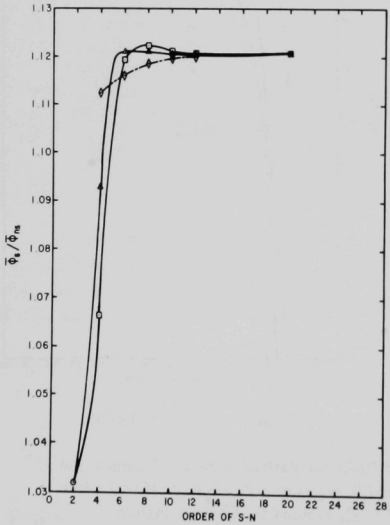
Fig. C.2. Degree of Convergence vs Angular Order for Various Choices of Source and Nonsource Region Thicknesses (A and B, respectively). In units of mean free path; ◇ TRIM angles; □ Method I; △ Method II; ○ combination.



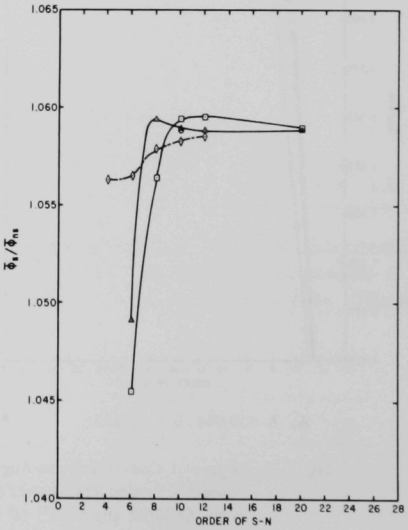
C. A = 0.016, B = 0.2048



D. A = 0.256, B = 0.6272

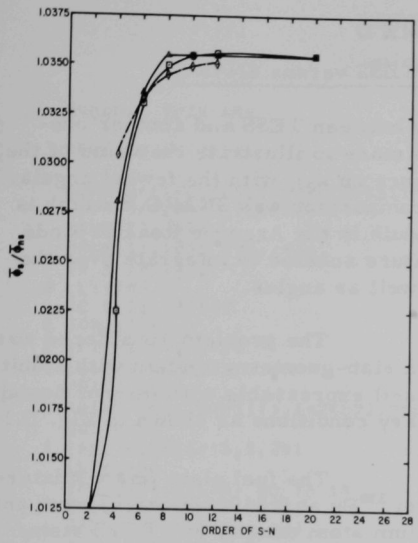


E. A = 0.128, B = 0.3136

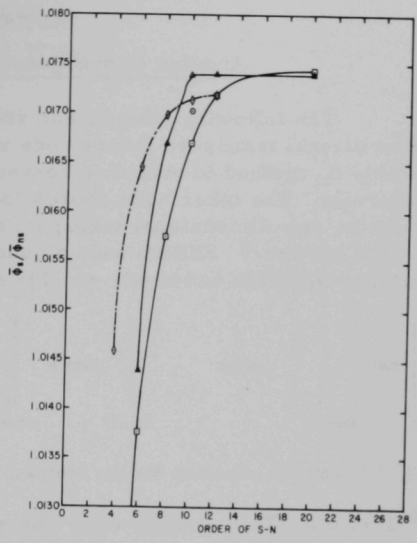


F. A = 0.064, B = 0.1568

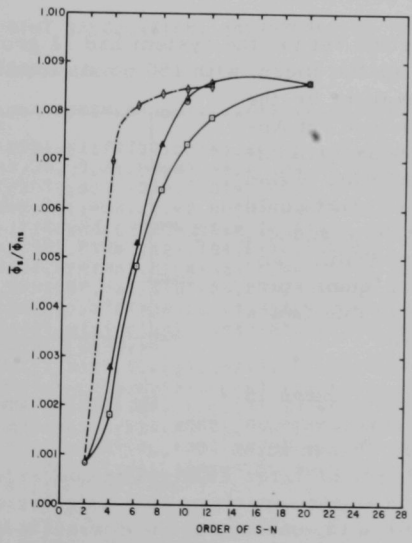
Fig. C.2 (Contd.)



G. A = 0.8192, B = 0.064



H. A = 0.4096, B = 0.032



I. A = 0.2048, B = 0.016

Fig. C.2 (Contd.)

APPENDIX D

Angular Convergence: TESS versus SNARG

The following comparison study between TESS and another one-dimensional transport-theory code was made to illustrate the value of the double S_n method in obtaining convergence on k_{eff} with the fewest angular intervals. The other code chosen for comparison was SNARG,[10] which is also the one-dimensional transport module in the Argonne Reactor Code (ARC) system.[11] SNARG uses a quadrature scheme to integrate over the angular variable and needs weights as well as angles.

Fig. D.1. Unit Cell with Mirror Boundary Conditions

The problem considered was a slab-geometry system with a unit cell expressable with mirror boundary conditions as shown in Fig. D.1.

The fuel plate was a mixture of ^{239}Pu at 0.0102685 atom/b-cm and ^{238}U at 0.0377315 atom/b-cm. The sodium atom density was 0.022 atom/b-cm and the carbon density was 0.072 atom/b-cm. The core was 38.5 cm thick (with a fuel plate at the center) and had a 20-cm-thick ^{238}U blanket (0.048 atom/b-cm) at each end.

The cross-section set for the system had 12 groups, and both codes were given the same spatial mesh, with 150 points total. The TESS angles were arbitrarily chosen to be the angles generated by Method I of Appendix C. Considerable experimentation was done with SNARG to find a set of angles and weights that would give reasonably quick convergence. Both single Gauss quadrature and modified single Gauss quadrature were rejected before double Gauss quadrature was chosen.

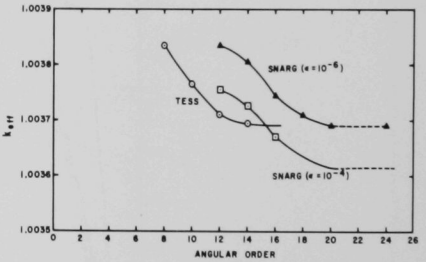


Fig. D.2. TESS and SNARG Convergence, as a Function of Angular Order

The results are shown in Fig. D.2. Pointwise convergence on the fission source was chosen with a convergence criterion of 10^{-4}, for TESS. The convergence in SNARG is on the eigenvalue, so it is not surprising that a tighter convergence criterion of 10^{-6} was required to converge to the correct k-effective. The results show that TESS was converged at S-14 while SNARG required S-20.

APPENDIX E

Listing of TESS

```
      PROGRAM TESS 69A
C
C     REVISED EDITION
C
      COMMON/A1/LL(250),COM1(25912)
      COMMON/A2/COM2(10797)
      BANK(0),/A2/
      BANK(1),TESS 69A,/A1/
      DO 8 I=1,250
    8 LL(I)=0
      DO 9 I=1,25912
    9 COM1(I)=0.0
    2 IF(LL(22))3,4,4
    3 LL(22)=-LL(22)
      PRINT 500
      PRINT 501,(COM1(I),I=25873,25884)
      GO TO 5
    4 CALL OVERLAY(1,0,29)
      LL(16)=0
      IF(LL(7).GT.10) GO TO 10
    5 CALL OVERLAY(2,0,29)
    6 CALL OVERLAY(3,0,29)
      IF(LL(197))4,4,7
    7 CALL OVERLAY(4,0,29)
      IF(LL(197))2,2,10
   10 CALL OVERLAY(5,0,29)
      GO TO 2
  500 FORMAT(*1 ADJOINT CALCULATION, CORRESPONDING TO PRECEEDING FLUX CA
     1LC*)
  501 FORMAT(1H0,12A6////)
      END TESS 69A
C*********************OVERLAY 1*************************************
      PROGRAM LINK1
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGBM1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),DE),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      DIMENSION ALL(700)
      COMMON/2/ALL(700),IREGNO(20),NIG(20),NEG(20)
      COMMON/[illegible],SEARCH,SCHECK
```

```
      COMMON/A2/EE(1300)
      DIMENSION SIGFIS(26,25),SIGFIV(650),SIGCAP(26,25),SIGCAV(650)
      EQUIVALENCE (EE(1),SIGFIV,SIGFIS),(EE(651),SIGCAV,SIGCAP)
      BANK(0),/2/,/A2/,INPR,MIXX
      EQUIVALENCE (ILL(1),ALL(1))
C     THIS IS THE MAIN INPUT CHAIN
      IF(LGO)2000,1,2
 2000 CALL SEARCH
      IF(JSP-1)83,83,65
    1 FAC = 1.0
      LGOC = 0
C     SET BOUNDARY CONDITIONS TO REFLECTIVE (LEFT) AND NO REENTRANT
C     CURRENT (RIGHT), IF OTHER CONDITIONS ARE DESIRED READ IN ALPHAS
      DO 9669 IJK=1,26
      ALPHA(1,IJK)=1.0
 9669 ALPHA(2,IJK)=1.0
      LFREG=1
      N=4
      SEN=1.
      EPS3=.003
      ITOUT=50
      MUTEST=1
    2 IF(LGO.GT.0) LGOC = 1
      IF(EOF,60) 80,4004
 4004 LGO = 1
      REWIND 1
      SEARCON=0.
      NCE=0
      IT=0
      READ 527,(PROBT(J),J=1,12),KTEST
  527 FORMAT(12A6,I2)
      IF(EOF,60) 80,4005
 4005 IF(KTEST-99)5142,5143,5142
 5143 DO 5144 IJK=1,223
 5144 LL(IJK)=0
      DO 5145 IJK=1,17480
 5145 E(IJK)=0.0
      LFREG=1
      N=4
      MIK=0
      ITOUT=50
      MUTEST=1
      EPS3=.003
      SEN=1.
      FAC=1.
      DO 9559 IJK=1,26
      ALPHA(1,IJK)=1.
 9559 ALPHA(2,IJK)=1.0
 5142 PRINT 533,(PROBT(J),J=1,12),KTEST
      IFLAG=XABSF(IG)
      ISAVE=0
      IF(KTEST.EQ.88) ISAVE = 1
      PRINT 521
      TIMEBEG=TIMELEFT(TIMEBEG)
   14 READ 501,NR,LC,NA,(NN(I),I=1,20)
      IF(NR) 100,100,15
   15 IF(NR-20)22,22,100
   22 IF(NA) 100,100,16
   16 IF(NA-250)23,23,100
   23 DO 17 I=1,NR
      J=I+NA-1
   17 LL(J)=NN(I)
```

```
C       END OF FIXED POINT DATA
        IF(LC-1) 14,202,202
  202   IGOM=XABSF(IG)
        IF(IFLAG.EQ.IGOM) GO TO 18
        IF(IGOM.NE.3) GO TO 18
        DO 203 IJK=1,26
        ALPHA(1,IJK)=1.0
  203   ALPHA(2,IJK)=0.0
   18   IF(LPG-1) 700,700,701
  700   DO 702K=1,189
  702   POWR1(K) = 0.0
  701   READ 502,NOF,LF,NAF,(CC(I),I=1,5)
        IF(NOF) 101,101,19
   19   IF(NOF-5)24,24,101
   24   IF(NAF) 101,101,20
   20   IF(NAF-17480)25,25,101
   25   DO 21 I=1,NOF
        J=I+NAF-1
   21   E(J)=CC(I)
        IF(LF-1)701,26,26
   26   IF(MMIX) 200,201,200
  200   READ 525,ISET,(MATNO(I),I=1,MMIX)
 1001   IFL=0
        IF(NRATE)1002,201,1007
 1002   NRATE=-NRATE
        IFL=1
 1007   READ 526,(LRATE(I),I=1,NRATE)
  526   FORMAT(12A6)
        IF(IFL.EQ.0) GO TO 201
 1004   READ 502,NOF,LF,NAF,(CC(I),I=1,5)
        IF(NOF)101,101,1005
 1005   IF(NOF-5)1006,1006,101
 1006   IF(NAF)101,101,1017
 1017   IF(NAF-1300)1018,1018,101
 1018   DO 1019 I=1,NOF
        J=I+NAF-1
 1019   EE(J)=CC(I)
        IF(LF-1)1004,201,201
  201   OUTCON=NOT
        NOT=XABSF(NOT)
        IF((IG.EQ.1).OR.(IG.EQ.-1))PRINT 529
        IF((IG.EQ.3).OR.(IG.EQ.-3))PRINT 530
        IF((IG.NE.1).AND.(IG.NE.3).AND.(IG.NE.-1).AND.(IG.NE.-3))NCE=1
        IF(NCE.EQ.1) PRINT 531
        IF(MADJ.EQ.2) MADJ=-MADJ
        IF(NOT.EQ.6.OR.NOT.EQ.7.OR.NOT.EQ.9) MADJ=0
        IF(NOT.EQ.5.OR.NOT.EQ.8.OR.NOT.EQ.10) MADJ=-2
        IF(MADJ.EQ.-2)PRINT 522
        IF(MADJ.EQ.-1.OR.MADJ.EQ.0)PRINT 523
        IF(MADJ.GT.0)PRINT 524
        IF(NOT.LE.10) GO TO 104
        PRINT 532
        GO TO 83
  104   IF(MAX-150)28,28,27
   27   PRINT 505
        NCE=1
        GO TO 30
   28   IF(MAX-2) 29,29,30
C       LESS THAN 3 POINTS - ERROR
   29   PRINT 506
        NCE=1
   30   IF(JMAX-40) 32,32,31
```

```
C       MORE  THAN  40  REGIONS = ERROR
   31 PRINT 507
      NCE=1
      GO TO 34
   32 IF(JMAX) 33,33,34
C       NO  REGIONS - ERROR
   33 PRINT 508
      NCE=1
   34 IF(NGR-26) 36,36,35
   35 PRINT 509
      NCE=1
      GO TO 38
   36 IF(NGR) 37,37,38
C       AT  LEAST  ONE  GROUP
   37 PRINT 510
      NCE=1
   38 IF(NDS-12)3000,3000,39
 3000 IF(NDS-NGR)3001,39,39
 3001 IF(NPS-1)3002,3002,39
 3002 IF(NPS-NGR)40,39,39
   39 PRINT 511
      NCE=1
   40 DO 42 I=1,JMAX
      IF(MIR(I)) 41,41,42
C       EVERY REGION MUST HAVE A MATERIAL  SPECIFIED
   41 PRINT 512
      NCE=1
      GO TO 43
   42 CONTINUE
   43 DO 300 I=1,JMAX
      IF(MIR(I)-25)300,300,301
  301 PRINT 520
      NCE=1
      GO TO 302
  300 CONTINUE
  302 IF(II(1)-1) 44,44,95
   95 IF(JMAX-1) 44,46,600
  600 IF(MAX.NE.II(JMAX)) GO TO 44
      M7=JMAX-1
      DO 45 I=1,M7
C       REGION UPPER POINT NUMBERS MUST BE MONOTONICALLY  INCREASING
      IF(II(I+1)-II(I)) 44,44,45
   45 CONTINUE
      GO TO 46
   44 PRINT 513
      NCE=1
   46 DO 48 I=1,JMAX
      IF(DELR(I)) 47,47,48
C       REGION DELTA R,S  MUST BE  POSITIVE
   47 PRINT 514
      NCE=1
      GO TO 49
   48 CONTINUE
      IF(NCE) 83,83,49
   83 CALL MIXX
      IF(NOT.GT.10) GO TO 69
   49 DO 50 J=1,JMAX
      K=MIR(J)
      DO 50 I=1,NGR
C       CHECK TO DETERMINE IF EVERYTHING IS VOID
      IF (SIGT(I,K)) 52,50,52
   50 CONTINUE
```

```
   51 PRINT 515
      NCE=1
   52 IF(LL(250),NE,0) GO TO 99
      IF(IBUK)88,99,88
   88 IF(IBUK-5)1023,1024,1024
 1024 DO 1025 I=1,NGR
      SUM1=SUM2=0.
      DO 720 J=1,JMAX
      IF(J.EQ.1) DELX=(II(1)-1)*DELR(J)
      IF(J.GT.1) DELX=(II(J)-II(J-1))*DELR(J)
      SUM1=SUM1+SIGT(I,MIR(J))*DELX
  720 SUM2=SUM2+DELX
      SITOT8=SUM1/SUM2
 1025 BUCK(I+26,1)=SITOT8/(SITOT8+BUCK(I,1))
      GO TO 99
 1023 DO 1020 J=1,25
      DO 1021 K=1,JMAX
      IF(MIR(K)-J)1021,1022,1021
 1021 CONTINUE
      GO TO 1020
 1022 DO 89 I=1,NGR
      IF(SIGT(I,J))1032,89,1032
 1032 IF(IBUK-2)2008,1010,1011
 2008 SIGT(I,J)=SIGT(I,J)+BUCK(1,1)/(3.0*SIGT(I,J))
      GO TO89
 1010 SIGT(I,J)=SIGT(I,J)+BUCK(J,1)/(3.0*SIGT(I,J))
      GO TO 89
 1011 IF(IBUK-4)1014,1015,1015
 1014 SIGT(I,J)=SIGT(I,J)+BUCK(I,1)/(3.0*SIGT(I,J))
      GO TO 89
 1015 SIGT(I,J)=SIGT(I,J)+BUCK(I,J)/(3.0*SIGT(I,J))
   89 CONTINUE
 1020 CONTINUE
   99 AN=N
      J1=N+2
      J2=J1/2
      IF((IG.EQ.3).OR.(IG.EQ.-3))J1=N+1
      EMU(1)=-1.
      EMU(J2)=0.
      EMU(J2+1)=0.0
      EMU(J1)=1.
      J3=J2-1
      IF(N-20)400,55,63
  400 IF(N-18)401,55,63
  401 IF(N-16)402,55,63
  402 IF(N-14)403,55,63
  403 IF(N-12)404,55,63
  404 IF(N-10)405,55,63
  405 IF(N-8)406,55,63
  406 IF(N-6)407,55,63
  407 IF(N-4)408,55,63
  408 IF(N-2)63,76,63
   55 IF(MUTEST-2)56,58,60
C
   56 AJ=2./AN
      DO 57 J=2,J3
C     EQUAL INTERVALS IN COSINE  THETA
      M1=J1-J+1
      EMU(J)=EMU(J-1)+AJ
   57 EMU(M1)=-EMU(J)
      GO TO 76
   58 DO 59 J=2,J3
```

```
C       EQUAL INTERVALS IN THETA
        M1=J1-J+1
        AJ=J-1
        AJ=AJ/AN
        EMU(J)=-COSF(3.14159265*AJ)
   59   EMU(M1)=-EMU(J)
        GO TO 76
C       INTERVALS READ  IN
   60   DO 61  J=2,J3
        M1=J1-J+1
        IF(EMU(J)*EMU(M1)) 63,61,63
C       CHECK FOR CONSISTENCY IF ANGULAR DATA IS INPUT
   61   CONTINUE
        DO 62 J=2,J3
        IF(EMU(J)-EMU(1)) 63,63,91
   91   IF(EMU(J)) 62,63,63
   62   CONTINUE
        GO TO 76
   63   PRINT 516
        NCE=1
        NCE=1
   76   IF(NCE) 65,65,80
   65   XPT(1)=XIN
        J=1
        DO 67 I=2,MAX
        IF(I-II(J)) 67,67,66
   66   J=J+1
   67   XPT(I)=XPT(I-1)+DELR(J)
        CALL SCHECK
        IF(NRATE)1027,1027,1030
 1030   IF(LGOC)1027,1028,1027
 1028   IF(UNIT,1)1028,1029
 1029   BUFFER OUT (1,1)(EE(1),EE(650))
        BUFFER OUT (1,1)(EE(651),EE(1300))
 1027   S1=IG
        IF(NCE) 68,68,80
   68   IF(IDP-1) 69,70,70
   69   IF(LGO)70,5068,5068
 5068   CALL INPR
   70   IF(LL(250),NE,0) GO TO 703
        RETURN
  703   PRINT 528
        MMIX=0
        GO TO 2
   80   CALL EXIT
  100   PRINT 503
        PRINT 501,NR,LC,NA,(NN(I),I=1,21)
        CALL EXIT
  101   PRINT 504
        PRINT 502,NOF,LF,NAF,(CC(I),I=1,6)
        CALL EXIT
  500   FORMAT(70H
       1                    ,I2)
  501   FORMAT(I2,I4,I6,20I3)
  502   FORMAT(I2,I4,I6,5E12.6)
  503   FORMAT (44H ERROR,ADDRESS FORMAT FOR FIXED POINT DATA  )
  504   FORMAT (48H ERROR,ADDRESS FORMAT FOR FLOATING POINT DATA   )
  505   FORMAT (34H ERROR,NUMBER OF POINTS TOO LARGE )
  506   FORMAT (34H ERROR,NUMBER OF POINTS TOO SMALL )
  507   FORMAT (35H ERROR,NUMBER OF REGIONS TOO LARGE )
  508   FORMAT (35H ERROR,NUMBER OF REGIONS IS  ZERO  )
  509   FORMAT (34H ERROR,NUMBER OF GROUPS TOO LARGE )
```

```
510 FORMAT (34H ERROR,NUMBER OF GROUPS IS  ZERO  )
511 FORMAT(47H ERROR,NUMBER OF DOWN-OR UP-SCATTERS TOO LARGE )
512 FORMAT (40H ERROR,ZERO OR NEGATIVE MATERIAL NUMBER )
513 FORMAT (87H ERROR,UPPER REGION BOUNDARY POINT NON-INCREASING OR
   1II(1)IS LESS THAN OR EQUAL TO 1 )
514 FORMAT (20H ERROR,ZERO DELTA R )
515 FORMAT (31H ERROR,NO NON-ZERO SIGMA TOTAL )
516 FORMAT (37H ERROR,   INCONSISTENT ANGULAR INPUT )
518 FORMAT(38H1  CHECKOUT-RADII-POWER-FISSION VOLUME)
519 FORMAT( 9(1X,E12.5))
520 FORMAT(33H MATERIAL NUMBER GREATER THAN 25 )
521 FORMAT(17H0PROGRAM TESS 69A)
522 FORMAT(* FLUX CALCULATION OF DUAL FLUX-ADJOINT OPTION*)
523 FORMAT(31H ****** FLUX CALCULATION ******)
524 FORMAT(34H ****** ADJOINT CALCULATION ******)
525 FORMAT(A5,1X,(11A6))
528 FORMAT(*0THIS PROBLEM SKIPPED*)
529 FORMAT(*0     SLAB GEOMETRY*)
530 FORMAT(*     SPHERICAL GEOMETRY*)
531 FORMAT(* ERROR, GEOMETRY SPECIFIED INCORRECTLY*)
532 FORMAT(83H * CROSS SECTION HOMOGENIZATION USING FLUXES AND ADJOINT
   1S FROM PRECEEDING PROBLEM *)
533 FORMAT(1H1,12A6,I2)
    END LINK 1
    SUBROUTINE SEARCH
    COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
   1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
   2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
   3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
   4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
   5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
    DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
   1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
   2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
   3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
   426,25),BETA(2,26),LRATE(25)
    EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
   1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
   2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
   3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
   420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),BG),(LL
   5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
   6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
   7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
    EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
   14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
   2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
   3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
   410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
   5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
   612522),SVM)
    DIMENSION ALL(700)
    COMMON/2/ILL(700),IRECNO(20),NIG(20),NEG(20)
    BANK(1),LINK1,/A1/,SEARCH,SCHECK
    COMMON/A2/EE(1300)
    DIMENSION SIGFIS(26,25),SIGFIV(650),SIGCAP(26,25),SIGCAV(650)
    EQUIVALENCE (EE(1),SIGFIV,SIGFIS),(EE(651),SIGCAV,SIGCAP)
    BANK(0),/2/,/A2/,INPR,MIXX
    EQUIVALENCE (ILL(1),ALL(1))
    NSOS=NSOS
    NFOS=NFOS
    KREG=KREG
```

```
      VXM=VX(3)
      IF(SEARCON)1,1,2
    1 IF(JSP-1)3,3,5
    3 XK(2)=S1
      VX(2)=CONC(NSOS)
      VX(3)=SGES
      CONC(NSOS)=SGES
      SEARCON=1.
      IF(NFOS)69,69,4
   69 RETURN
    4 CONC(NFOS)=CONC(NFOS)-RR*(VX(3)-VX(2))
      GO TO 69
    5 XK(2)=S1
      VX(2)=DELR(KREG)
      VX(3)=SGES
      DELR(KREG)=SGES
      SEARCON=1.
      GO TO 69
    2 XK(3)=S1
      IF(SEARCON-2.)10,11,11
   10 VXI=((XK(3)-SEN)*VX(2)+(SEN-XK(2))*VX(3))/(XK(3)-XK(2))
      IF (VXI) 22,22,21
   22 IF (XK(3)-SEN) 25,30,24
   24 IF (VX(3)-VX(2))27,75,26
   26 VX(3)=VX(2)
      XK(3) = XK(2)
   27 VXI=.2*VX(3)
      GO TO 150
   25 IF (VX(3)-VX(2)) 28,75,29
   28 VX(3)=VX(2)
      XK(3)=XK(2)
   29 VXI=5.*VX(3)
      GO TO 150
   30 VXI=1.01*VX(3)
      GO TO 150
   21 SEARCON=2.
  150 IF(JSP-1)15,15,16
   15 CONC(NSOS)=VXI
   17 IF(NFOS)71,71,18
   18 CONC(NFOS)=CONC(NFOS)-RR*(CONC(NSOS)-VXM)
      GO TO 71
   16 DELR(KREG)=VXI
      GO TO 71
   11 XKG=(XK(1)+XK(2)+ABSF(XK(1)-XK(2)))/2.
      XKL=(XK(1)+XK(2)-ABSF(XK(1)-XK(2)))/2.
   13 XKA=((XK(2)*XK(2)-XK(3)*XK(3))*VX(1)+(XK(3)*XK(3)-XK(1)*XK(1))*VX(
     1      2)+(XK(1)*XK(1)-XK(2)*XK(2))*VX(3))/(2.*((XK(2)-XK(3))*VX(1)+
     2      (XK(3)-XK(1))*VX(2)+(XK(1)-XK(2))*VX(3)))
      IF (XK(3)-SEN)80,12,90
   80 IF(XKG-SEN)83,12,12
   83 IF (XKA-SEN)62,12,12
   90 IF (XKL-SEN)12,12,91
   91 IF (XKA-SEN)12,12,62
   12 AB1=
     1   (SEN-XK(2))*(XK(3)-SEN)/((XK(1)-XK(2))*(XK(3)-XK(1)))
      AB2=(SEN-XK(1))*(XK(3)-SEN)/((XK(1)-XK(2))*(XK(2)-XK(3)))
      AB3=(SEN-XK(1))*(XK(2)-SEN)/((XK(3)-XK(2)) *(XK(1)-XK(3)))
      /(AB1+AB2+AB3)
      IF(VXI)22,22,62,150
   62 SEARCON=1.
      IF(XK(3)-SEN)63,12,64
   63 XK(2)=XKG
```

```
      IF(XK(1)-XKG)10,65,10
   64 XK(2)=XKL
      IF(XK(1)-XKL)10,65,10
   65 VX(2)=VX(1)
      GO TO 10
   71 DO 72 I=1,2
      VX(I)=VX(I+1)
   72 XK(I)=XK(I+1)
      IF(JSP-1) 81,81,82
   81 VX(3)=CONC(NSOS)
      GO TO 69
   82 VX(3)=DELR(KREG)
      GO TO 69
   75 PRINT 500
  500 FORMAT (/86H SEARCH FAILED, TWO CONSECUTIVE SEARCH GUESSES ARE EQU
     1AL,CHECK SECOND GUESS                                          )
      CALL EXIT
      END
      SUBROUTINE INPR
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),DG),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      DIMENSION ALL(700)
      COMMON/2/ILL(700),IRECNO(20),NIG(20),NEG(20)
      BANK(1),LINK1,/A1/,SEARCH,SCHECK
      COMMON/A2/EE(1300)
      DIMENSION SIGFIS(26,25),SIGFIV(650),SIGCAP(26,25),SIGCAV(650)
      EQUIVALENCE (EE(1),SIGFIV,SIGFIS),(EE(651),SIGCAV,SIGCAP)
      BANK(0),/2/,/A2/,INPR,MIXX
      EQUIVALENCE (ILL(1),ALL(1))
      PRINT 500
      PRINT 501,IG,MAX,JMAX,NGR,NDS,MUTEST,N,LPG,LCO,NOT,NMIX,IDP,ITOUT
      PRINT 600,NPS,MMIX,JSP,KREG,NSOS,NFOS
      PRINT 526,IBUK,NRATE,NHGP,NHREG
      PRINT 502,EPS1,XIN,THETA,FAC
      PRINT 601, SGES,SEN,RR
      PRINT 503
C     PRINT OUT REGION DATA
      DO 2 I=1,JMAX
```

```
      K=II(I)
    2 PRINT 504,I,MIR(I),K,DELR(I),XPT(K)
C     PRINT OUT ANGULAR DATA
      PRINT 505
      DO 3 I=1,J1
    3 PRINT 506,I,EMU(I)
C     PRINT OUT MIXTURE DATA, IF ANY
      IF(NMIX)6,6,4
    4 PRINT 507
      DO 5 I=1,NMIX
      IF(CONC(I))22,21,22
   21 IK=6H  MIX.
      GO TO 5
   22 DO 20 J=1,20
      IF(MIX(I).NE.NTMIX(J)) GO TO 20
      IK=MATNO(J)
      GO TO 5
   20 CONTINUE
      IK=6H
    5 PRINT 508,I,MIX(I),CONC(I),IK
      GO TO 7
    6 PRINT 509
    7 DO 8 I=1,NGR
      DO 8 J=1,MAX
      IF(SVM(J,I)) 9,8,9
    8 CONTINUE
      PRINT 510
      GO TO 11
    9 PRINT 511
      M2=MAX+JMAX-1
      DO 10 I=1,NGR
      PRINT 512,I
   10 PRINT 513,(SVM(J,I),J=1,M2)
C     PRINT OUT BOUNDARY CONDITIONS
   11 PRINT 514
      IF(MFR.GT.0)PRINT 525
      DO 25 I=1,NGR
      DO 26 J=1,2
      IF(BETA(J,I))27,29,27
   29 DO 28 K=1,10
      IF(GAMMA(K,J,I))27,28,27
   28 CONTINUE
   26 CONTINUE
      DO 33 K=1,J1
      IF(DELTA(K,I))27,33,27
   33 CONTINUE
   25 CONTINUE
      DO 30 I=2,NGR
      IF(ALPHA(1,I).NE.ALPHA(1,I-1))GO TO 31
      IF(ALPHA(2,I).NE.ALPHA(2,I-1))GO TO 31
   30 CONTINUE
      PRINT 527,ALPHA(1,1),ALPHA(2,1)
      GO TO 32
   31 PRINT 528
      DO 34 I=1,NGR
      PRINT 529,I,(ALPHA(K,I),K=1,2)
   34 CONTINUE
      GO TO 32
   27 DO 12 I=1,NGR
      PRINT 515,I,(ALPHA(K,I),K=1,2),(BETA(K,I),K=1,2),((GAMMA(K,J,I),
     1J=1,2),K=1,10)
      K=1
```

```
      L=N+2
   69 PRINT 524,L,K,DELTA(L,I),DELTA(K,I)
      K=K+1
      L=L-1
      IF(K-J2) 69,69,12
   12 CONTINUE
   32 PRINT 516
      BSQRD=0.0
      IF(IBUK.EQ.1)BSQRD=BUCK(1,1)
      DO 16 M1=1,25
      DO 17 J=1,JMAX
      K=MIR(J)
      IF(K.EQ.M1) GO TO 18
   17 CONTINUE
      GO TO 16
   18 PRINT 517,K
      IF(IBUK.EQ.2)BSQRD=BUCK(K,1)
      DO 13 I=1,NGR
      IF(IBUK.EQ.3.OR.IBUK.EQ.5)BSQRD=BUCK(I,1)
      IF(IBUK.EQ.4)BSQRD=BUCK(I,K)
   13 PRINT 518,I,SIGT(I,K),SIGS(I,K),SIGS1(I,K),VUSIG(I,K),CHI(I,K),
     1BSQRD
      IF(NDS)100,100,14
   14 PRINT 519
      M=1
      DO 15 I=1,NDS
      L=NGR-I-1
      M7=L+M
      PRINT 520,I,(STR(M6,K),M6=M,M7)
   15 M = 26+M-I
C     PRINT UPSCATTER MATRIX IF NECESSARY
  100 IF(NPS)16,16,101
  101 PRINT 604
  604 FORMAT(9X,15HP-1 DOWNSCATTER)
      M=NGR-1
      PRINT 605,(STR1(I,K),I=1,M)
  605 FORMAT(5E18.6)
   16 CONTINUE
      PRINT 521
      AC68=(TIMEBEG-TIMELEFT(AC68))*.001
      PRINT 603,AC68
      RETURN
  603 FORMAT(7X,4HTIMEF9.3)
  500 FORMAT (2X,25HINPUT DATA ***************)
  501 FORMAT (5X,23HGEOMETRY INDICATOR ****I3/5X,23HNUMBER OF POINTS ***
     1***I3/5X,23HNUMBER OF REGIONS *****I3/5X,23HNUMBER OF GROUPS *****
     2*I3/5X,23HDOWNSCATTER GROUPS ****I3/5X,23HANGULAR APPROXIMATION *I
     33/5X,23HANGULAR INTERVALS *****I3/5X,23HPOWER GUESS OPTION ****I3/
     45X,23HCONVERGENCE OPTION ****I3/5X,23HOUTPUT OPTION *********I3/
     5                                5X,23HELEMS. IN MIX. VECT. **I3/5X,2
     63HINPUT PRINT OPTION ****I3/5X,23HITERATION MAXIMUM *****I3)
  502 FORMAT (1H0,4X,23HEPSILON ****************,  E12.5/5X,
     123HINITIAL RADIUS ********,
     2          E12.5/5X,23HEXTRAPOLATION FACTOR **E12.5/5X,23HNORMALIZA
     3TION FACTOR **E12.5)
  503 FORMAT (////1X,11HREGION DATA//7X,6HREGION,7X,6HREGION,8X,7HMAXIMU
     1M,7X,5HDELTA,6X,5HOUTER/8X,3HNO.,8X,8HMATERIAL,5X,11HPOINT INDEX,7
     2X,1HR,8X,6HRADIUS)
  504 FORMAT(8X,I2,12X,I2,12X,I3,5X,  2E12.5)
  505 FORMAT(////2X,12HANGULAR DATA)
  506 FORMAT(6X,2HMUI2,1H*,  E12.5)
  507 FORMAT(1H0,2X,12HMIXTURE DATA)
```

```
508 FORMAT(6X,I2,3X,I2,3X,  E12.5,1H*,3X,A6)
509 FORMAT(1H0,2X,24HNO MIXTURES TO BE FORMED)
510 FORMAT(1H0,2X,21HNO FIXED SOURCE INPUT)
511 FORMAT(1H1,2X,18HFIXED SOURCE INPUT)
512 FORMAT(1H0,3X, 6HGROUP I1,6H *****)
513 FORMAT(6X,  10E11.4)
514 FORMAT(1H1,2X,32HBOUNDARY CONDITION SPECIFICATION)
515 FORMAT(1H0,4X,6HGROUP I2,6H *****,9X,8H**LEFT**,6X,9H**RIGHT**//6X
   1,12HALPHA ******,8X,E12.5,5X,E12.5/6X,12HBETA *******,8X,E12.5,5X,
   2 E12.5/6X,12HGAMMA ******/8X,10HL=0 ******,8X,E12.5,5X,E12.5/8X,10
   3HL=1 ******,8X,E12.5,5X,E12.5/8X,10HL=2 ******,8X,E12.5,5X,E12.5/8
   4X,10HL=3 ******,8X,E12.5,5X,E12.5/8X,10HL=4 ******,8X,E12.5,5X,E12
   5.5/8X,10HL=5 ******,8X,E12.5,5X,E12.5/8X,10HL=6 ******,8X,E12.5,5X
   6,E12.5/8X,10HL=7 ******,8X,E12.5,5X,E12.5/8X,10HL=8 ******,8X,E12.
   75,5X,E12.5/8X,10HL=9 ******,8X,E12.5,5X,E12.5/6X,12HDELTA ******)
516 FORMAT(1H1,2X,18HCROSS SECTION DATA)
517 FORMAT(///4X,11H***********/3X,9H*MATERIAL,I3,1H*,5X,7H*SIGMA*,6X,
   19H*SIGMA S*,4X,9H*SIGMA S*,4X,10H*NU-SIGMA*,3X,9H*FISSION*,4X,10H*
   2BUCKLING*/4X,11H***********,6X,7H*TOTAL*,6X,9H* ZERO  *,4X,9H*  ON
   3E  *,4X,9H*FISSION*,4X,10H*SPECTRUM*)
518 FORMAT(7X,5HGROUPI3,3X,6(1X,  E12.5))
519 FORMAT(1H0,3X,15HTRANSFER MATRIX)
520 FORMAT(9X,9HDOWN**** I2/ (5E18.6))
521 FORMAT(1H0,3X,18HEND OF INPUT PRINT)
522 FORMAT(1H1,5X,16HFISSION SPECTRUM)
523 FORMAT(6X,10HGROUP**** I2,(6E13.5))
524 FORMAT (7X,3HMU(I2,2H)(I2,2H)*,8X,E12.5,5X,E12.5)
525 FORMAT(1H0,10X,27HPERIODIC BOUNDARY CONDITION)
526 FORMAT(5X,23HBUCKLING INPUT OPTION *I3/5X,23HNO. OF REACTION RATES
   1 *I3/5X,23HHOMOGENIZED GROUPS ****I3/5X,23HHOMOGENIZED REGIONS ***
   2I3)
602 FORMAT(9X,8H  UP*** I1,5(2X,  E12.5))
600 FORMAT(5X,23HP1-DOWNSCATTER ********I3/5X,23HTAPE ELEMENTS *******
   1**I3/5X,23HSEARCH OPTION *********I3/5X,23HSEARCH ZONE ************
   2I3/5X,23HSEARCH POS. IN MIX*****I3/5X,23H FILL POS. IN MIX***** I3
   3)
601 FORMAT(5X,23HSECOND GUESS **********, E12.5/5X,23HEIGENVALUE DESIR
   1ED ****, E12.5/5X,23HSEARCH RATIO **********, E12.5)
527 FORMAT(1H0,4X,22HALPHA FOR ALL GROUPS = E12.5,11H LEFT, AND  E12.5
   1,6H RIGHT)
528 FORMAT(1H0,4X,11HALPHA *****,12X,8H**LEFT**,6X,9H**RIGHT**/)
529 FORMAT(5X,6HGROUP I2,14X,E12.5,5X,E12.5)
    END
    SUBROUTINE MIXX
    COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
   1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
   2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
   3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
   4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
   5EM1,MATNO(20),ISET,PRORT(12),ISAVE,ELOWER(27)
    DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
   1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
   2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
   3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BEUCK(
   426,25),BETA(2,26),LRATE(25)
    EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
   1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
   2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
   3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
   420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IC),(LL
   5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),TIX),
   6(LL(166),NTMIX),(LL(186),NWGP),(LL(187),NWREG),(LL(188),LHREG),
```

```
     7(LL(198),LHGP),(LL(224),LFREQ),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREQ),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),IN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      DIMENSION ALL(700)
      COMMON/2/ILL(700),IRECNO(20),NIG(20),NEG(20)
      BANK(1),LINK1,/A1/,SEARCH,SCHECK
      COMMON/A2/EE(1300)
      DIMENSION SIGFIS(26,25),SIGFIV(650),SIGCAP(26,25),SIGCAV(650)
      EQUIVALENCE (EE(1),SIGFIV,SIGFIS),(EE(651),SIGCAV,SIGCAP)
      BANK(0),/2/,/A2/,INPR,MIXX
      EQUIVALENCE (ILL(1),ALL(1))
      NMIX=NMIX
      REWIND 11
      IF(LGO)800,5000,5000
 5000 IF(MMIX)800,800,700
  700 PRINT 506
      LIB=11
      REWIND LIB
      MAXIG=MAXEG=0
    1 IF(UNIT,LIB)1,2,3
    3 PRINT 4
    4 FORMAT(*0EOF OR PARITY ERROR ON LIB TAPE*)
      CALL Q8QERROR(0,4HBUG.)
    2 BUFFER IN (LIB,1)(ILL(1),ILL(8))
    5 IF(UNIT,LIB)5,6,3
    6 K1=ILL(1)+1
      BUFFER IN(LIB,1)(ILL(1),ILL(K1))
    7 IF(UNIT,LIB)7,8,3
    8 DO 10 I1=2,K1
      IF(ISET.EQ.ILL(I1))9,10
    9 K2=I1-1
      BUFFER IN(LIB,1)(K3,K3)
  799 IF(UNIT,LIB)799,11
   10 CONTINUE
      PRINT 12
   12 FORMAT(*0CANT FIND SET ID*)
      CALL Q8QERROR(0,4HBUG.)
   11 K3=K2-1
      DO 13 I1=1,K3
   13 CALL SKIPFILE(LIB)
      BUFFER IN(LIB,1)(ILL(1),ILL(8))
   16 IF(UNIT,LIB)16,17,3
   17 IF(ISET.EQ.ILL(1))20,18
   18 PRINT 19
   19 FORMAT(*0FOUND WRONG SET*)
      CALL Q8QERROR(0,4HBUG.)
   20 NGR=ILL(4)
      M=ILL(5)
      K3=3*NGR+2
      BUFFER IN(LIB,1)(ALL(1),ALL(K3))
      K1=2+M
      K3=2*NGR+2
   21 IF(UNIT,LIB)21,22,3
   22 DO 201 I1=1,NGR
      ELOWER(I1+1)=ALL(I1+1)
      DO 201 I2=1,MMIX
  201 VINV(I1,NTMIX(I2))=1./ALL(I1+K3)
```

```
      ELOWER(1)=ALL(1)
      BUFFER IN (LIB,1)(ILL(1),ILL(K1))
   23 IF(UNIT,LIB)23,24,3
   24 DO 28 I1=1,MMIX
      DO 26 I2=1,M
      IF(MATNO(I1).EQ.ILL(I2))GO TO 280
   26 CONTINUE
      PRINT 27,MATNO(I1),I1,I2
   27 FORMAT(37HOTHIS MAT NOT IN CROSS SECTION SET - ,A6,2I6)
      CALL Q8QERROR(0,4HBUG.)
  280 NEG(I1)=ILL(I2+M)
   28 CONTINUE
      ICON1=0
      DO 282 I1=1,MMIX
      ICON=100000
      DO 283 I2=1,MMIX
      IF(NEG(I2).LE.ICON1) GO TO 283
      IF(ICON.LT.NEG(I2)) GO TO 283
      ICON=NEG(I2)
      NIG(I1)=I2
  283 CONTINUE
      ICON1=ICON
  282 IRECNO(I1)=ICON
      K4=4
      DO 52 K5=1,MMIX
      I=NIG(K5)
      IFLAG=0
      DO 284 IJK1=1,NRATE
      IF(MATNO(I).EQ.LRATE(IJK1))IFLAG=IJK1
  284 CONTINUE
  286 DO 121 J=1,NGR
      SIGT(J,NTMIX(I))=0.
      SIGS(J,NTMIX(I))=0.
      SIGS1(J,NTMIX(I))=0.
      CHI(J,NTMIX(I))=0.0
  121 VUSIG(J,NTMIX(I))=0.
      DO 122 J=1,234
  122 STR(J,NTMIX(I))=0.
      DO 123 J=1,25
  123 STR1(J,NTMIX(I))=0.0
      DO 101 I2=1,20
  101 NEG(I2)=0
      K8=IRECNO(K5)+K4
      DO 281 IJ=1,K8
      BUFFER IN (LIB,1)(ILL(1),ILL(1))
      K4=K4+1
   31 IF(UNIT,LIB)31,281,3
  281 CONTINUE
      BUFFER IN(LIB,1)(ALL(1),ALL(17))
      K4=K4+1
   32 IF(UNIT,LIB)32,331,3
  331 PRINT 15,(ILL(J),J=1,10),NTMIX(I)
   15 FORMAT(5X,A6,9A8,5X,2H** I3,2H**)
      DO 332 I2=2,8
  332 ILL(I2)=ILL(I2+9)
   34 IF(ILL(4))37,391,35
   35 PRINT 36
   36 FORMAT(*0GROUP DEPENDENT FISSION SPECTRUM*)
      CALL Q8QERROR(0,4HBUG.)
   37 K3=NGR+8
      BUFFER IN(LIB,1)(ILL(9),ILL(K3))
      K4=K4+1
```

```
 38 IF(UNIT,LIB)38,381,3
381 DO 382 I2=1,NGR
382 CHI(I2,NTMIX(I))=ALL(I2+8)
 39 KK=14
    GO TO 40
391 KK=12
 40 ILEL=ILL(5)+1
    ILIN=ILL(6)+1
    ILN2N=ILL(7)+1
 49 K3=KK+3*(ILEL+ILIN+ILN2N)
    DO 496 J=1,NGR
    BUFFER IN(LIB,1)(ILL(9),ILL(K3))
    K4=K4+1
420 IF(UNIT,LIB)420,430,3
430 IF(ILEL.EQ.0) GO TO 4300
    IK=KK+ILEL+1
    ILSIGEL=IK+ILEL
    IGEL=ILL(IK)
    IFEL=ILL(KK+1)
4300 IF(ILIN.EQ.0) GO TO 4301
    IK=KK+3*ILEL+ILIN+1
    ILSIGIN=IK+ILIN
    IGIN=ILL(IK)
    IFIN=ILL(IK-ILIN)
4301 IF(ILN2N.EQ.0) GO TO 4302
    IK=KK+3*(ILEL+ILIN)+ILN2N+1
    ILSIGN2N=IK+ILN2N
    IGN2N=ILL(IK)
    IFN2N=ILL(IK-ILN2N)
4302 IF(NDS.LT.IGEL) NDS=IGEL
    IF(NDS.LT.IGIN) NDS=IGIN
    IF(NDS.LT.IGN2N) NDS=IGN2N
    IF(NDS.GT.12) PRINT 480
    IF(NDS.GT.12) NDS=12
    IF(IFEL.GT.0) PRINT 481
480 FORMAT(*0 THIS SET HAS MORE THAN 12 DOWNSCATTERS, EXCESS ARE IGNOR
   1ED*)
481 FORMAT(*0 THIS SET HAS UPSCATTER, WHICH IS BEING IGNORED*)
    IF(IFIN.GT.0) PRINT 481
    IF(IFN2N.GT.0) PRINT 481
488 SIGT(J,NTMIX(I))=ALL(10)
    IF(IFLAG.GT.0)SIGCAP(J,IFLAG)=ALL(11)
    IF(ILL(4).LT.0)472,473
472 VUSIG(J,NTMIX(I))=ALL(13)*ALL(14)
    IF(IFLAG.GT.0)SIGFIS(J,IFLAG)=ALL(13)
473 M2=IFEL+IGEL
    IF(M2.GT.0)M2=M2+1
    M3=IFIN+IGIN
    IF(M3.GT.0) M3=M3+1
    M4=IFN2N+IGN2N
    IF(M4.GT.0)M4=M4+1
    K8=M2+M3+M4
    IF(K8.GT.0)112,111
111 K8=7
112 M5=K3+1
    K8=M5+1+K8
    BUFFERIN(LIB,1)(ILL(M5),ILL(K8))
    K4=K4+1
113 IF(UNIT,LIB)113,114,3
114 IF(M2.EQ.0) GO TO 119
    SIGS(J,NTMIX(I))=ALL(M5+IFEL+1)+ALL(ILSIGEL)
    DO 118 M7=1,IGEL
```

```
      IF (M7.GT.12) GO TO 119
      K1=25
      K2=0
      DO 117 M9=2,M7
      K2=K2+K1
  117 K1=K1-1
      K1=K2+J
  118 STR(K1,NTMIX(I))= ALL (M5+M7+IFEL+1)*ALL(ILSIGEL)
  119 IF(M3.EQ.0) GO TO 126
      SIGS(J,NTMIX(I))=SIGS(J,NTMIX(I))+ALL(M5+M2+IFIN+1)*ALL(ILSIGIN)
      DO 125 M7=1,IGIN
      IF (M7.GT.12) GO TO 126
      K1=25
      K2=0
      DO 124 M9=2,M7
      K2=K2+K1
  124 K1=K1-1
      K1=K2+J
  125 STR(K1,NTMIX(I))=STR(K1,NTMIX(I))+ALL(M5+M7+M2+IFIN+1)*
     1ALL(ILSIGIN)
  126 IF(M4.EQ.0) GO TO 127
      SIGS(J,NTMIX(I))=SIGS(J,NTMIX(I))+ALL(M5+M2+M3+IFN2N+1)*
     1ALL(ILSIGN2N)
      DO 129 M7=1,IGN2N
      IF (M7.GT.12) GO TO 127
      K1=25
      K2=0
      DO 128 M9=2,M7
      K2=K2+K1
  128 K1=K1-1
      K1=K2+J
  129 STR(K1,NTMIX(I))=STR(K1,NTMIX(I))+ALL(M7+M5+M2+M3+IFN2N+1)*
     1ALL(ILSIGN2N)
  127 IF(M2+M3+M4.GT.0) GO TO 1143
      IF(ILEL.GT.0)SIGS(J,NTMIX(I))=ALL(ILSIGEL)
      IF(ILIN.GT.0)SIGS(J,NTMIX(I))=SIGS(J,NTMIX(I))+ALL(ILSIGIN)
      IF(ILN2N.GT.0)SIGS(J,NTMIX(I))=SIGS(J,NTMIX(I))+ALL(ILSIGN2N)
 1143 IF(ILEL.GT.1)130,131
  130 M2=ILL(KK+2)+ILL(KK+ILEL+2)+1
      GO TO 132
  131 M2=0
  132 IF(ILIN.GT.1)133,134
  133 M3=ILL(ILSIGEL+ILEL+1)+ILL(ILSIGEL+ILEL+ILIN+1)+1
      GO TO 135
  134 M3=0
  135 IF(ILN2N.GT.1)136,137
  136 M4=ILL(ILSIGIN+ILIN+1)+ILL(ILSIGIN+ILIN+ILN2N+1)+1
      GO TO 138
  137 M4=0
  138 K8=M2+M3+M4
      IF(K8.EQ.0)496,139
  139 K8=M5+K8+1
      BUFFER IN(LIB,1)(ILL(M5),ILL(K8))
      K4=K4+1
  140 IF(UNIT.LIB)140,141,3
  141 EM=0.
      IF(ILEL.GT.1)142,143
  142 IF(M2.GT.1) GO TO 1420
      SIGS1(J,NTMIX(I))=ALL(ILSIGEL+1)
      EN1=0.0
      GO TO 1421
 1420 M6=ILL(KK+2)
```

```
      SIGS1(J,NTMIX(I))=ALL(M5+M6+1)+ALL(ILSIGEL+1)
      NPS=1
      STR1(J,NTMIX(I))=ALL(M5+M6+2)+ALL(ILSIGEL+1)
      EN1=STR1(J,NTMIX(I))
 1421 EM=ALL(ILSIGEL+1)-SIGS1(J,NTMIX(I))-EN1
      IF(M2.LE.2)EM=0.
  143 IF(ILIN.GT.1)144,145
  144 IF(M3.GT.1) GO TO 1440
      EN=ALL(ILSIGIN+1)
      SIGS1(J,NTMIX(I))=SIGS1(J,NTMIX(I))
      EN1=0.0
      GO TO 1441
 1440 M6=ILL(ILSIGEL+ILEL+1)+ILL(KK+ILEL+2)+M6+1
      EN=ALL(M5+M6+1)+ALL(ILSIGIN+1)
      SIGS1(J,NTMIX(I))=SIGS1(J,NTMIX(I))+EN
      EN1=ALL(M5+M6+2)+ALL(ILSIGIN+1)
      STR1(J,NTMIX(I))=STR1(J,NTMIX(I))+EN1
 1441 EM=EM+ALL(ILSIGIN+1)-EN-EN1
  145 IF(ILN2N.GT.1)146,147
  146 IF(M4.GT.1) GO TO 1460
      EN=ALL(ILSIGN2N+1)
      SIGS1(J,NTMIX(I))=SIGS1(J,NTMIX(I))+EN
      EN1=0.0
      GO TO 1461
 1460 M6=ILL(ILSIGIN+ILIN+1)+ILL(ILSIGIN-ILIN+1)+M6+1
      EN=ALL(M5+M6+1)+ALL(ILSIGN2N+1)
      SIGS1(J,NTMIX(I))=SIGS1(J,NTMIX(I))+EN
      EN1=ALL(M5+M6+2)+ALL(ILSIGN2N+1)
      STR1(J,NTMIX(I))=STR1(J,NTMIX(I))+EN1
 1461 EM=EM+ALL(ILSIGN2N+1)-.5*(EN+EN1)
  147 SIGT(J,NTMIX(I))=SIGT(J,NTMIX(I))-EM
      SIGS(J,NTMIX(I))=SIGS(J,NTMIX(I))-EM
      EM=0.
      IF((ILEL.LT.3).AND.(ILIN.LT.3).AND.(ILN2N.LT.3))GO TO 496
      DO 148 M8=3,ILEL
  148 EM=EM+ALL(ILSIGEL+M8-1)
      DO 149 M8=3,ILIN
  149 EM=EM+ALL(ILSIGIN+M8-1)
      DO 150 M8=3,ILN2N
  150 EM=EM+ALL(ILSIGIN+M8-1)
      SIGT(J,NTMIX(I))=SIGT(J,NTMIX(I))-EM
      SIGS(J,NTMIX(I))=SIGS(J,NTMIX(I))-EM
  496 CONTINUE
   52 CONTINUE
      REWIND LIB
  800 IF(NMIX)50,50,1000
C     IS THERE ANY MIXING REQUIRED, IF NOT, RETURN
C     IF THERE IS MIXING, MIXTURE VECTOR MUST HAVE AT LEAST TWO ELEMENTS
 1000 IF(NMIX-2)1031,1002,1002
 1031 PRINT 501
      NCE=1
      GO TO 50
C     FIRST VECTOR IN CONCENTRATIONS MUST BE ZERO, LAST MUST NOT
 1002 IF(CONC(1))1003,1021,1003
 1021 IF(CONC(NMIX))4,1003,4
 1003 PRINT  502
      NCE=1
      GO TO 50
    4 DO 300 I=1,NMIX
      IF(MIX(I)-25) 300,300,301
  301 PRINT 503
      NCE=1
```

```
      GO TO 50
  300 CONTINUE
      M3=1
      DO 1008 I=2,NMIX
C     NO TWO NEIGHBORING ELEMENTS BE ZERO
      IF(CONC(I))1006,1005,1006
 1005 IF(M3)1009,1007,1009
 1006 M3=0
      GO TO 1008
 1007 M3=1
 1008 CONTINUE
      GO TO 1010
 1009 PRINT  502
      NCE=1
      M3=0
      GO TO 50
C     BEGIN MIXING ELEMENTS
 1010 DO 1011 I=1,6
 1011 CC(I)=0.0
      ZNORM6=0.0
      M5=1
      M4=2
      DO 15 I=2,NMIX
      IF (I-NMIX) 100,1101,100
 1101 M6 = NMIX
      GO TO 102
  100 IF(CONC(I)) 15,12,15
   12 M6=I-1
  102 DO 1014 L=1,NGR
      DO 1013 J=M4,M6
      K=MIX(J)
      CC(1)=CC(1)+SIGT(L,K)*CONC(J)
      CC(2)=CC(2)+SIGS(L,K)*CONC(J)
      CC(3)=CC(3)+SIGS1(L,K)*CONC(J)
      IF(L.LT.NGR)CC(4)=CC(4)+STR1(L,K)*CONC(J)
      IF(CONC(J).GT.ZNORM6)1027,1028
 1027 ZNORM6=CONC(J)
      M10=K
 1028 IF(VUSIG(L,K))1013,1013,1022
 1022 IF(CONC(J)-CC(6))1013,1013,1023
 1023 CC(6)=CONC(J)
      M9=K
 1013 CC(5)=CC(5)+VUSIG(L,K)*CONC(J)
      M7=MIX(M5)
      SIGT(L,M7)=CC(1)
      SIGS(L,M7)=CC(2)
      SIGS1(L,M7)=CC(3)
      IF(L.LT.NGR) STR1(L,M7)=CC(4)
      VUSIG(L,M7)=CC(5)
      CC(1)=0.0
      CC(2)=0.0
      CC(3)=0.0
      CC(4)=0.0
 1014 CC(5)=0.0
      DO 1029 M8=1,NGR
 1029 VINV(M8,M7)=VINV(M8,M10)
      IF(CC(6))1026,1025,1026
 1026 CC(6)=0.0
      DO 1024 M8=1,NGR
 1024 CHI(M8,M7)=CHI(M8,M9)
 1025 M5=I
      M4=I+1
```

```
   15 CONTINUE
      IF(NDS) 50,50,1016
 1016 M5=1
      M4=2
      DO 1020 I=2,NMIX
      IF (I-NMIX) 200,1201,200
 1201 M6 = NMIX
      GO TO 202
  200 IF(CONC(I))1020,1017,1020
 1017 M6=I-1
  202 DO 19 L=1,234
      DO 1018 J=M4,M6
      K=MIX(J)
 1018 CC(6)=CC(6)+STR(L,K)*CONC(J)
      M7=MIX(M5)
      STR(L,M7)=CC(6)
   19 CC(6)=0.0
      M5=I
      M4=I+1
 1020 CONTINUE
   50 RETURN
  501 FORMAT(47H MIXING VECTOR MUST HAVE AT LEAST TWO ELEMENTS )
  502 FORMAT(35H INCONSISTENT CONCENTRATION VECTOR )
  503 FORMAT(51H MATERIAL NUMBER GREATER THAN 25 IN MIXTURE VECTOR )
  506 FORMAT(29H ELEMENTS REQUESTED FROM TAPE,53X,15HMATERIAL NUMBER)
      END
      SUBROUTINE  SCHECK
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),DC),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NWGP),(LL(187),NWREG),(LL(188),LWREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      DIMENSION ALL(700)
      COMMON/2/ILL(700),IRECNO(20),NIG(20),NEG(20)
      BANK(1),LINK1,/A1/,SEARCH,SCHECK
      COMMON/A2/EE(1300)
      DIMENSION SIGFIS(26,25),SIGFIV(650),SIGCAP(26,25),SIGCAV(650)
      EQUIVALENCE (EE(1),SIGFIV,SIGFIS),(EE(651),SIGCAV,SIGCAP)
      BANK(0),/2/,/A2/,INPR,MIXX
      EQUIVALENCE (ILL(1),ALL(1))
   39 S2=0.0
```

```
      M2=MAX+JMAX-1
C     IS THERE FIXED SOURCE INPUT
      DO 1 I=1,NGR
      DO 1 J=1,M2
      IF(SVM(J,I)) 6,1,6
    1 CONTINUE
C     IF NOT, IS THERE BOUNDARY SOURCE INPUT
      DO 2 I=1,NGR
      DO 2 J=1,2
      DO 2 K=1,10
      IF(GAMMA(K,J,I)) 6,2,6
    2 CONTINUE
C     IF,NEITHER, IS THERE FIXED BOUNDARY FLUX
      DO 3 I=1,NGR
      DO 3 K=1,J1
      IF(DELTA(K,I)) 6,3,6
    3 CONTINUE
C     NO SOURCES PRESENT, CHECK IF THEY ARE NECESSARY
C     ITOUT IS SET TO 50 UNLESS READ OVER
      IF(ITOUT) 4,4,5
C     THEY ARE, ERROR INDICATION
    4 PRINT 500
      NCE=1
      RETURN
C     THEY ARE NOT. CHECK FOR FISSION CROSS SECTION
C     S2=1 IMPLIES NO SOURCES,S2=0 IMPLIES THERE ARE SOURCES
    5 S2=1.
    6 DO 7 I=1,JMAX
      K=MIR(I)
      K1=K
      DO 7 J=1,NGR
      IF(VUSIG(J,K)) 7,7,10
    7 CONTINUE
C     NO FISSION CROSS SECTIONS, CHECK IF NEEDED
      IF(S2) 9,9,8
C     THEY ARE, ERROR INDICATION
    8 PRINT 501
      NCE=1
C     THEY ARE NOT, RETURN
    9 ITOUT=0
      RETURN
C     THERE ARE FISSION X-SECTS AND OUTER ITERATIONS ARE NEEDED
C     CHECK FOR NON-ZERO CHI,S
   10 DO 50 I=1,JMAX
      K=MIR(I)
      DO 41 J=1,NGR
      IF(VUSIG(J,K))41,41,42
   41 CONTINUE
      DO 43 J=1,NGR
   43 CHI(J,K)=0.0
      GO TO 50
   42 DO 40 J=1,NGR
      IF(CHI(J,K))40,40,50
   40 CONTINUE
   47 DO 48 J=1,NGR
   48 CHI(J,K)=CHI(J,K1)
      DO 49 J=1,NGR
      IF(CHI(J,K1))49,49,50
   49 CONTINUE
      PRINT 502
      NCE=1
      RETURN
```

```
   50 CONTINUE
C     IS NORMALIZATION REQUIRED
   51 IF(S2) 11,11,15
C     S2=0, THERE ARE SOURCES AND NO NORMALIZATION IS REQUIRED
C     CHECK TO DETERMINE IF PREVIOUS SOURCE IS AVAILABLE AND DESIRED
C     PREVIOUS INCLUDES INPUT GUESS
   11 IF(LPG) 14,12,14
C     NO GUESS AVAILABLE, START WITH ZERO GUESS
   12 M2=MAX*JMAX-1
      DO 600 I=1,M2
  600 POWR1(I)=0.0
C     GUESS AVAILABLE, USE IT
   14 RETURN
C     NORMALIZATION REQUIRED.HOMOGENEOUS CASE
C     CALCULATE FISSIONABLE VOLUME
   15 VOL=0.0
      DO 21 I=1,JMAX
      K=MIR(I)
  101 DO 16 J=1,NGR
      IF(VUSIG(J,K)) 16,16,17
   16 CONTINUE
      GO TO 21
   17 M7=II(I)
      IF(I-1) 18,18,19
   18 M6=1
      GO TO 20
   19 M6=II(I-1)
   20 IF(IG.EQ.1) VOL=VOL+XPT(M7)-XPT(M6)
      IF(IG.EQ.-1)VOL=VOL+XPT(M7)-XPT(M6)
      IF(IG.EQ.3) VOL=VOL+(XPT(M7)**3-XPT(M6)**3)*4.1887902
      IF(IG.EQ.-3) VOL=VOL+(XPT(M7)**3-XPT(M6)**3)*4.1887902
   21 CONTINUE
C     NOW NORMALIZE THE POWER TO TOTAL INTEGRAL OF FAC
C     IS GUESS FROM PREVIOUS CASE AVAILABLE (PREVIOUS INCLUDES INPUT)
   23 IF(LPG) 29,29,24
C     YES,INTEGRATE OVER THIS REACTOR
   24 SUM=0.0
      M7=1
C     USE OLD SHAPE OVER NEW FISSIONABLE VOLUME ONLY
      DO 25 I=1,JMAX
      M6=MIR(I)
  102 DO 100 M=1,NGR
      IF(VUSIG(M,M6)) 100,100,99
  100 CONTINUE
      GO TO 25
   99 L=II(I)-1
      DO 37 J=M7,L
      K=I+J-1
      IF((IG.EQ.3).OR.(IG.EQ.-3))GO TO 301
      EM1=.5*DELR(I)
      EM2=EM1
      GO TO 37
  301 EM1=DELR(I)*(XPT(J+1)*XPT(J+1)+2.*XPT(J+1)*XPT(J)+3.*XPT(J)*XPT(J)
     1)*1.0471976
      EM2=DELR(I)*(3.*XPT(J+1)*XPT(J+1)+2.*XPT(J+1)*XPT(J)+XPT(J)*XPT(J)
     1)*1.0471976
   37 SUM=SUM+EM1*POWR1(K)+EM2*POWR1(K+1)
   25 M7=II(I)
   27 M2=MAX+JMAX-1
      DO 28 I=1,M2
   28 POWR1(I)=(POWR1(I)/SUM)*FAC
C     NO PREVIOUS GUESS, ESTABLISH A FLAT GUESS
```

```
      RETURN
   29 DO 36 I=1,JMAX
C     START WITH FLAT GUESS OVER ALL FISSIONABLE REGIONS
  103 DO 30 J=1,NGR
      K=MIR(I)
      IF (VUSIG(J,K)) 30,30,31
   30 CONTINUE
      GO TO 36
   31 IF(I-1) 32,32,33
   32 M6=1
      GO TO 34
   33 M6=II(I-1)+I-1
   34 M7=II(I)+I-1
      DO 35 L=M6,M7
   35 POWR1(L)=FAC/VOL
   36 CONTINUE
      RETURN
  500 FORMAT(39H ERROR,NO SOURCE IN INHOMOGENEOUS CASE )
  501 FORMAT(34H ERROR, ALL FISSION X-SECTS  ZERO )
  502 FORMAT(23H ERROR,NO NON-ZERO CHI )
       END
C*********************OVERLAY 2*********************************************
      PROGRAM LINK 2
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGE 1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IG),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      COMMON/2/XL(6000)
      COMMON/A2/SB(22,26),AL7(22),U(22),V(22),WU(22),AQZ(22),AQ(22,3,22)
     1,Q1Z(22),Q1(22,9),W(11,12),P1(11,9),
     IP2(11,9),AL1(22,9),AL2(22,9),Z(22),SMSC(22,22),
     3LMAX,A(11, 22),B(2,11,22),C(11,22),D(11,22),UA(6000)
      COMMON/A3/D1(242),DM(484)
      BANK(0),/2/,/A2/,OPINT,LINK2,ALQS,/A3/,SETUP,AQSJL
      BANK(1),/A1/
      REWIND 2
      REWIND 8
      PI=3.14159265
      LMAX=9
      IJM=JMAX
```

```
      IJ1=J1
      J1=IJM
      JMAX=IJ1
      IF ((IG.EQ.3).OR.(IG.EQ.-3)) JMAX=JMAX+1
      IF(NOT.LT.5)GO TO 3
      IF(MADJ.LE.0)GO TO 3
      DO 16 I=1,IJM
      DO 10 J=1,NGR
      K=MIR(I)
      IF(VUSIG(J,K))10,10,11
   10 CONTINUE
      GO TO 16
   11 IF(I-1)12,12,13
   12 M6=1
      GO TO 14
   13 M6=II(I-1)+I-1
   14 M7=II(I)+I-1
      DO 15 L=M6,M7
   15 POWR1(L)=FAC/VOL
   16 CONTINUE
    3 IF(IG)115,116,116
  115 PRINT 506
  506 FORMAT(/14H POWR1, LINK 2)
      J=MAX+IJM-1
      PRINT 502,(POWR1(I),I=1,J)
  116 IF((IG.EQ.3).OR.(IG.EQ.-3))GO TO 103
      DO 100 J=1,JMAX
  100 U(J)=EMU(J)
      GO TO 105
  103 DO 104 J=1,J2
      M6=JMAX-J+1
      U(J)=EMU(J)
  104 U(M6)=EMU(M6-1)
  105 CALL OPINT
      CALL ALQS
      DO 101 J=1,JMAX
      DO 101 L=1,LMAX
  101 Q1(J,L)=Q1(J,L)/Q1Z(J)
      LMAX=2
      CALL AQSJL
      PIF=4.*PI
      PIFI=1./PIF
      DO 102 J=1,JMAX
      AQZ(J)=ALZ(J)*.5
      DO 102 JA=1,JMAX
      DO 102 L=1,LMAX
      EL=L
  102 AQ(JA,L,J)=AQ(JA,L,J)*(2.*EL+1.)*.5
      JMAX=IJM
      J1=IJ1
      J3=J2-1
      CALL SETUP
      IDIM=0
      DO 800 J=1,J1
  800 IDIM=IDIM+J
      ICR=1
      IF(MFR.GT.0)ICR=2
      IUDIM=IDIM+ICR*J1*J2
      IK=J2
      IF(XABSF(IG).EQ.3)IK=J3
      ILDIM=J1*J1-IDIM+ICR*J1*IK
      M9=IUDIM*(MAX-1)+IDIM
```

```
      IF(MFR.GT.0)M9=M9-J2*J2
      M10=IUDIM*MAX
      IF(M9.LE.3000)K2=1
      IF(M9.GT.3000)K2=(M10/3000)+1
  106 K1=MAX/K2
      IF(MAX-K1*K2.GT.1)K1=K1+1
      IF(K1*IUDIM.LE.3000) GO TO 107
      K2=K2+1
      GO TO 106
  107 NN(1)=K1
      NN(2)=K2
      NN(3)=K1*ILDIM
      NN(5)=(MAX-1)*IUDIM-K1*IUDIM*(K2-1)*IDIM-(ICR-1)*J2*J2
      IF(K2.EQ.1)NN(3)=NN(3)-ICR*J1*IK
 1006 NN(4)=(MAX-K1*(K2-1))*ILDIM-J1*IK*ICR
      NN(6)=K1*IUDIM
 1003 PRINT 509,K1,K2,NN(3),NN(5),NN(4),NN(6)
  509 FORMAT(26H NO. OF POINTS PER READ = I6,18H   NO. OF READS = I4/43H
     1  NO. OF LOWER AND UPPER WORDS FIRST READ =I6,4H ANDI6/42H  NO. OF
     2 LOWER AND UPPER WORDS LAST READ =I6,4H AND I6)
      IF((NOT.GE.5).AND.(MADJ.GT.0))31,2
   31 CALL SEGMENT(2,3,29)
      GO TO 112
    2 IF((IG.EQ.3).OR.(IG.EQ.-3)) GO TO 20
      CALL SEGMENT(2,1,29)
      GO TO 21
   20 CALL SEGMENT(2,2,29)
   21 CONTINUE
      IF(IG.GT.0) GO TO 112
  117 PRINT 507
  507 FORMAT(/14H SB, ALZ, U, V)
      PRINT 502,((SB(I,J),I=1,J1),J=1,NGR)
      PRINT 502,(ALZ(I),I=1,J1)
      PRINT 502,(U(I),I=1,J1)
      PRINT 502,(V(I),I=1,J1)
  112 IF (UNIT,2) 112,113
  113 IF (UNIT,8) 113,114
  114 REWIND 2
      REWIND 8
      RETURN
  500 FORMAT (4(5X,E13.5))
  501 FORMAT(//33H CHECKOUT-ANGULAR DATA-U,V,Z,WU  )
  502 FORMAT (6(5X,E13.5))
  503 FORMAT(///35H MATRIX ELEMENTS -A,B,C,D  I=1,NGR )
  504 FORMAT(///37H UPPER DIAGONAL PLUS FACTORED MATRIX )
  505 FORMAT(5X,100(1H*))
      END LINK 2
      SUBROUTINE OPINT
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),EUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
```

```
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),BG),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),TIX),
     6(LL(166),NTMIX),(LL(186),NWGP),(LL(187),NWREG),(LL(188),LWREG),
     7(LL(198),LWGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),IN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      COMMON/2/XL(6000)
      COMMON/A2/SB(22,26),AL7(22),U(22),V(22),WU(22),AQZ(22),AQ(22,3,22)
     1,Q1Z(22),Q1(22,9),W(11,12),P1(11,9),
     IP2(11,9),AL1(22,9),AL2(22,9),7(22),SMSC(22,22),
     3LMAX,A(11, 22),B(2,11,22),C(11,22),D(11,22),UA(6000)
      BANK(0),/2/,/A2/,OPINT,LINK2,ALOS,/A3/,SETUP,AQSJL
      BANK(1),/A1/
      J=1
618   IF(J-J2) 619,619,627
619   MM=2
620   W (J,1)=U(J)
621   W(J,MM)=W(J,MM-1)*U(J)
622   IF(MM-12) 623,625,625
623   MM=MM+1
624   GO TO 621
625   J=J+1
626   GO TO 618
627   J=1
628   IF(J-J2) 629,633,633
629   DO 630 MM=1,12
630   W(J,MM)=W(J+1,MM)-W(J,MM)
631   J=J+1
632   GO TO 628
633   IF(LMAX)42,42,30
 30   J=2
 31   P1(J,1)=0.5*W(J,2)
      P2(J,1)=W(J,3)/3.0
      IF(LMAX-1)41,41,32
 32   P1(J,2)=(W(J,3)+W(J,1))/2.0
      P2(J,2)=(3.0*W(J,4)-2.0*W(J,2))/8.0
      IF(LMAX-2)41,41,33
 33   P1(J,3)=(5.0*W(J,4)-6.0*W(J,2))/8.0
      P2(J,3)=(W(J,5)-W(J,3))/2.0
      IF(LMAX-3)41,41,34
 34   P1(J,4)=(7.0*W(J,5)-10.0*W(J,3)+3.0*W(J,1))/8.0
      P2(J,4)=(35.0*W(J,6)-45.0*W(J,4)+9.0*W(J,2))/48.0
      IF(LMAX-4)41,41,35
 35   P1(J,5)=(21.0*W(J,6)-35.0*W(J,4)+15.0*W(J,2))/16.0
      P2(J,5)=(9.0*W(J,7)-14.0*W(J,5)+5.0*W(J,3))/8.0
      IF(LMAX-5)41,41,36
 36   P1(J,6)=(33.0*W(J,7)-63.0*W(J,5)+35.0*W(J,3)-5.0*W(J,1))/16.0
      P2(J,6)=(231.0*W(J,8)-420.0*W(J,6)+210.0*W(J,4)-20.0*W(J,2))/128.0
      IF(LMAX-6)41,41,37
 37   P1(J,7)=(429.*W(J,8)-924.*W(J,6)+630.*W(J,4)-140.*W(J,2))/128.
      P2(J,7)=(143.*W(J,9)-297.*W(J,7)+189.*W(J,5)-35.*W(J,3))/48.
      IF(LMAX-7)41,41,38
 38   P1(J,8)=(715.*W(J,9)-1716.*W(J,7)+1386.*W(J,5)-420.*W(J,3)+35.*W(J
     1,1))/128.
      P2(J,8)=(1287.*W(J,10)-3003.*W(J,8)+2310.*W(J,6)-630.*W(J,4)+35.*
     1W(J,2))/256.
```

```
      IF(LMAX-8)41,41,39
   39 P1(J,9)=(2431.*W(J,10)-6435.*W(J,8)+6006.*W(J,6)-2310.*W(J,4)+315.
     1*W(J,2))/256.
      P2(J,9)=(1105.*W(J,11)-2860.*W(J,9)+2574.*W(J,7)-924.*W(J,5)+105.*
     1W(J,3))/128.
   41 J=J+1
      IF(J-J2)31,42,42
   42 P1(1,1)=-.5
      P1(1,2)=0.0
      P1(1,3)=.125
      P1(1,4)=0.0
      P1(1,5)=-.0625
      P1(1,6)=0.0
      P1(1,7)=5./128.
      P1(1,8)=0.0
      P1(1,9)=-7./256.
      P1(1,10)=0.0
      P2(1,1)=1./3.
      P2(1,2)=-.125
      P2(1,3)=0.0
      P2(1,4)=1./48.
      P2(1,5)=0.0
      P2(1,6)=-1./128.
      P2(1,7)=0.0
      P2(1,8)=1./256.
      P2(1,9)=0.0
      DO 44 L=1,LMAX
      J=2
   43 P1(1,L)=P1(1,L)-P1(J,L)
      P2(1,L)=P2(1,L)-P2(J,L)
      J=J+1
      IF(J-J2) 43,44,44
   44 CONTINUE
      RETURN
      END
      SUBROUTINE SETUP
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),DG),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),DIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
```

```
      COMMON/2/XL(6000)
      COMMON/A2/SB(22,26),AL7(22),U(22),V(22),WU(22),AQZ(22),AQ(22,3,22)
     1,Q1Z(22),Q1(22,9),W(11,12),P1(11,9),
     IP2(11,9),AL1(22,9),AL2(22,9),7(22),SMSC(22,22),
     3LMAX,A(11, 22),B(2,11,22),C(11,22),D(11,22),UA(6000)
      BANK(0),/2/,/A2/,OPINT,LINK2,ALOS,/A3/,SETUP,AQSJL
      BANK(1),/A1/
C     THIS SUBROUTINE MANUFACTURES ANGULAR DATA FOR THE MATRIX
C     COEFFICIENT CALCULATION
      K2=J2
      IF((IG.EQ.3).OR.(IG.EQ.-3)) K2=J1
      Z(1)=0.
      U(1)=-1.
      U(J1)=-1.
      WU(1)=1.
      WU(J1)=1.
      DO 4 J=2,K2
      U(J)=EMU(J)/3.+EMU(J-1)/6.
      V(J)=EMU(J)/6.+EMU(J-1)/3.
      WU(J)=.5
      Z(J)=(3.-EMU(J)*EMU(J)-EMU(J)*EMU(J+1)-EMU(J-1)*EMU(J-1))/(3.*Q1Z(
     1J))
      IF((IG.EQ.3).OR.(IG.EQ.-3)) GO TO 4
      M1=J1-J+1
      U(M1)=U(J)
      V(M1)=V(J)
      WU(M1)=.5
    5 Z(M1)=Z(J)
    4 CONTINUE
      RETURN
      END
      SUBROUTINE ALQS
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(15 ),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IG),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),IN),(
     2E(10),DELR),(E(500),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      COMMON/2/XL(6000)
      COMMON/A2/SB(22,26),AL7(22),U(22),V(22),WU(22),AQZ(22),AQ(22,3,22)
     1,Q1Z(22),Q1(22,9),W(11,12),P1(11,9),
     IP2(11,9),AL1(22,9),AL2(22,9),7(22),SMSC(22,22),
```

```
     3LMAX,A(11,22),B(2,11,22),C(11,22),D(11,22),UA(6000)
      BANK(0),/2/,/A2/,OPINT,LINK2,ALOS,/A3/,SETUP,AQSJL
      BANK(1),/A1/
      JMAX=JMAX
      J1=J2-1
      IF(LMAX) 14,14,11
   11 EM=-1.0
      DO 13 L=1,LMAX
      J=1
      Q1(1,L)=EM
      AL1(1,L)=(U(2)*P1(1,L)-P2(1,L))/W(1,1)
      AL2(JMAX,L)=EM*AL1(1,L)
      Q1(JMAX,L)=1.0
   15 J=J+1
      J3=JMAX-J+1
      IF(J-J2) 16,17,17
   16 AL1(J,L)=(U(J+1)*P1(J,L)-P2(J,L))/W(J,1)
      AL2(J,L)=(-U(J-1)*P1(J-1,L)+P2(J-1,L))/W(J-1,1)
      AL2(J3,L)=EM*AL1(J,L)
      AL1(J3,L)=EM*AL2(J,L)
      Q1(J,L)=P1(J-1,L)
      Q1(J3,L)=EM*Q1(J,L)
      GO TO 15
   17 AL2(J2,L)=(-U(J1)*P1(J1,L)+P2(J1,L))/W(J1,1)
      AL1(J2+1,L)=EM*AL2(J2,L)
      Q1(J2,L)=P1(J1,L)
      Q1(J2+1,L)=EM*Q1(J2,L)
   13 EM=-EM
   14 Q1Z(1)=1.0
      ALZ(1)=(U(2)-U(1))*0.5
      Q1Z(J2)=U(J2)-U(J1)
      ALZ(J2)=Q1Z(J2)*0.5
      Q1Z(J2+1)=Q1Z(J2)
      ALZ(J2+1)=ALZ(J2)
      Q1Z(JMAX)=1.0
      ALZ(JMAX)=ALZ(1)
      J=2
   21 IF(J-J2) 22,23,23
   22 J3=JMAX-J+1
      Q1Z(J)=U(J)-U(J-1)
      ALZ(J)=(U(J+1)-U(J-1))*0.5
      Q1Z(J3)=Q1Z(J)
      ALZ(J3)=ALZ(J)
      J=J+1
      GO TO 21
   23 RETURN
      END
      SUBROUTINE AQSJL
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
```

```
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),DE),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NWGP),(LL(187),NWREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),TIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      COMMON/A2/XL(6000)
      COMMON/A2/SB(22,26),AL2(22),U(22),V(22),WU(22),AQZ(22),AQ(22,3,22)
     1,Q1Z(22),Q1(22,9),W(11,12),P1(11,9),
     1P2(11,9),AL1(22,9),AL2(22,9),Z(22),SMSC(22,22),
     3LMAX,A(11, 22),B(2,11,22),C(11,22),D(11,22),UA(6000)
      BANK(0),/2/,/A2/,OPINT,LINK2,ALOS,/A3/,SETUP,AOSJL
      BANK(1),/A1/
      IF(LMAX) 23,23,11
   11 DO 21 J=1,JMAX
      JA=1
   12 DO 13 L=1,LMAX
   13 AQ(JA,L,J)=AL1(JA,L)*Q1(J,L)
   14 JA=JA+1
      IF(JA-J2) 16,18,15
   15 IF(JA-JMAX) 16,18,18
   16 DO 17 L=1,LMAX
   17 AQ(JA,L,J)=(AL1(JA,L)+AL2(JA,L))*Q1(J,L)
      GO TO 14
   18 DO 19 L=1,LMAX
   19 AQ(JA,L,J)=AL2(JA,L)*Q1(J,L)
      IF(JA-JMAX) 20,21,21
   20 JA=JA+1
      GO TO 12
   21 CONTINUE
   23 RETURN
      END
C*********************SEGMENT 1*********************************************
      PROGRAM MTXSET
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(22,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),DE),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NWGP),(LL(187),NWREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),TIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
```

```
3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINY),(E(
410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11040),DELTA),(E
5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
612522),SVM)
 COMMON/2/XL(6000)
 COMMON/A2/SB(22,26),AL7(22),U(22),V(22),WU(22),AOZ(22),AO(22,3,22)
1,Q1Z(22),Q1(22,9),C1(11,11),B1(2,11,22),
2Z(22),SMSC(22,22),
3LMAX,A(11, 22),B(2,11,22),C(11,22),D(11,22),UA(6000)
 BANK(1),MTXSET,/A1/
 BANK(0),/2/,/A2/,/A3/
 COMMON/A3/,D1(2,11,11),DM(11,22)
 TYPE DOUBLE C1,DM,EN
 J11=J1+1
 J21=J2+1
 III=2
 DO 6000 IGRP=1,NGR
 I=IGRP
 IF(MADJ.GT.0)I=NGR-IGRP+1
 NAF=1
 AC67=(TIMEBEG-TIMELEFT(AC67))*.001
 IF(NGR.EQ.1) GO TO 34
 IF(IGRP.GT.NGR/2) III=8
 IF(IGRP.EQ.NGR/2+1) REWIND 2
34 DO 71 K=1,J2
 DO 74 L=1,J1
 B1(1,K,L) = 0.0
 B1(2,K,L) = 0.0
74 DM(K,L)=0.0
 DO 75 L=1,J2
 D1(1,K,L) = 0.0
 D1(2,K,L) = 0.0
75 C1(K,L) = 0.0
71 CONTINUE
 REWIND 3
 IZ=1
 EM5=.000001
 DO 202 K=1,J2
 L=K+J2
 IF((DELTA(K,I).GT.EM5).OR.(DELTA(L,I).GT.EM5))GO TO 201
202 CONTINUE
 DO 203 K=1,10
 IF((GAMMA(K,1,I).GT.EM5).OR.(GAMMA(K,2,I).GT.EM5))GO TO 201
203 CONTINUE
 IZ=0
 DO 204 K=1,J2
 L=K+J2
 SB(K,I)=0.
204 SB(L,I)=0.
201 NTCTR=0
 IF(IGRP.NE.1) GO TO 23
 MM5=3000
 M5=MM5
 MM6=3001
 MM3=3001
 MN4=6000
23 MM5=-MM5
 M5=-M5
 MN4=MN4+MM5
 MM4=MN4
 MM6=MM6+MM5
 MM3=MM3+M5
```

```
      M4=MM3
      DO11K=1,J2
      DO11L=1,J1
   11 B(1,K,L)=0.0
      IF(MFR.GT.0) GO TO 70
      B(1,1,1)=1.
      B(1,1,J1)=-ALPHA(1,I)
      DO1J=2,J2
      M6=J1-J+1
      B(1,J,J)=.5
      B(1,J,J-1)=.5
      B(1,J,M6)=-.5*ALPHA(1,I)
    1 B(1,J,M6+1)=-.5*ALPHA(1,I)
      GO TO 73
   70 B(1,1,1)=1.
      D1(1,1,1)= -ALPHA(1,I)
      DO 72 J=2,J2
      B(1,J,J)=.5
      B(1,J,J-1)=.5
      D1(1,J,J) = -.5*ALPHA(1,I)
   72 D1(1,J,J-1) = -.5*ALPHA(1,I)
      DM(J2,J1) = 1.0
      B1(1,J2,J1) = -ALPHA(2,I)
      DO 82 J=2,J2
      K3 = J1-J+1
      K2=J2-J+1
      B1(1,K2,K3)=-.5*ALPHA(2,I)
      B1(1,K2,K3+1) = -.5*ALPHA(2,I)
      DM(K2,K3) = .5
   82 DM(K2,K3+1) = .5
   73 J5=J2+1
      IF(BETA(1,I).EQ.0.)GO TO 200
      M7=J1-1
      DO2J=J5,M7
      EM=2.*Q1Z(J)
      M6=J1-J+1
      DO2K=1,J2
      B(1,K,J)=B(1,K,J)+(BETA(1,I)*EM*U(M6))
    2 B(1,K,J+1)=B(1,K,J+1)+(BETA(1,I)*EM*V(M6))
  200 IF(IZ.EQ.0) GO TO 205
      SB(1,I)=GAMMA(1,1,I)+DELTA(J1,I)
      SB(J1,I)=GAMMA(1,2,I)+DELTA(1,I)
      DO4J=2,J2
      M6=J1-J+1
    4 SB(M6,I)=GAMMA(1,2,I)+(DELTA(J,I)+DELTA(J-1,I))*.5
      EM=1.
      DO5J=1,J2
      M6=J1-J+1
      DO5L=1,9
      EM=-EM
      SB(J,I)=SB(J,I)+GAMMA(L+1,1,I)+Q1(M6,L)
    5 SB(M6,I)=SB(M6,I)+GAMMA(L+1,2,I)+Q1(J,L)*EM
  205 M7=1
      M1 = 1
      M2 = 2
C     PICK UP REGION INDEX
      DO22J5=1,JMAX
C     PICK UP PROPER MATERIAL NUMBER
      M6=MIR(J5)
      EM=SIGS(I,M6)
      IF(IBUK.EQ.5)EM=EM+BUCK(I+26,1)
      EM1=SIGS1(I,M6)
```

```
      DLTA1=.5*DELR(J5)
      DO 51 JA=1,J1
      DO 51 JJ=1,J1
   51 SMSC(JA,JJ)=(EM*AQZ(JA)+EM1*AQ(JA,1,JJ))*DLTA1
      EM3=SIGT(I,M6)*DLTA1
C     PICK UP PROPER POINT INDEX
      DO 60 JA=1,J2
      SOME=EM3*WU(JA)
      Q1(JA,1)=SOME+U(JA)
      Q1(JA,2)=SOME+V(JA)
      Q1(JA,3)=SOME-U(JA)
      Q1(JA,4)=SOME-V(JA)
   60 CONTINUE
      IF(J5-JMAX)801,802,802
  802 L=II(J5)
      GO TO 803
  801 L = II(J5) - 1
  803 DO 21 K = M7,L
      IF(K-MAX)6,7,7
    7 DO10K1=1,J2
      DO10JC=1,J1
   10 D(K1,JC)=0.0
      IF(MFR.NE.1) GO TO 81
      IF(K.NE.MAX) GO TO 81
      DO 69 J=1,J1
      DO 69 JC=1,J2
   69 D(JC,J)=D(JC,J) +DM(JC,J)
      GO TO 68
   81 D(J2,J1) = 1.0
      D(J2,1)=-ALPHA(2,I)
      DO12J=2,J2
      K3=J1-J+1
      K2=J2-J+1
      D(K2,K3)=.5
      D(K2,K3+1)=.5
      D(K2,J)=-.5*ALPHA(2,I)
   12 D(K2,J-1)=-.5*ALPHA(2,I)
   68 JC = J2 + 1
      IF(BETA(2,I).EQ.0.0) GO TO 602
      K2=J1-1
      DO13J=JC,K2
      EM=2.0*Q1Z(J)
      K3=J1-J+1
      DO13K1=1,J2
      D(K1,K3)=D(K1,K3)+(BETA(2,I)*EM*U(J))
   13 D(K1,K3-1)=D(K1,K3-1)+(BETA(2,I)*EM*V(J))
      GOTO602
C     BEGIN SETTING IN SCATTERING COEFFICIENTS
    6 DO 52 JC=1,J1
      DO 52 JR=1,J2
      JCI=J11-JC
      JRB=J11-JR
      JRD=J21-JR
      A(JR,JC)=B(M2,JR,JC)=-SMSC(JCI,JRB)
      C(JR,JC)=D(JR,JC)=-SMSC(JCI,JRD)
   52 CONTINUE
      DO 67 JT=1,J2
      CD1=Q1(JT,1)
      CO1=Q1(JT,2)
      CD2=Q1(JT,3)
      CO2=Q1(JT,4)
C     ADD IN GEOMETRIC AND TRANSPORT COMPONENTS WHERE NECESSARY
```

```
      JC=J11-JT
      JR=J21-JT
      A(JT,JT)=A(JT,JT)+CD1
      B(M2,JT,JT)=B(M2,JT,JT)+CD2
      C(JR,JC)=C(JR,JC)+CD1
      D(JR,JC)=D(JR,JC)+CD2
      IF(JT.LT.2) GO TO 67
      A(JT,JT-1)=A(JT,JT-1)+CO1
      B(M2,JT,JT-1)=B(M2,JT,JT-1)+CO2
      C(JR,JC+1)=C(JR,JC+1)+CO1
      D(JR,JC+1)=D(JR,JC+1)+CO2
   67 CONTINUE
      IF(IG)648,602,602
  648 PRINT 503,I
  503 FORMAT(///33H MATRIX ELEMENTS -B,D,A,C  GROUP ,I2)
      PRINT 505,((B(M1,JA,JJ),JJ=1,J1),JA=1,J2)
  505 FORMAT(6(5X,E13.5))
      PRINT 505,((D(JA,JJ),JJ=1,J1),JA=1,J2)
      PRINT 505,((A(JA,JJ),JJ=1,J1),JA=1,J2)
      PRINT 505,((C(JA,JJ),JJ=1,J1),JA=1,J2)
C     FIRST DIVIDE ALL SUB DIAGONAL ELEMENTS BY DIAGONAL
  602 DO 647 K1=1,J1
      K2=K1+1
      IF(K1.GT.J2) GO TO 603
      M3=1
      EM4=1./B(M1,K1,K1)
      GO TO 604
  603 KA=K1-J2
      M3=KA+1
      EM4=1./D(KA,K1)
  604 IF(K1.GE.J2) GO TO 608
      DO 607 JC=K2,J2
      XL(M4)=-EM4*B(M1,JC,K1)
      DO 605 J=K2,J1
  605 B(M1,JC,J)=B(M1,JC,J)+XL(M4)*B(M1,K1,J)
      IF(MFR.NE.1)GO TO 607
      DO 606 J=1,J2
  606 D1(M1,JC,J)=D1(M1,JC,J)+XL(M4)*D1(M1,K1,J)
  607 M4=M4+1
  608 IF(K1.EQ.J1) GO TO 635
      DO 611 J=M3,J2
      XL(M4)=-EM4*D(J,K1)
      IF(K1.GT.J2) GO TO 610
      DO 609 JC=K2,J1
  609 D(J,JC)=D(J,JC)+XL(M4)*B(M1,K1,JC)
      IF(MFR.NE.1) GO TO 611
      IF(K.EQ.MAX) GO TO 611
      DO 612 JC=1,J2
      EN=XL(M4)*D1(M1,K1,JC)
      IF(K.EQ.MAX-1)GO TO 613
      C1(J,JC)=C1(J,JC)+EN
      GO TO 612
  613 C(J,JC)=C(J,JC)+EN
  612 CONTINUE
      GO TO 611
  610 IF(K1.EQ.J1) GO TO 611
      DO 614 JC=K2,J1
  614 D(J,JC)=D(J,JC)+XL(M4)*D(KA,JC)
      DO 615 JC=1,J1
  615 C(J,JC)=C(J,JC)+XL(M4)*C(KA,JC)
      IF(MFR.NE.1)GO TO 611
      IF((K1.EQ.J1).OR.(K.GE.MAX-1))GO TO 611
```

```
      DO 616 JC=1,J2
  616 C1(J,JC)=C1(J,JC)+XL(M4)*C1(KA,JC)
  611 M4=M4+1
  635 IF(K.EQ.MAX) GO TO 647
      DO 617 JC=1,J2
      XL(M4)=-EM4*A(JC,K1)
      IF(K1.GT.J2) GO TO 619
      DO 618 J=K2,J1
  618 A(JC,J)=A(JC,J)+XL(M4)*B(M1,K1,J)
      IF((MFR.NE.1).OR.(K.EQ.MAX)) GO TO 617
      DO 620 J=1,J2
      EN=XL(M4)*D1(M1,K1,J)
      IF(K.LT.MAX-1) GO TO 621
      B(M2,JC,J)=B(M2,JC,J)+EN
      GO TO 620
  621 D1(M2,JC,J)=D1(M2,JC,J)+EN
  620 CONTINUE
      GO TO 617
  619 IF(K1.EQ.J1)GO TO 624
      DO 622 J=K2,J1
  622 A(JC,J)=A(JC,J)+XL(M4)*D(KA,J)
  624 DO 623 J=1,J1
  623 B(M2,JC,J)=B(M2,JC,J)+XL(M4)*C(KA,J)
      IF(MFR.NE.1) GO TO 617
      IF(K.GE.MAX-1)GO TO 617
      DO 625 J=1,J2
  625 D1(M2,JC,J)=D1(M2,JC,J)+XL(M4)*C1(KA,J)
  617 M4=M4+1
      IF(MFR.NE.1)GO TO 647
      DO 626 JC=1,J2
      XL(M4)=-EM4*B1(M1,JC,K1)
      IF(K1.GT.J2)GO TO 627
      DO 628 J=K2,J1
  628 B1(M1,JC,J)=B1(M1,JC,J)+XL(M4)*B(M1,K1,J)
      DO 629 J=1,J2
  629 DM(JC,J)=DM(JC,J)+XL(M4)*D1(M1,K1,J)
      GO TO 626
  627 DO 634 J=K2,J1
  634 B1(M1,JC,J)=B1(M1,JC,J)+XL(M4)*D(KA,J)
      IF(K.EQ.MAX-1) GO TO 631
      DO 632 J=1,J1
      B1(M2,JC,J)=B1(M2,JC,J)+XL(M4)*C(KA,J)
      IF(J.GT.J2) GO TO 632
      DM(JC,J)=DM(JC,J)+XL(M4)*C1(KA,J)
  632 CONTINUE
      GO TO 626
  631 DO 633 J=1,J1
  633 DM(JC,J)=DM(JC,J)+XL(M4)*C(KA,J)
  626 M4=M4+1
  647 CONTINUE
  643 IF(IG.GT.0) GO TO 649
      PRINT 504,I,K
  504 FORMAT(///58H FACTORED MATRIX ELEMENTS, B, D, A, C, B1, D1, C1,  G
     1ROUP I2,2X,5HPOINTI4)
      PRINT 505,((B(M1,JA,JJ),JJ=1,J1),JA=1,J2)
      PRINT 505,((D(JA,JJ),JJ=1,J1),JA=1,J2)
      PRINT 505,((A(JA,JJ),JJ=1,J1),JA=1,J2)
      PRINT 505,((C(JA,JJ),JJ=1,J1),JA=1,J2)
      IF(MFR.NE.1)GO TO 649
      PRINT 505,((B1(M1,JA,JJ),JJ=1,J1),JA=1,J2)
      PRINT 505,((D1(M1,JA,JJ),JJ=1,J2),JA=1,J2)
      PRINT 505,((C1(JA,JJ),JJ=1,J2),JA=1,J2)
```

```
649 K2=1
    DO 601 J=1,J2
    DO 601 JC=1,J1
601 B1(M1,J,JC)=0.
    DO 703 J=1,J2
    DO 700 JA=J,J1
    UA(MM4)=B(M1,J,JA)
700 MM4=MM4-1
    IF(MFR.NE.1)GO TO 703
    IF(K.GT.MAX-1) GO TO 703
    DO 630 JA=1,J2
    UA(MM4)=D1(M1,J,JA)
630 MM4=MM4-1
703 CONTINUE
    K2=J2+1
    DO 705 J=1,J2
    JC=K2+J-1
    DO 706 JA=JC,J1
    UA(MM4)=D(J,JA)
706 MM4=MM4-1
    IF(K.EQ.MAX)GO TO 705
    DO 704 JA=1,J1
    UA(MM4)=C(J,JA)
704 MM4=MM4-1
    IF(MFR.NE.1)GO TO 705
    IF(K.G E.MAX-1)GO TO 705
    DO 701 JA=1,J2
    UA(MM4)=C1(J,JA)
701 MM4=MM4-1
705 CONTINUE
708 J=M1
    M1=M2
    M2=J
    DO 642 J=1,J2
    DO 642 JC=1,J2
    C1(J,JC)=0.
642 D1(M2,J,JC)=0
    AC68=TIMELEFT(AC68)
C   CHECK IF BUFFERS ARE FULL
    IF(K-MAX)709,711,711
709 IF((NTCTR+1.LT.NN(2)).AND.(K.LT.(NTCTR+1)*NN(1))) GO TO 21
    IF(NTCTR+1.EQ.NN(2)) GO TO 21
711 IF(UNIT,III)711,712
712 IF(UNIT,3)712,713
713 IF(UNIT,8)713,714
714 M4=M4-1
    IF(NGR.NE.1)GO TO 715
    BUFFER OUT(8,1)(XL(MM3),XL(M4))
715 BUFFER OUT(III,1)(XL(MM3),XL(M4))
    IF(IG)651,716,716
651 PRINT 501,I,M4,K,IDIM,IUDIM,IIDIM
501 FORMAT(29HXL WRITTEN ON 2 AND 8, GROUP ,I2,I6,12HWORDS, POINT,I4
   1,I5,I5,I5)
    PRINT 505,(XL(J),J=MM3,M4)
716 MM4=MM4+1
    IF(K.EQ.MAX)GO TO 21
    IF(NTCTR.EQ.NN(2)-2) GO TO 656
    BUFFER OUT(3,1)(UA(MM4),UA(MN4))
    IF(IG)655,656,656
655 PRINT 502,I,MM4,K
502 FORMAT(25H UA WRITTEN ON 10, GROUP ,I2,I6,12HWORDS, POINT,I4)
656 NTCTR=NTCTR+1
```

```
  740 NN(10)=MM4
      MM5==-MM5
      M5=-M5
      MM6=MM6+MM5
      MM3=MM3+M5
      M4=MM3
      MM4=MN4+MM5
      MN4=MM4
   21 CONTINUE
C     SET BEGINNING OF NEXT REGION
      M7=II(J5)
   22 CONTINUE
      AC69=(TIMEBEG-AC68)*.001
      PRINT 508,AC67,AC69,I
  508 FORMAT(7H TIME1=F9.3,7H TIME2=F9.3,6H GROUPI3)
      NCTR=NN(2)
      IF(NCTR.LT.3) GO TO 723
  735 IF(UNIT,3)735,726
  726 BACKSPACE 3
  723 IF(UNIT,III) 723,721
  721 IF(UNIT,8)721,722
  722 BUFFER OUT(III,1)(UA(MM4),UA(MN4))
      IF(NGR.NE.1)GO TO 736
      BUFFER OUT(8,1)(UA(MM4),UA(MN4))
  736 NCTR=NCTR-1
      IF(NCTR.EQ.0) GO TO 742
      MM5=-MM5
      MM6=MM6+MM5
      MN4=MN4+MM5
      MM4=NN(10)
  724 IF(UNIT,3)724,725,728
  725 IF(NCTR.EQ.1)GO TO 723
      M4=MN4-MM5
      M8=M4-NN(6)+1
  731 IF(UNIT,18)731,732
  732 BUFFER IN(3,1)(UA(M8),UA(M4))
      NN(10)=M8
      BACKSPACE 3
      GO TO 726
  742 REWIND 3
      GO TO 6000
  728 DO 734 ICTR=1,20
      BACKSPACE 3
      PRINT 500
  729 IF(UNIT,3)729,730
  730 BUFFER IN(3,1)(UA(MM4),UA(MN4))
  733 IF(UNIT,3)733,725,734
  734 CONTINUE
      PRINT 507
  507 FORMAT(*0 FAILED *)
      CALL EXIT
 6000 CONTINUE
      RETURN
  500 FORMAT(*0EOF OR PARITY ERROR IN MTXSET, TAPE 10, TRIED 20 TIMES*)
      END
C*********************SEGMENT 2*********************************************
      PROGRAM MTXSES
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
```

```
5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
 DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
426,25),BETA(2,26),LRATE(25)
 EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IG),(LL
5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
6(LL(166),NTMIX),(LL(186),NWGP),(LL(187),NWREG),(LL(188),LHREG),
7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
 EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
612522),SVM)
 COMMON/2/XL(6000)
 COMMON/A2/SB(22,26),AL2(22),U(22),V(22),WU(22),AQZ(22),AQ(22,3,22)
1,Q1Z(22),Q1(22,9),W(11,12),P1(11,9),
IP2(11,9),AL1(22,9),AL2(22,9),Z(22),SMSC(22,22),
3LMAX,A(11,22),B(2,11,22),C(11,22),D(11,22),UA(6000)
 BANK(1),MTXSES,/A1/
 BANK(0),/2/,/A2/,/A3/
 J11=J1+1
 J21=J2+1
 J3=J2-1
 III=2
 J11=J1+1
 J21=J2+1
 DO 6000 IGRP=1,NGR
 IF(NGR.EQ.1)GO TO 24
 IF(IGRP.GT.NGR/2) III=8
 IF(IGRP-1.EQ.NGR/2) REWIND 2
 AC67=(TIMEBEG-TIMELEFT(AC67))*.001
24 I=IGRP
 I=IGRP
 IF(MADJ.GT.0)I=NGR-IGRP+1
 REWIND 3
 NTCTR=0
 IF(IGRP.NE.1) GO TO 23
 MM5=3000
 M5=MM5
 MM6=3001
 MM3=3001
 MN4=6000
23 MM5=-MM5
 M5=-M5
 MN4=MN4+MM5
 MM4=MN4
 MM6=MM6+MM5
 MM3=MM3+M5
 M4=MM3
 EM5=.000001
 IZ=1
 DO 202 K=1,J2
 L=K+J2
 IF((DELTA(K,I).GT.EM5).OR.(DELTA(L,I).GT.EM5)) GO TO 201
```

```
202 CONTINUE
    DO 203 K=1,10
    IF(ABSF(GAMMA(K,1,I)).GT.EM5) GO TO 201
    IF(ABSF(GAMMA(K,2,I)).GT.EM5) GO TO 201
203 CONTINUE
    IZ=0
    DO 204 K=1,J2
    L=K+J2
204 SB(K,I)=SB(L,I)=0.0
201 DO 11 K=1,J3
    DO11L=1,J1
 11 B(1,K,L)=0.0
    DO1J=2,J2
    M6=J1-J+1
    B(1,J-1,J-1)=U(M6+1)
    B(1,J-1,J)=V(M6+1)
    B(1,J-1,M6+1)=-ALPHA(1,I)*U(M6+1)
  1 B(1,J-1,M6)=B(1,J-1,M6)-ALPHA(1,I)*V(M6+1)
    IF(BETA(1,I).EQ.0.0) GO TO 200
    DO2J=2,J2
    M6=J1-J+1
    DO2K=1,J3
    M7=J1-K+1
    EN=(EMU(M7)+EMU(M7-1))*BETA(1,I)*Q1Z(J)
    B(1,K,M6)=B(1,K,M6)+U(J)*EN
  2 B(1,K,M6+1)=B(1,K,M6+1)+V(J)*EN
200 IF(IZ.EQ.0) GO TO 205
    SB(J1,I)=GAMMA(1,2,I)*DELTA(1,I)
    DO4J=2,J2
    M6=J1-J+1
    SB(M6,I)=GAMMA(1,2,I)*(U(J)+V(J))+U(J)*DELTA(J,I)
   1+V(J)*DELTA(J-1,I)
  4 SB(J-1,I)=GAMMA(1,1,I)*(U(M6+1)+V(M6+1))+U(M6+1)*DELTA(M6+2,I)
   1+V(M6+1)*DELTA(M6+1,I)
    EM=-1
    DO6L=1,9
    EM=-EM
    DO5J=2,J2
    M6=J1-J+1
    EN=P2(J-1,L)/Q1Z(J)
    SB(M6,I)=SB(M6,I)+GAMMA(L+1,2,I)*EN
  5 SB(J-1,I)=SB(J-1,I)+GAMMA(L+1,1,I)*EN*EM
  6 SB(J1,I)=SB(J1,I)+GAMMA(L+1,2,I)*Q1(1,L)
C   NOW GO INTO THE GENERAL MATRIX ROUTINE
205 M7=1
    M1=1
    M2=2
C   PICK UP REGION INDEX
    DO22J5=1,JMAX
C   PICK UP PROPER MATERIAL NUMBER
    M6=MIR(J5)
    DELRR=1.047197755*DELR(J5)
    DELRRT=2.*DELRR
    DO 60 JT=1,J2
    L5=J11-JT
    Q1(JT,1)=SIGT(I,M6)*WU(JT)
    Q1(L5,1)=SIGT(I,M6)*WU(L5)
 60 CONTINUE
    EM4=SIGS(I,M6)
    EM5=SIGS1(I,M6)
    DO 51 JA=1,J3
    K5=J11-JA
```

```
      K6=K5+1
      DO 51 JJ=1,J1
      SMSC(JA,JJ)=EM4*AQZ(JA)+EM5*AQ(JA,1,JJ)
   51 SMSC(K5,JJ)=EM4*AQZ(K6)+EM5*AQ(K6,1,JJ)
      DO 14 JJ=1,J1
   14 SMSC(J2,JJ)=EM4*(AQZ(J2)+AQZ(J21))+EM5*(AQ(J2,1,JJ)+AQ(J21,1,JJ))
C     PICK UP PROPER POINT INDEX
      IF(J5-JMAX)804,802,802
  802 L=II(J5)
      GO TO 803
  804 L=II(J5)-1
  803 DO 21 K=M7,L
      IF(K-MAX)8,7,7
    7 DO10K1=1,J2
      DO10JC=1,J1
   10 D(K1,JC)=0.0
      D(J2,J1)=1.0
      D(J2,1)=-ALPHA(2,I)
      DO12J=2,J2
      K3=J1-J+1
      K2=J2-J+1
      D(K2,K3)=U(J)
      D(K2,K3+1)=V(J)
      D(K2,J-1)=-ALPHA(2,I)*V(J)
      IF(J-J2)15,12,12
   15 D(K2,J)=-ALPHA(2,I)*U(J)
   12 CONTINUE
      EM=1.-ALPHA(2,I)
      IF(EM)658,659,658
  658 D(1,J2)=EM*U(J2)
  659 IF(BETA(2,I).EQ.0.0) GO TO 602
      DO 16 J=2,J2
      K3=J11-J
      DO 17 K1=1,J3
      K2=J11-K1
      MM8=J21-K1
      EN=EMU(MM8)+EMU(MM8-1)
      EM=BETA(2,I)*Q1Z(K3+1)
      D(K1,J-1)=D(K1,J-1)-U(K3+1)*EN*EM
   17 D(K1,J)=D(K1,J)-V(K3+1)*EN*EM
      D(J2,J-1)=D(J2,J-1)-U(K3+1)*2.*EM
   16 D(J2,J)=D(J2,J)-V(K3+1)*2.*EM
      GOTO602
    8 EM3=XPT(K+1)
      EM4=XPT(K)
      EM5=EM3*EM3
      EM6=EM3*EM4
      EM7=EM4*EM4
      EM8=EM6+EM6
      DLTA1=DELRR*(EM5+EM8+3.*EM7)
      DLTA2=DELRR*(3.*EM5+EM8+EM7)
      BTA=(EM5+EM6+EM7)*4.1887902
      GMMA1=DELRRT*(EM3+EM4+EM4)
      GMMA2=DELRRT*(EM3+EM3+EM4)
C     BEGIN SETTING IN SCATTERING COEFFICIENTS
      DO53JC=1,J1
      JCI=J11-JC
      DO 52 JR=1,J2
      JRB=J11-JR
      JRD=J21-JR
      C(JR,JC)=-DLTA2*SMSC(JCI,JRD)
      D(JR,JC)=-DLTA1*SMSC(JCI,JRD)
```

```
      IF(JR-J2)740,53,53
  740 A(JR,JC)=-DLTA1*SMSC(JCI,JRB)
      B(M2,JR,JC)=-DLTA2*SMSC(JCI,JRB)
   52 CONTINUE
   53 CONTINUE
      DO67JT=1,J2
      JC=J11-JT
      L5=JC
      EM=Q1(JT,1)
      EN=Q1(JC,1)
      SOME=DLTA1*EM
      ASOME=DLTA1*EN
      SUM=DLTA2*EM
      ASUM=DLTA2*EN
      CD1=U(JT)*BTA
      CO1=V(JT)*BTA
      CR1=Z(JT)*GMMA1
      CR2=Z(JT)*GMMA2
      ACD1=U(L5)*BTA
      ACO1=V(L5)*BTA
      ACR1=Z(L5)*GMMA1
      ACR2=Z(L5)*GMMA2
C     ADD IN GEOMETRIC AND TRANSPORT COMPONENTS WHERE NECESSARY
      JR=J21-JT
      C(JR,JC)=C(JR,JC)+CD1+CR2+SUM
      D(JR,JC)=D(JR,JC)-CD1*CR1+SOME
      IF(JT-J2)66,69,69
   66 A(JT,JT)=A(JT,JT)-ACD1+ACR1+ASOME
      B(M2,JT,JT)=B(M2,JT,JT)+ACD1+ACR2+ASUM
      A(JT,JT+1)=A(JT,JT+1)-ACO1-ACR1+ASOME
      B(M2,JT,JT+1)=B(M2,JT,JT+1)+ACO1-ACR2+ASUM
   68 IF(JT-1)67,67,69
   69 C(JR,JC+1)=C(JR,JC+1)+CO1-CR2+SUM
      D(JR,JC+1)=D(JR,JC+1)-CO1-CR1+SOME
   67 CONTINUE
      IF(IG)648,602,602
  648 PRINT 503
      PRINT 505,((B(M1,JA,JJ),JJ=1,J1),JA=1,J3)
      PRINT 505,((D(JA,JJ),JJ=1,J1),JA=1,J2)
      PRINT 505,((A(JA,JJ),JJ=1,J1),JA=1,J3)
      PRINT 505,((C(JA,JJ),JJ=1,J1),JA=1,J2)
C     NOW FACTOR AND PLACE IN BUFFERS
  602 DO 647 K1=1,J1
      K2=K1+1
      IF(K1.GT.J3) GO TO 603
      M3=1
      EM4=1./B(M1,K1,K1)
      GO TO 604
  603 KA=K1-J3
      M3=KA+1
      EM4=1./D(KA,K1)
  604 IF(K1.GE.J3) GO TO 608
      DO 607 JC=K2,J3
      XL(M4)=-EM4*B(M1,JC,K1)
      DO 605 J=K2,J1
  605 B(M1,JC,J)=B(M1,JC,J)+XL(M4)*B(M1,K1,J)
  607 M4=M4+1
  608 IF(K1.EQ.J1) GO TO 635
      DO 611 J=M3,J2
      XL(M4)=-EM4*D(J,K1)
      IF(K1.GT.J3) GO TO 610
      DO 609 JC=K2,J1
```

```
609 D(J,JC)=D(J,JC)+XL(M4)*B(M1,K1,JC)
    GO TO 611
610 IF(K1.EQ.J1) GO TO 611
    DO 614 JC=K2,J1
614 D(J,JC)=D(J,JC)+XL(M4)*D(KA,JC)
    DO 615 JC=1,J1
615 C(J,JC)=C(J,JC)+XL(M4)*C(KA,JC)
611 M4=M4+1
635 IF(K.EQ.MAX) GO TO 647
    DO 617 JC=1,J3
    XL(M4)=-EM4*A(JC,K1)
    IF(K1.GT.J3) GO TO 619
    DO 618 J=K2,J1
618 A(JC,J)=A(JC,J)+XL(M4)*B(M1,K1,J)
    GO TO 617
619 IF(K1.EQ.J1) GO TO 624
    DO 622 J=K2,J1
622 A(JC,J)=A(JC,J)+XL(M4)*D(KA,J)
624 DO 623 J=1,J1
623 B(M2,JC,J)=B(M2,JC,J)+XL(M4)*C(KA,J)
617 M4=M4+1
647 CONTINUE
643 IF(IG)660,649,649
660 PRINT 504
    PRINT 505,((B(M1,JA,JJ),JJ=1,J1),JA=1,J3)
    PRINT 505,((D(JA,JJ),JJ=1,J1),JA=1,J2)
    PRINT 505,((A(JA,JJ),JJ=1,J1),JA=1,J3)
    PRINT 505,((C(JA,JJ),JJ=1,J1),JA=1,J2)
649 K2=1
    DO 703 J=1,J3
    JC=J
    DO 700 JA=JC,J1
    UA(MM4)=B(M1,J,JA)
700 MM4=MM4-1
703 CONTINUE
    K2=J2
    DO 705 J=1,J2
    JC=K2+J-1
    DO 706 JA=JC,J1
    UA(MM4)=D(J,JA)
706 MM4=MM4-1
    IF(K-MAX)701,705,705
701 DO 704 JA=1,J1
    UA(MM4)=C(J,JA)
704 MM4=MM4-1
705 CONTINUE
C   SWITCH B MATRIX
708 J=M1
    M1=M2
    M2=J
C   CHECK IF BUFFERS ARE FULL
    IF(K-MAX)710,711,711
710 IF((NTCTR+1.LT.NN(2)).AND.(K.LT.NN(1)*(NTCTR+1))) GO TO 21
    IF(NTCTR+1.EQ.NN(2)) GO TO 21
711 IF(UNIT,III)711,712
712 IF(UNIT,3)712,713
713 IF(UNIT,8)713,714
714 M4=M4-1
    IF(NGR.NE.1)GO TO 715
    BUFFER OUT(8,1)(XL(MM3),XL(M4))
715 BUFFER OUT(III,1)(XL(MM3),XL(M4))
    AC68=TIMELEFT(AC68)
```

```
      IF(IG)651,716,716
  651 PRINT 501,I,M4,K,IDIM,IUDIM,ILDIM
      PRINT 505,(XL(J),J=MM3,M4)
  716 MM4=MM4+1
      IF(K.EQ.MAX)GO TO 21
      IF(NTCTR.EQ.NN(2)-2) GO TO 656
      BUFFER OUT(3,1)(UA(MM4),UA(MN4))
      IF(IG)655,656,656
  655 PRINT 502,I,MM4,K
  656 NTCTR=NTCTR+1
  741 NN(10)=MM4
      MM5=-MM5
      M5=-M5
      MM6=MM6+MM5
      MM3=MM3+M5
      M4=MM3
      MM4=MN4+MM5
      MN4=MM4
   21 CONTINUE
C     SET BEGINING OF NEXT REGION
      M7=II(J5)
   22 CONTINUE
      NCTR=NN(2)
      IF(NCTR.LT.3) GO TO 723
      AC69=(TIMEBEG-AC68)*.001
      PRINT 510, AC67,AC69,I
  510 FORMAT(7H TIME1=F9.3,7H TIME2=F9.3,6H GROUPI3)
  735 IF(UNIT,3)735,726
  726 BACKSPACE 3
  723 IF(UNIT,III)723,721
  721 IF(UNIT,8)721,722
  722 BUFFER OUT(III,1)(UA(MM4),UA(MN4))
      IF(NGR.NE.1) GO TO 736
      BUFFER OUT(8,1)(UA(MM4),UA(MN4))
  736 NCTR=NCTR-1
      PRINT 508,I,MM4,MN4
  508 FORMAT(32H UA WRITTEN ON 12 AND 18, GROUP ,I2,6H, FROMI6,4H TO I6)
      IF(NCTR.EQ.0) GO TO 742
      MM5=-MM5
      MM6=MM6+MM5
      MN4=MN4+MM5
      MM4=NN(10)
  724 IF(UNIT,3)724,725,728
  725 IF(NCTR.EQ.1)GO TO 723
      M4=MN4-MM5
      M8=M4-NN(6)+1
  731 IF(UNIT,18)731,732
  732 BUFFER IN(3,1)(UA(M8),UA(M4))
      NN(10)=M8
      BACKSPACE 3
      GO TO 726
  742 REWIND 3
      GO TO 6000
  728 DO 734 ICTR=1,20
      BACKSPACE 3
      PRINT 500
  729 IF(UNIT,3)729,730
  730 BUFFER IN(3,1)(UA(MM4),UA(MN4))
  733 IF(UNIT,3)733,725,734
  734 CONTINUE
      PRINT 509
  509 FORMAT(*0 FAILED *)
```

```
      CALL EXIT
 6000 CONTINUE
      RETURN
  500 FORMAT(*0EOF OR PARITY ERROR ON TAPE 10 IN MTXSET*)
  501 FORMAT(28H XL WRITTEN ON 2 AND 8, GROUP,I2,I6,12HWORDS,POINT,I4)
  502 FORMAT(25H UA WRITTEN ON 10, GROUP ,I2,I6,12HWORDS, POINT,I4)
  503 FORMAT(///35H MATRIX ELEMENTS -B,D,A,C   I=1,NGR )
  504 FORMAT(///25H FACTORED MATRIX, B,D,A,C)
  505 FORMAT(6(5X,E13.5))
      END
C*********************SEGMENT 3*****************************************
      PROGRAM REVERSE
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SOM,AJ,S1,S2,COL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),DE),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      COMMON/2/XL(6000)
      COMMON/A2/SB(22,26),AL7(22),U(22),V(22),WU(22),AQZ(22),AQ(22,3,22)
     1,Q1Z(22),Q1(22,9),W(11,12),P1(11,9),
     IP2(11,9),AL1(22,9),AL2(22,9),7(22),SMSC(22,22),
     3LMAX,A(11, 22),B(2,11,22),C(11,22),D(11,22),UA(6000)
      BANK(0),/2/,/A2/,/A3/
      BANK(1),REVERSE,/A1/
C     THIS ROUTINE REVERSES THE GROUP ORDER OF THE COEFFICIENT MATRIX
C     NOTE, IF ONLY ONE GROUP, REVERSAL NOT NEEDED
      IF(NGR.EQ.1) GO TO 100
      REWIND 2
      REWIND 8
      REWIND 3
    1 IF(UNIT,2)1,2
    2 IF(UNIT,8)2,3
    3 IF(UNIT,3)3,4
    4 M1=NGR/2
      M2=NGR-M1
      M11=M1-1
C     FIRST, SKIP DOWN TO LAST GROUP ON TAPE 2
      M3=NN(2)+NN(2)
C     IF NGR IS EVEN, SAME NO. OF GROUPS ON EACH TAPE, OTHERWISE 8 HAS
C     ONE MORE GROUP
    6 M4=-3000
```

```
      M5=1
      M6=NN(3)
      IF(NGR.EQ.M1+M1)GO TO 10
      DO 9 I=1,M3
      BUFFER IN(8,1)(XL(M5),XL(M6))
      PRINT 501,M5,M6
      M8=M5
      M9=M6
      M4=-M4
      M5=M5+M4
      M6=NN(3)
      IF((I+1.EQ.NN(2)).AND.(NN(2).GT.1))M6=NN(4)
      IF(I.EQ.NN(2))M6=NN(5)
      IF((I.GT.NN(2)).AND.(NN(2).GT.1))M6=NN(6)
      PRINT 502,M8,M9
      M6=M6+M5-1
    7 IF(UNIT,3)7,8
    8 IF(UNIT,8)8,9
    9 BUFFER OUT(3,1)(XL(M8),XL(M9))
C     NOW, POSITION TAPE 8 AT LAST GROUP
   10 M7=M11+M3
      DO 11 I=1,M7
      BUFFER IN(2,1)(B1,B1)
   11 BUFFER IN(8,1)(A1,A1)
C     NOW, TRANSFER MATRACIES FROM TAPE 2 TO TAPE 3
      L=2
      K=3
   13 DO 19 J=1,M1
      DO 16 I=1,M3
      M6=NN(3)
      IF((I.EQ.NN(2)).AND.(NN(2).GT.1))M6=NN(4)
      IF(I.EQ.NN(2)+1)M6=NN(5)
      IF(I.GT.NN(2)+1)M6=NN(6)
      M6=M6+M5-1
  110 IF(UNIT,L) 110,111
  111 BUFFER IN (L,1)(XL(M5),XL(M6))
      PRINT 503,L,M5,M6
      M8=M5
      M9=M6
      M4=-M4
      M5=M5+M4
      PRINT 504,K,M8,M9
   14 IF(UNIT,K)14,15
   15 IF(UNIT,L)15,160
  160 BUFFER OUT(K,1)(XL(M8),XL(M9))
   16 CONTINUE
      IF((L.EQ.3).OR.(J.EQ.M1)) 180,38
   38 M7=M3+M3
      DO 18 I=1,M7
      BACKSPACE L
   17 IF(UNIT,L)17,18
   18 CONTINUE
  180 CONTINUE
   19 CONTINUE
      REWIND L
      REWIND K
C     NOW, TRANSFER MATRICIES FROM TAPE 8 TO TAPE 2
      IF(L.EQ.8) GO TO 20
      IF(L.EQ.3) GO TO 100
      L=8
      K=2
      GO TO 13
```

```
C     NOW, TRANSFER FROM 3 TO 8
   20 M1=M2
      K=8
      L=3
      GO TO 13
  100 RETURN
  501 FORMAT(15H  IN ON 8, FROM I6,2HTO I6)
  502 FORMAT(15H OUT ON 3, FROM I6,2HTO I6)
  503 FORMAT( 7H  IN ONI2,4HFROMI6,2HTO I6)
  504 FORMAT( 7H OUT ONI2,4HFROMI6,2HTO I6)
      END
C**********************OVERLAY 3*****************************************
      PROGRAM LINK 3
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEM1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),BC),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      COMMON/2/XL(6000)
      COMMON/A2/SB(22,26),ALZ(22),U(22),V(22),S(150,22),SLIS(189),X(150,
     122),EMAX,EMIN,MAXMAX,MAXMIN,UA2(3366)
      DIMENSION XBP(22,26)
      EQUIVALENCE(UA2(1),XBP)
      BANK(1),LINK3,SOURCE,/A1/
      BANK(0),/2/,/A2/,EXTRAP,CONV
C     THIS IS THE MAIN ITERATIVE BRANCH OF THE PROGRAM
      REWIND 4
      EX0=(TIMEBEG-TIMELEFT(EX0))*.001
      PRINT 514,EX0
      REWIND 3
      EX1=EX2=EX3=EX4=EX5=EX0=0.
  514 FORMAT(27H ITERATION BEGUN AT TIME = F9.3)
      IK=J2
      IF((IG.EQ.3).OR.(IG.EQ.-3))IK=J3
      NPI=8
      IF(LPG.EQ.2)NPI=4
      KREG=KREG
      EIGEN2=EIGM3
      NEXT=IEXOP+1
      IF(JSP.GT.0) NEXT=3
      NSOS=NSOS
```

```
      NFOS=NFOS
      IF(LGO)2000,2001,2001
 2001 IT=0
      EIGEN1=1.
      EIGEN2=1.
      KIT=1
 2000 III=2
      M7=1
      DO 38 J=1,JMAX
      EM=DELR(J)*.5
      EM3=EM
      A4=DELR(J)*1.047197551
      M6=II(J)-1
      DO 37 K=M7,M6
      IF(XABSF(IG).EQ.1) GO TO 36
      AM1=XPT(K+1)**2
      AM2=XPT(K+1)*XPT(K)
      AM3=XPT(K)**2
      AM4=AM2+AM2
      EM=A4*(AM1+AM4+3.*AM3)
      EM3=A4*(3.*AM1+AM4+AM3)
   36 UA2(400+K)=EM
      UA2(550+K)=EM3
   37 CONTINUE
   38 M7=II(J)
      IF(XABSF(IG).EQ.1)GO TO 43
      DO 42 J=2,J1
   42 ALZ(J)=.5*(EMU(J)-EMU(J-1))
   43 DO 44 J=2,J2
      M1=J1-J+1
      UA2(700+J)=.5*(EMU(M1)+EMU(M1+1))
   44 UA2(700+M1)=.5*(EMU(J)+EMU(J-1))
      MM5=1
      K3=-3000
      JJJ=-6
      NCE=0
      K1=NN(1)
      K2=NN(2)
      MM4=MM5+NN(3)-1
      IF(NGR.EQ.1)GO TO 109
      III=2
  109 IF(UNIT,III)109,110
  110 BUFFER IN(III,1)(XL(MM5),XL(MM4))
    1 IT=IT+1
      IF(IG)658,659,659
  658 PRINT 517,I,MM4
  659 DO 27 IJK1=1,NGR
      EX1=TIMELEFT(EX1)
      IF(MADJ)111,111,112
  112 I=NGR-IJK1+1
      M80=-1
      M81=0
      M82=NGR
      M83=NPS
      GO TO 113
  111 I=IJK1
      M80=1
      M81=-1
      M82=1
      M83=0
  113 M2=MAX+JMAX-1
  517 FORMAT(26H XL READ IN LINK 3, GROUP ,I2,15H NO. OF WORDS =I5)
```

```
C     START THE MAIN ITERATIVE LOOP
C     ZERO OUT THE SOURCE VECTOR
      DO 2 K=1,M2
      UA2(K)=0.
    2 SLIS(K)=0.0
C     IS THIS THE FIRST GROUP, IF SO, NO SLOWING IN SOURCE
      IF(IJK1-1)9,9,3
C     TEST TO DETERMINE WHETHER THERE IS ANY DOWNSCATTER
    3 IF(NDS)2003,2003,5
C     TEST TO DETERMINE FOR P-1 DOWNSCATTER
 2003 IF(NPS)9,9,5
C     THIS IS NOT THE FIRST GROUP, PROCEED TO CALCULATE
    5 M7=1
C     PICK UP REGION INDEX
      DO 8 K=1,JMAX
C     PICK UP MATERIAL INDEX
      M6=MIR(K)
C     SET UPPER REGION BOUNDARY
      L=II(K)
C     PICK UP POINT INDEX
      DO 7 J=M7,L
C     NO OF POINTS INCREASES BY ONE AS EACH BOUNDARY IS CROSSED.
      M5=K+J-1
C     PICK UP SLOWING OUT GROUP INDEX
      M4=I+M80
C     PICK UP TRANSFER VECTOR INDEX
      M3=I+M81
C     PROCEED TO CALCULATE SLOWING IN SOURCE-1 DOWN, 2 DOWN ETC.
      IF(NPS)2002,2002,2007
 2007 UA2(M5)=UA2(M5)+(SCFLUX(J,M4)*STR1(M3,M6))*.238732414
 2002 DO 6 M=1,NDS
C     SET DOWNSCATTER LENGTH IN TRANSFER VECTOR
      M1 = 26 - M
C     CALCULATE SOURCE
      SLIS(M5)=SLIS(M5)+(SC(J,M4)*STR(M3,M6))*.079577471
C     RESET TRANSFER VECTOR INDEX
      M3=M3+M1+M81
C     RESET SLOWING OUT INDEX
C     WAS THE LAST SLOWING OUT GROUP, GROUP 1
      IF(M4-M82)60,7,60
   60 M4=M4-M80
C     CONTINUE ON SOURCE FOR THIS POINT
    6 CONTINUE
C     NEXT POINT IN REGION
    7 CONTINUE
C     NEXT REGION
      M7=II(K)
    8 CONTINUE
      IF(IG)100,9,9
  100 PRINT 500,I
      PRINT 501,(SLIS(J),J=1,M2)
C     IF ITOUT=0, NO FISSIONS
    9 IF(ITOUT)12,12,10
C     ADD FISSION SOURCE POINT BY POINT
   10 M7=1
      DO 11 K=1,JMAX
      M6=MIR(K)
      L=II(K)
      IF(MADJ)46,46,126
   46 EN=CHI(I,M6)
      GO TO 39
  126 EN=VUSIG(I,M6)
```

```
   39 DO 135 J=M7,L
      M5=K+J-1
  135 SLIS(M5)=SLIS(M5)+EN*POWR1(M5)*.079577471
   11 M7=II(K)
      IF(IG)101,12,12
  101 PRINT 502,I
      PRINT 501,(SLIS(J),J=1,M2)
C     ADD IN FIXED SOURCE POINT BY POINT
   12 DO 13 K=1,M2
   13 SLIS(K)=SLIS(K)+SVM(K,I)*.079577471
      IF (IBUK-5) 1310,1311,1310
 1311 DO 1313 K=1,M2
 1313 SLIS(K)=SLIS(K)+BUCK(I+26,1)
 1310 IF(IG)102,14,14
  102 PRINT 503,I
      PRINT 501,(SLIS(J),J=1,M2)
C     BEGIN CALCULATION OF ANGULAR SOURCE VECTOR
   14 S1=IG
      IK=J2
      IF(XABSF(IG).EQ.3)IK=J3
      SUM=0.0
      M7=1
      DO 21 J=1,JMAX
      M6=II(J)-1
      DO 20 K=M7,M6
      EM=UA2(400+K)
      EM3=UA2(550+K)
      M5=K+J-1
C     CONTRIBUTION FROM LEFT SIDE OF INTERVAL
      SUM=SLIS(M5)*EM
C     CONTRIBUTION FROM RIGHT SIDE OF INTERVAL
      SUM=SUM+SLIS(M5+1)*EM3
C     EQUATE TO ANGULAR SOURCE AND PLACE IN VECTOR
      UA2(K)=UA2(M5)*EM+UA2(M5+1)*EM3
      IF(UA2(K).EQ.0.0) GO TO 17
      S(K,J1)=SUM*UA2(K)
      IF(IK.EQ.J2)S(K+1,1)=SUM+UA2(K)
      DO 210 L=2,J2
      M1=J1-L+1
      L1=L-J2+IK
      S(K+1,L1)=SUM+UA2(700+M1)*UA2(K)
  210 S(K,M1)=SUM+UA2(700+M1)*UA2(K)
      GO TO 20
   17 IK=J2
      IF((IG.EQ.1).OR.(IG.EQ.-1))GO TO 18
      IK=J3
      S(K,J1)=SUM
   18 DO 19 L=1,IK
      S(K+1,L)=SUM
      M1=L+IK
      S(K,M1)=SUM
   19 CONTINUE
   20 CONTINUE
   21 M7=II(J)
C     THE SOURCES ARE CALCULATED, PROCEED TO CALCULATE FLUX
C     ADD IN BOUNDARY SOURCES
      DO 200 K=1,IK
      M1=K+IK
      S(1,K)=SB(K,I)
  200 S(MAX,M1)=SB(M1,I)
      IF((IG.EQ.3).OR.(IG.EQ.-3))S(MAX,J1)=SB(J1,I)
  137 IF(IG)103,104,104
```

```
  103 PRINT 504,I
      PRINT 501,((S(K,J),J=1,J1),K=1,MAX)
  104 EX2=TIMELEFT(EX2)
C
      CALL SOURCE
      EX3=TIMELEFT(EX3)
C     IF ONE ITERATION CHECK FOR OUTPUT CONTROL
      IF(IG)105,106,106
  105 PRINT 505
      PRINT 501,((S(K,J),J=1,J1),K=1,MAX)
      PRINT 506,I
      DO 300 L=1,MAX
  300 PRINT 507,XPT(L),SC(L,I),(X(L,K),K=1,J1)
  106 IF(ITOUT) 22,22,24
   22 IF(NOT-1) 27,23,23
   23 IF(MADJ)129,129,128
  128 IF(UNIT,3)128,130
  130 BUFFER OUT(3,1)(X(1,1),X(MAX,J1))
      GO TO 27
  129 IF(UNIT,4)129,220
  220 BUFFEROUT(4,1)(X(1,1),X(MAX,J1))
      GO TO 27
C     IF OUTER ITERATION PROBLEM, CHECK FOR CONVERGENCE
   24 IF(NCE) 27,27,25
C     IF CONVERGED, CHECK FOR OUTPUT CONTROL
   25 IF(NOT-1) 27,26,26
   26 IF(MADJ)133,133,131
  131 IF(UNIT,3)131,132
  132 BUFFER OUT(3,1)(X(1,1),X(MAX,J1))
      GO TO 27
  133 IF(UNIT,4)133,206
  206 BUFFEROUT(4,1)(X(1,1),X(MAX,J1))
   27 CONTINUE
  144 EX4=TIMELEFT(EX4)
   30 IF(ITOUT)32,32,29
C     MOVE OLD FISSION SOURCE
   29 DO 180 J=1,M2
  180 POWR2(J)=POWR1(J)
C     CALCULATE NEW FISSION SOURCE AND INTEGRATE
C     TEST FOR CONVERGENCE
      EX5=TIMELEFT(EX5)
   41 CALL CONV
      EX1=(TIMEBEG-EX1)*.001
      EX2=(TIMEBEG-EX2)*.001
      EX3=(TIMEBEG-EX3)*.001
      EX4=(TIMEBEG-EX4)*.001
      EX5=(TIMEBEG-EX5)*.001
      EX6=(TIMEBEG-TIMELEFT(EX6))*.001
      PRINT 515,EX1,EX2,EX3,EX4,EX5,EX6
C     IS PROBLEM CONVERGED OR OUT OF ITERATIONS OR FORCED
      IF((JSP.LT.1).OR.((NOT.GE.5).AND.(MADJ.GT.0))) GO TO 2010
 2011 IF (NCE) 4149,4149,2012
 4149 KIT=KIT+1
      IF(KIT-KIT1)2049,2012,2012
 2049 NK4=1
      GO TO 2019
 2012 KIT=0
      NK4=2
      LGO=-XABSF(LGO)
      S1=EIGEN1
      IF(JSP-1)2010,2013,2014
 2013 IF(NFOS)2015,2015,2016
```

```
2015 ALP=0.
     GO TO 2017
2016 ALP=CONC(NFOS)
2017 PRINT 3000,EIGEN1,NSOS,CONC(NSOS),NFOS,ALP
     GO TO 2018
2014 PRINT 3001,EIGEN1,KREG,DELR(KREG)
2018 AJ16=ABSF(EIGEN1-SEN)
3541 CONTINUE
     IF(AJ16-EPS3)2010,2019,2019
2019 IF(IT-ITOUT)2020,2020,712
 712 NCE=NCE+1
     GO TO 2010
2020 CONTINUE
2039 GO TO (2010,2021),NK4
2021 NCE=0
     LPG=1
     EIGM3=EIGEN2
     DO 6268 IJK1=1,189
     SLIS(IJK1)=0.0
6268 POWR2(IJK1)=0.0
     LL(197)=0
     RETURN
2010 IF(NCE-1)1,31,32
  31 IF(NOT-1)32,1,1
  32 DO 6269 IJK1=1,189
     SLIS(IJK1)=0.0
6269 POWR2(IJK1)=0.0
     LL(197)=4
 202 IF(MADJ)203,203,201
 203 IF(UNIT,4)203,204
 204 BUFFEROUT(4,1)(SC(1,1),SC(MAX,NGR))
     RETURN
 201 IF(UNIT,3) 201,251
 251 BUFFER OUT(3,1)(SC(1,1),SC(MAX,NGR))
     RETURN
 713 PRINT516,III
 516 FORMAT(29H0EOF OR PARITY ERROR ON TAPE ,I2)
     CALL Q8QERROR(0,4HBUG.)
 500 FORMAT(34H1JUST SLOWING IN SOURCE FOR GROUP I2)
3000 FORMAT (/33H CONCENTRATION SEARCH PARAMETERS /2X,12H EIGENVALUE=
    1E12.5,2X,6H CONC(I3,2H)=E14.5,2X,6H CONC(I3,2H)=E14.5  )
3001 FORMAT (/33H DIMENSION SEARCH PARAMETERS      /2X,12H EIGENVALUE=
    1E12.5,2X,6H DELR(I3,2H)=E14.5        )
 501 FORMAT(6(4X,E13.5))
 502 FORMAT(46H1SLOWING IN SOURCE + FISSION SOURCE FOR GROUP I2)
 503 FORMAT(48H1SLOWING IN + FISSION + FIXED SOURCES FOR GROUP I2)
 504 FORMAT(34H1 ANGULAR SOURCE VECTOR FOR GROUP I2)
 505 FORMAT (//46H SOURCE VECTOR  MULTIPLIED BY FACTORED MATRIX )
 506 FORMAT (///47H  CHECKOUT SCALAR AND DIRECTIONAL FLUXES GROUP=I3)
 507 FORMAT (5X,8(1X,E13.5))
 515 FORMAT(7H TIME1=F9.3,7H TIME2=F9.3,7H TIME3=F9.3,7H TIME4=F9.3,
    17H TIME5=F9.3,7H TIME6=F9.3)
     END LINK 3
     SUBROUTINE SOURCE
     COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
    1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
    2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
    3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
    4),XK(3),VX(3),TIMEREG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
    5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
     DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(22,26),
    1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
```

```
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IG),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHRBG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      COMMON/2/XL(6000)
      COMMON/A2/SB(22,26),ALZ(22),U(22),V(22),S(150,22),SLIS(189),X(150,
     122),EMAX,EMIN,MAXMAX,MAXMIN,UA2(3366)
      DIMENSION XBP(22,26)
      EQUIVALENCE(UA2(1),XBP)
      BANK(0),/2/,/A2/,EXTRAP,CONV
      BANK(1),LINK3,SOURCE,/A1/
C     SUBROUTINE SOURCE
      IF((IG.EQ.1).OR.(IG.EQ.-1))IK=J2
      IF((IG.EQ.3).OR.(IG.EQ.-3))IK=J3
      M4=MM4
      M5=MM5
      K3=-K3
      MM5=MM5+K3
      MM4=MM4+K3
      NCR=1
      M=1
    1 IF(UNIT,III)1,2,713
    2 IF(NCR+1-K2)701,691,695
  695 MM4=NN(5)+MM5-1
  697 IF(UNIT,III)697,104,713
  104 BUFFER IN(III,1)(XL(MM5),XL(MM4))
      J5=J5+1
      IF(IG)100,101,101
  100 PRINT 501,I,NCR,M4,M5,MM4,MM5
      M3=MM5-K3
      PRINT 503,(XL(J),J=M3,M4)
  101 NCTR=1
      GO TO 711
  691 MM4=NN(4)+MM5-1
  701 IF(UNIT,III)701,702,713
  702 BUFFER IN(III,1)(XL(MM5),XL(MM4))
      J5=J5+1
      IF(IG)102,103,103
  102 PRINT 502,I,NCR,M4,M5,MM4,MM5
      PRINT 503,(XL(J),J=M5,M4)
  103 NCR=NCR+1
  711 CONTINUE
  698 M3=M5
  699 DO 12 L=2,J1
  703 K=L-1
  704 DO 10 JJ=L,J1
      S(M,JJ)=S(M,JJ)+XL(M3)*S(M,K)
   10 M3=M3+1
```

```
      IF(M-MAX)705,12,12
  705 DO 706 JJ=1,IK
      S(M+1,JJ)=S(M+1,JJ)+XL(M3)*S(M,K)
  706 M3=M3+1
      IF(MFR.NE.1)GO TO 12
      DO 22 JJ=1,J2
      S(MAX,JJ+J2)=S(MAX,JJ+J2)+XL(M3)*S(M,K)
   22 M3=M3+1
   12 CONTINUE
      IF(M-MAX)714,13,13
  714 DO 715 JJ=1,IK
      S(M+1,JJ)=S(M+1,JJ)+XL(M3)*S(M,J1)
  715 M3=M3+1
      IF(MFR.NE.1)GO TO 13
      DO 21 JJ=1,J2
      S(MAX,JJ+J2)=S(MAX,JJ+J2)+XL(M3)*S(M,J1)
   21 M3=M3+1
   13 M=M+1
  707 IF(M-MAX)708,708,709
  708 IF(M3-M4)699,710,710
  709 GO TO 832
  710 M4=MM4
      M5=MM5
      K3=-K3
      MM5=MM5+K3
      MM4=MM4+K3
      GO TO 2
C     SUBROUTINE FLUX
  832 NCR=1
  200 IF(UNIT,3) 200,201
  201 IF(UNIT,4) 201,202
  202 M=MAX
      SUM=0.
      M5=MM5
      M4=MM4
      K3=-K3
      MM5=MM5+K3
      MM4=NN(6)+MM5-1
  840 IF(UNIT,III)840,789,819
  789 IF(K2-NCR)790,790,791
  790 IF(NGR.EQ.1) GO TO 680
      IF(IJK1.NE.NGR/2) GO TO 680
      REWIND 2
      III=8
  680 IF(IJK1-NGR) 792,28,28
   28 REWIND III
      IF(NGR.NE.1) GO TO 29
      JJJ=-JJJ
      III=III+JJJ
      GO TO 31
   29 III=2
   31 MM4=MM5+NN(3)-1
   32 IF(UNIT,III)32,33
   33 BUFFER IN(III,1)(XL(MM5),XL(MM4))
      GO TO 811
  792 MM4=NN(3)+MM5-1
  795 IF(UNIT,III)795,796,819
  796 BUFFER IN(III,1)(XL(MM5),XL(MM4))
      IF(IG)105,106,106
  105 PRINT 505,I,NCR,M4,M5,MM4,MM5
      PRINT 503,(XL(JC),JC=M5,M4)
  106 GO TO 811
```

```
791 IF(UNIT,III)791,797,819
797 BUFFER IN(III,1)(XL(MM5),XL(MM4))
    J5=J5+1
    NCR=NCR+1
    IF(IG)107,811,811
107 PRINT 506,I,NCR,M4,M5,MM4,MM5
    PRINT 503,(XL(JC),JC=M5,M4)
811 CONTINUE
798 M3=M5
808 M1=J1+1
799 DO 15 L=1,J1
    IF(M.GE.MAX)GO TO 807
    IF(MFR.NE.1)GO TO 805
    IF(M.GE.MAX-1)GO TO 805
    DO 23 JJ=1,J2
    K=J2-JJ+1
    SUM=SUM+X(MAX,K)*XL(M3)
 23 M3=M3+1
805 IF(L-J2-1)806,809,809
806 DO 16 JJ=1,J1
    K=J1-JJ+1
    SUM=SUM+X(M+1,K)*XL(M3)
 16 M3=M3+1
    GO TO 807
809 IF(M.NE.MAX-1) GO TO 807
    IF(MFR.NE.1) GO TO 807
    DO 24 JJ=1,J2
    K=J2-JJ+1
    SUM=SUM+X(MAX,K)*XL(M3)
 24 M3=M3+1
807 DO 14 JJ=M1,J1
    K=J1-JJ+M1
    SUM=SUM+X(M,K)*XL(M3)
 14 M3=M3+1
    M1=M1-1
    X(M,M1)=(S(M,M1)-SUM)/XL(M3)
    SUM=0.0
 15 M3=M3+1
    IF(M-1)816,816,815
815 M=M-1
    IF(M3-M4)808,808,818
816 DO 828 M=1,MAX
825 SUM=0.0
    SOME=0.0
    IK2=1
    KG=XABSF(IG)
    IF(KG.EQ.3) IK2=2
    DO 827 J=IK2,J1
    IF(KG.EQ.3) GO TO 30
    SUM=SUM+ALZ(J)*X(M,J)
    GO TO 827
 30 SUM=SUM+ALZ(J)*(X(M,J)+X(M,J-1))
827 CONTINUE
    IF(NPS.EQ.0) GO TO 833
    DO 831 J=2,J2
    M3=J1-J+1
    EM=EMU(J)-EMU(J-1)
    IF(KG.EQ.1) GO TO 34
    EM3=-V(M3+1)
    EM4=-U(M3+1)
    GO TO 35
 34 EM3=U(J)
```

```
      EM4=V(J)
   35 SOME=SOME+EM*(U(J)*X(M,M3)+V(J)*X(M,M3+1)-EM3*X(M,J)-EM4*X(M,J-1))
  831 CONTINUE
      SCFLUX(M,I)=SOME*6.28318531
  833 SC(M,I)=SUM*      6.28318531
  828 CONTINUE
  830 RETURN
  818 M4=MM4
      M5=MM5
      K3=-K3
      MM5=MM5+K3
      MM4=NN(6)+MM5-1
      GO TO 840
  819 PRINT 500,III
      CALL Q8QERROR(0,4HBUG.)
  713 PRINT 500,III
      CALL Q8QERROR(0,4HBUG.)
  501 FORMAT(29H UPPER READ IN SOURCE, GROUP I2,I4,I6,I6,I6,I6)
  502 FORMAT(29H LOWER READ IN SOURCE, GROUP I2,I4,I6,I6,I6,I6)
  503 FORMAT(6(4X,E13,5))
  504 FORMAT(29H0EOF OR PARITY ERROR ON TAPE I2,8H IN FLUX)
  505 FORMAT(27H LOWER READ IN FLUX, GROUP I2,I4,I6,I6,I6,I6)
  506 FORMAT(27H UPPER READ IN FLUX, GROUP I2,I4,I6,I6,I6,I6)
  500 FORMAT(29H0EOF OR PARITY ERROR ON TAPE I2,10H IN SOURCE)
      END
      SUBROUTINE EXTRAP
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IB),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      COMMON/2/XL(6000)
      COMMON/A2/SB(22,26),ALZ(22),U(22),V(22),S(150,22),SLIS(189),X(150,
     122),EMAX,EMIN,MAXMAX,MAXMIN,UA2(3366)
      DIMENSION XBP(22,26)
      EQUIVALENCE(UA2(1),XBP)
      BANK(1),LINK3,SOURCE,/A1/
      BANK(0),/2/,/A2/,EXTRAP,CONV
      DIMENSION XBP1(22,26),XBP2(22,26)
      NCHMAX=4
      GO TO (30,19,10),NEXT
```

```
 30 IF(IT-NPI)10,90,40
 90 GO TO(61,62),M
 62 SIGO=(EIGEN-1.)/(EIGM1-1.)
    GO TO 125
 61 SIGO=(EMAX-EMIN)/(EIGMX1-EIGMN1)
125 IF(SIGO) 200,200,131
131 IF(SIGO-1.) 126,126,200
126 AL=2./(2.-SIGO)
    EM1=2.0/SIGO-1.0
    SIGOT=0.
 70 NCH=1
    NCHT=NCHMAX
    NSIGT=1
    GO TO 156
 40 NCH=NCH+1
    IF(NCH-NCHT  )50,50,60
 60 IF(SIGOT) 70,70,80
 80 SIGO=SIGOT
    GO TO 131
 50 IF(NCH-3)51,100,110
 51 GO TO(140,141),M
141 CEIG=(EMAX+EMIN)*.5-1.
    GO TO 210
140 CEIG=EMAX-EMIN
    GO TO 210
100 CH=EM1
    CH1=1
    GO TO 210
110 OLDC=CH1
    CH1=CH
    CH=2.0*EM1*CH-OLDC
160 GO TO(161,162),M
162 ER=((EMAX+EMIN)*.5-1.)/CEIG
    GO TO 181
161 ER=(EMAX-EMIN)/CEIG
181 BEO=ER*CH
    IF(BEO-1.) 183,182,182
182 CSHIBO=LOGF(BEO+SQRT(BEO**2-1.0))
    ENCH=NCH-1
    XEM1=CSHIBO/ENCH
    BEEO=((EXP(XEM1))+1./(EXP(XEM1)))*.5
    SIGOT=SIGO*(BEEO+1.)/2.
183 IF(ER-1.0)190,200,200
200 NPI=IT+2
201 NCH=0
    NSIGT=1
    GO TO 10
190 GO TO (191,210),NSIGT
191 IF(SIGOT)210,210,221
221 NCHT=NCH+4
260 NSIGT=2
210 EL=NCH
    G=LOGF(EM1+SQRT(EM1**2-1.))
    XEM1=EL*G
    ENCH=((EXP(XEM1))+1./(EXP(XEM1)))*.5
    XEM1=(EL-1.0)*G
    XEM2=((EXP(XEM1))+1./(EXP(XEM1)))*.5
    AM=(4./SIGO)*XEM2/ENCH
    XEM1=(EL-2.0)*G
    XEM2=((EXP(XEM1))+1.0/(EXP(XEM1)))*.5
    BE=XEM2/ENCH
    GO TO 157
```

```
 10 SIGO=0.0
    GO TO 300
 19 IF(IT-1)10,10,20
 20 DO 21 K=1,M2
 21 POWR1(K)=POWR1(K)+THETA*(POWR1(K)-POWR2(K))
    GO TO 300
156 DO 158 K=1,M2
    POWR3(K)=POWR2(K)
158 POWR1(K)=POWR2(K)+AL*(POWR1(K)-POWR2(K))
    GO TO 300
157 DO 159 K=1,M2
    WOLD=POWR3(K)
    POWR3(K)=POWR2(K)
159 POWR1(K)=POWR2(K)+AM*(POWR1(K)-POWR2(K))+BE*(POWR2(K)-WOLD)
300 EIGMX1=EMAX
    EIGMN1=EMIN
    RETURN
    END
    SUBROUTINE CONV
    COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
   1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
   2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
   3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
   4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
   5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
    DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
   1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
   2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
   3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
   426,25),BETA(2,26),LRATE(25)
    EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
   1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
   2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
   3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
   420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IC),(LL
   5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
   6(LL(166),NTMIX),(LL(186),NWGP),(LL(187),NWREG),(LL(188),LWREG),
   7(LL(198),LWGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRAT )
    EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
   14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
   2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
   3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
   410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
   5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
   612522),SVM)
    COMMON/2/XL(6000)
    COMMON/A2/SB(22,26),AL7(22),U(22),V(22),S(150,22),SLIS(189),X(150,
   122),EMAX,EMIN,MAXMAX,MAXMIN,UA2(3366)
    DIMENSION XBP(22,26)
    EQUIVALENCE(UA2(1),XBP)
    BANK(1),LINK3,SOURCE,/A1/
    BANK(0),/2/,/A2/,EXTRAP,CONV
CCONV
C   THIS SUBROUTINE TESTS FOR CONVERGENCE
C   CALCULATE FISSION DENSITY AT EACH POINT
    M7=1
    IG=XABSF(IG)
    SOME=0.0
    DO 18 K=1,JMAX
    M6=MIR(K)
    L=II(K)
    DO 19 J=M7,L
```

```
      M5=K+J-1
      IF(J.EQ.M7) GO TO 10
      EM=UA2(399+J)
      EM3=UA2(549+J)
      GO TO 5
   10 EM=EM3=0.0
    5 SUM=0.0
      DO 20 I=1,NGR
      IF(MADJ.GT.0) GO TO 6
      SUM=SUM+SC(J,I)*VUSIG(I,M6)
      GO TO 20
    6 SUM=SUM+SC(J,I)*CHI(I,M6)
   20 CONTINUE
      POWR1(M5)=SUM
      SOME=SOME+POWR1(M5-1)*EM +SUM*EM3
   19 CONTINUE
      M7=II(K)
   18 CONTINUE
      EIGEN1=SOME
      IF(S2) 100,102,100
C     S2=0.0 IMPLIES THERE ARE FIXED SOURCES SO DONT NORMALIZE
  100 EIGEN1=EIGEN1/FAC
      M2=MAX+JMAX-1
      DO 101 K=1,M2
  101 POWR1(K)=POWR1(K)/EIGEN1
      M=1
      GO TO 105
  102 M=2
  105 IF(NOT-3)104,103,104
  103 PRINT 501,IT
      PRINT 502,(POWR1(K),K=1,M2)
  104 EMAX=1.0
      EIGM1=EIGEN
      EIGEN=EIGEN1/EIGEN2
      EMIN=1.
      DO 16  I=1,M2
      IF(POWR1(I)) 110,16,110
  110 SUM=POWR1(I)/POWR2(I)
      IF(EMAX-SUM) 13,14,14
   13 EMAX=SUM
      MAXMAX=I
   14 IF(SUM-EMIN) 15,16,16
   15 EMIN=SUM
      MAXMIN=I
   16 CONTINUE
      IF(LCO) 115,115,120
  115 SOME=ABSF((EMAX-EMIN)/EMAX)
      IF(SOME-EPS1) 50,50,40
  120 SOME=ABSF((EIGEN1-EIGEN2)/EIGEN1)
      IF(SOME-EPS1) 50,40,40
   40 EIGEN2=EIGEN1
   72 PRINT 500,IT,EIGEN1,EMAX,MAXMAX,EMIN,MAXMIN,SOME,EPS1
      IF(S2) 41,43,41
   41 IF(IT-3) 44,300,300
  300 CALL EXTRAP
      GO TO 44
   43 EIGEN 1=1.
   44 IF(IT-ITOUT) 45,51,51
   45 CONTINUE
   46 RETURN
   50 IF(IT-1) 40,40,51
C     MUST GO AT LEAST TWO ITERATIONS
```

```
   51 EIGEN2=EIGEN1
  201 PRINT 500,IT,EIGEN1,EMAX,MAXMAX,EMIN,MAXMIN,SOME,EPS1
   55 NCE=NCE +1
      GO TO 46
  500 FORMAT(56HOITER,  EIGENVALUE       EMAX      POINT      EMIN     POINT
     1,4X,22HCAL.=EPS,     INP.=EPS../2X,I2,3X,E12.5,1X,E12.5,I4,3X,
     2E12.5,I4,3X,E12.5,1X,E12.5)
  501 FORMAT(30H1  FISSION DENSITY-ITERATION  I2)
  502 FORMAT(8(1X,E12.5))
      END
C*************************OVERLAY 4*****************************************
      PROGRAM LINK 4
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LE,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SOM,AJ,S1,S2,OL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEREG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IG),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      EQUIVALENCE(UA,UA2)
      DIMENSIONUA2(400),SIGCAP(26,25),SIGCAV(650),SIGFIS(26,25),SIGFIV
     1(650),ESCAP(26,41),EEAK(40,26)
      COMMON/2/ESCAP1(1066),EEAK1(1040),BSC(40,26),BSV(40,26),SHAP(5),
     1DON(5),SIGTH(26,6),SIGSH(26,6),SIGS1H(26,6),STRH(234,6),STR1H(26,6
     2),VUSIGH(26,6)
      COMMON/A2/SB(22,26),AL7(22),U(22),V(22),X(150,22),CHIH(26,10),
     1XFLUX(150,22),SCPROD1(150),SCPROD2(150),SCPROD3(150),B1(22),B2(22)
     2,SCJF(40,26),SCJA(40,26),UA(400),SCPR1(40) ,SCPRJ(40),VUSIGH1(26,
     36),CHIH1(26),VOLH1(40),UA3(13),VOLH(10)
      EQUIVALENCE(ESCAP1,ESCAP,SIGFIS,SIGFIV),(EEAK1,EEAK,SIGCAP,SIGCAV)
      DIMENSION CCA(26),CCF(26),CCS(26),CCS1(26)
      BANK(0),/2/,/A2/
      BANK(1),LINK4,/A1/
C     THIS CHAIN PRINTS THE FINAL EIGENVALUE,FLUXES,POWER,AVERAGE FLUXES
C     AND REGION VOLUMES
C     PRINT LABEL
  102 IF(UNIT,4)102,103
  103 REWIND4
      REWIND 1
   40 IF(UNIT,3)40,41
   41 REWIND 3
      PRINT 500
C     WERE OUTER ITERATIONS REQUIRED
```

```
      IF(ITOUT) 4,4,1
C     IF SO, PRINT EIGENVALUE OR FISSION DENSITY
    1 IF(S2) 2,2,3
C     IF SOURCES ARE PRESENT -LABEL FINAL FISSION DENSITY
    2 PRINT 501,IT,EIGEN1
      GO TO 5
C     IF NO SOURCES, LABEL FINAL MULTIPLICATION FACTOR
    3 PRINT 502,IT,EIGEN1
      IF(OUTCON)1000,5,5
 1000 J=11678
      K=1
      JL=MAX+JMAX-1
 1001 KL=K+4
      IF(JL-KL)1004,1002,1002
 1002 NAB=KL-K+1
      J=J+5
      PUNCH 3000,NAB,J,(POWR1(I),I=K,KL)
      IF(JL-KL)5,5,1003
 1003 K=KL+1
      GO TO 1001
 1004 KL=JL
      GO TO 1002
C     IF NO OUTER ITERATIONS, LABEL ONE ITERATION PROBLEM
    4 PRINT 503
C     PRINT RADII, FLUXES AND POWER
    5 NBB=1
      NAB=NGR
      NCB=NGR-6
      IF(NCB)2000,2000,2001
 2000 NAB=NGR
      GO TO 2069
 2001 NAB=6
 2069 NBB1=NBB+1
      NBB2=NBB+2
      NBB3=NBB+3
      NBB4=NBB+4
      NBB5=NBB+5
      IF(MADJ)201,201,202
  201 PRINT 504,NBB,NBB1,NBB2,NBB3,NBB4,NBB5
      GO TO 203
  202 PRINT 519,NBB,NBB1,NBB2,NBB3,NBB4,NBB5
  203 M7=1
      DO 7 J=1,JMAX
      L=II(J)
      DO 6 M=M7,L
      K=J+M-1
C     PRINT POINT BY POINT AND REGION BOUNDARY TWICE
    6 PRINT 505,M,XPT(M),MIR(J),POWR1(K),(SC(M,M6),M6=NBB,NAB)
      PRINT 506
C     DESIGNATE REGION BOUNDARY
    7 M7=II(J)
      NBB=NAB+1
      IF(NCB)2080,2080,2070
 2070 NCB=NCB-6
      IF(NCB) 2071,2071,2072
 2071 NAB=NAB+NCB+6
      GO TO 2069
 2072 NAB=NAB+6
      GO TO 2069
C     INTEGRATE FLUX OVER EACH GROUP AND REGION AND CALCULATE NEUTRON
C     BALANCE
 2080 IF(NOT.LT.1)GO TO 50
```

```
 10 DO 14 IGROP=1,NGR
    IF(MADJ)104,104,112
112 ITAP=3
    IFORM=518
    I=NGR-IGROP+1
    GO TO 113
104 ITAP=4
    IFORM=508
    I=IGROP
113 IF(UNIT,ITAP)113,114
114 BUFFER IN(ITAP,1)(X(1,1),X(MAX,J1))
    IF(NOT.EQ.1) GO TO 105
    PRINT 507,I
    IF(IFORM.EQ.508) PRINT 508
    IF(IFORM.EQ.518) PRINT 518
105 IF(UNIT,ITAP)105,108,200
108 M7=1
    M9=I
    DO 21 J=1,JMAX
    L1=II(J)-1
    IF(J.EQ.JMAX)L1=II(J)
    DO 12 L=M7,L1
    IF((NOT.EQ.1).AND.((L.NE.M7).AND.(L.NE.MAX))) GO TO 12
    SUM=0.0
    SOME=0.0
    DO 11 M=2,J2
    M3=J1-M+1
    EM3=U(M)
    EM4=V(M)
    IF(IG.EQ.3)EM3=-V(M3+1)
    IF(IG.EQ.3)EM4=-U(M3+1)
    SUM=SUM+(EMU(M)-EMU(M-1))*(U(M)*X(L,M3)+V(M)*X(L,M3+1))*6.2831853
 11 SOME=SOME-(EMU(M3+1)-EMU(M3))*(EM3*X(L,M)+EM4*X(L,M-1))*6.2831853
    IF(L.EQ.M7)ESCAP(I,J)=SUM+SOME
    IF(L.EQ.II(JMAX))ESCAP(I,J+1)=SUM+SOME
    IF((IG.EQ.3).AND.(L.EQ.M7))ESCAP(I,J)=ESCAP(I,J)*XPT(L)*XPT(L)*
   X12.5663706
    IF((IG.EQ.3).AND.(L.EQ.II(JMAX)))ESCAP(I,J+1)=ESCAP(I,J+1)*XPT(L)*
   XXPT(L)*12.5663706
    IF(NOT.GE.11) GO TO 12
    IF(NOT.EQ.1) GO TO 12
801 PRINT 509,L,XPT(L),SC(L,M9),SUM,SOME,(X(L,K),K=1,4)
    IF(J1-4)12,12,13
 13 PRINT 510,(X(L,K),K=5,J1)
 12 CONTINUE
 21 M7=L1+1
 14 CONTINUE
C   CALCULATE REGION LEAKAGE
 20 REWIND 4
    REWIND 3
C   THE NET CURRENTS AT THE BOUNDARY ARE CALCULATED
C   NOW CALCULATE THE REGION LEAKAGE
    DO 27 I=1,NGR
    DO 27 J=1,JMAX
 27 EEAK(J,I)=ESCAP(I,J+1)-ESCAP(I,J)
C   NOW OBTAIN THE INTEGRATED FLUX IN EACH REGION AND GROUP
C   ALSO THE INTEGRATED FIXED SOURCE
    DO 32 M=1,NGR
    SUM=0.0
    SOME=0.0
    M7=1
    DO 31 I=1,JMAX
```

```
      EM3=.5*DELR(I)
      EM4=EM3
      L=II(I)-1
      DO 30 J=M7,L
      K=I+J-1
      IF(IG.EQ.3)EM3=DELR(I)*(XPT(J+1)*XPT(J+1)+2.*XPT(J+1)*XPT(J)+3.*
     1XPT(J)*XPT(J))*1.0471976
      IF(IG.EQ.3)EM4=DELR(I)*(3.*XPT(J+1)*XPT(J+1)+2.*XPT(J+1)*XPT(J)+
     1XPT(J)*XPT(J))*1.0471976
      SUM=SUM+SVM(K,M)*EM3+SVM(K+1,M)*EM4
   30 SOME=SOME+SC(J,M)*EM3+SC(J+1,M)*EM4
  301 BSC(I,M)=SOME
      BSV(I,M)=SUM
      SOME=0.0
      SUM=0.0
   31 M7=II(I)
   32 CONTINUE
C     NOW OBTAIN REACTION RATES AND PRINT NEUTRON BALANCE
      DO 101 L=1,5
  101 SHAP(L)=0.0
      DO 37 I=1,NGR
      J=I
      IF(MADJ.GT.0)J=NGR-I+1
      DO 100 L=1,5
  100 DON(L)=0.0
      PRINT 511,I
      IF (S2) 369,369,370
  369 PRINT 516
      GO TO 371
  370 PRINT 512
  371 DO 36 K=1,JMAX
      M6=MIR(K)
      M3=I
      M69=NGR-I
      M7=XMINOF(NDS,M69)
      SOME=SIGT(I,M6)-SIGS(I,M6)
      IF(M7)35,35,499
  499 DO 34 M=1,M7
      IF(M-NDS)33,33,35
   33 SOME=SOME-STR(M3,M6)
      M3=M3+26-M
   34 CONTINUE
   35 SOME=SOME+BSC(K,I)
      SUM=VUSIG(I,M6)+BSC(K,I)/EIGEN1
      PRINT 513,                    K,BSC(K,I),BSV(K,I),SOME,SUM,EEAK(K,I)
C     OBTAIN VALUES FOR WHOLE REACTOR
      DON(1)=DON(1)+BSC(K,I)
      DON(2)=DON(2)+BSV(K,I)
      DON(3)=DON(3)+SOME
      DON(4)=DON(4)+SUM
      DON(5)=DON(5)+EEAK(K,I)
   36 CONTINUE
      PRINT 514,(DON(M),M=1,5)
      DO 38 L=1,5
   38 SHAP(L)=SHAP(L)+DON(L)
   37 CONTINUE
      PRINT 515
      PRINT 514,(SHAP(M),M=1,5)
   50 REWIND 4
      REWIND 3
      REWIND 3
      IF(IBUK)4034,4000,4034
```

```
4034 IF(NOT-5)4001,4035,4035
4035 IF(MADJ)4000,4001,4001
4001 DO 4022 K=1,25
     DO 4002 J=1,JMAX
     IF(MIR(J).EQ.K) GO TO 4021
4002 CONTINUE
     GOTO4022
4021 DO4003I=1,NGR
     IF(SIGT(I,K))4033,4003,4033
4033 IF(IBUK-2)4008,4010,4011
4008 SIGT(I,K)=0.5*(SIGT(I,K)+SQRTF(SIGT(I,K)*SIGT(I,K)-4.0*BUCK(I,1)/3
    1.0))
     GO TO 4003
4010 SIGT(I,K)=0.5*(SIGT(I,K)+SQRTF(SIGT(I,K)*SIGT(I,K)-4.0*BUCK(K,1)/3
    1.0))
     GO TO 4003
4011 IF(IBUK-4)4014,4015,4003
4014 SIGT(I,K)=0.5*(SIGT(I,K)+SQRTF(SIGT(I,K)*SIGT(I,K)-4.0*BUCK(I,1)/3
    1.0))
     GO TO 4003
4015 SIGT(I,K)=0.5*(SIGT(I,K)+SQRTF(SIGT(I,K)*SIGT(I,K)-4.0*BUCK(I,K)/3
    1.0))
4003 CONTINUE
4022 CONTINUE
4000 AC69=TIMELEFT(AC69)*.001
5669 FORMAT (20H THIS CASE CONSUMED F9.3,24HSECONDS OF COMPUTER TIME)
     LGO=1
     LL(197)=0
     IF(NOT.EQ.6.OR.NOT.EQ.7.OR.NOT.EQ.9) GO TO 137
     IF(NOT.NE.5.AND.NOT.NE.8.AND.NOT.NE.10)136,135
 135 IF(MADJ)136,137,137
 137 LL(197)=4
 136 AC68=TIMELEFT(AC68)*.001
     AC68=TIMEBEG*.001-AC68
     PRINT 5669,AC68
     IF(NRATE)401,401,402
 401 RETURN
 402 PRINT 521,NRATE
 521 FORMAT(25H1 REACTION RATE SUMMARY - I3,19H NUCLIDES REQUESTBD)
     BUFFER IN (1,1)(SIGFIV(1),SIGFIV(650))
     BUFFER IN (1,1)(SIGCAV(1),SIGCAV(650))
     DO 403 I=1,MAX
     SUM=0.0
     DO 404 J=1,NGR
 404 SUM=SUM+SC(I,J)
 403 SCPROD1(I)=SUM
 405 IF(UNIT,1) 405,406,407
 407 PRINT 522
 522 FORMAT(*0EOF OR PARITY ERROR ON TAPE 10. READING FIS OF CAP )SECTS
    1*)
     CALL Q8QERROR(1,4HBUG.)
 406 DO 408 I=1,NRATE
     PRINT 523,LRATE(I)
     DO 409 J=1,MAX
     SUM=0.0
     SOME=0.0
 523 FORMAT(39H1 REACTION RATE SUMMARY FOR NUCLIDE NO. A6)
     DO 410 K=1,NGR
     SOME=SOME+SIGCAP(K,I)*SC(J,K)
 410 SUM=SUM+SIGFIS(K,I)*SC(J,K)
     SCPROD2(J)=SUM
 409 SCPROD3(J)=SOME
```

```
      PRINT 524
  524 FORMAT(23H0 FISSION RATE TRAVERSE/1H0,1X,4(5X,6HRADIUS,6X,4HRATE,4
     1X))
      M5=1
  420 M7=1
  412 DO 411 J=1,JMAX
      PRINT 525,J
  525 FORMAT(7H REGION,I3)
      M4=II(J)
      PRINT 526,(XPT(K),SCPROD2(K),K=M7,M4)
  526 FORMAT(2X,4(5X,F6.3,2X,E12.5))
  411 M7=M4+1
      GO TO(413,414)M5
  413 PRINT 527
  527 FORMAT(23H0 CAPTURE RATE TRAVERSE)
      GO TO 415
  414 PRINT 529
  529 FORMAT(*0 ONE-GROUP CAPTURE CROSS SECTION*)
  415 PRINT 528
  528 FORMAT(1H0,1X,4(5X,6HRADIUS,6X,4HRATE,4X))
      M7=1
      DO 416 J=1,JMAX
      PRINT 525,J
      M4=II(J)
      PRINT 526,(XPT(K),SCPROD3(K),K=M7,M4)
  416 M7=M4+1
      GO TO(417,408)M5
  417 M5=2
      PRINT 530
  530 FORMAT(*1  ONE-GROUP FISSION CROSS SECTION*)
      DO 418, K=1,MAX
      SCPROD2(K)=SCPROD2(K)/SCPROD1(K)
  418 SCPROD3(K)=SCPROD3(K)/SCPROD1(K)
      GO TO 420
  408 CONTINUE
      RETURN
  200 PRINT 517
      CALL QBQERROR(1,4HBUG,)
  517 FORMAT(*0EOF OR PARITY ERROR ON TAPE 4, READING IN ANGULAR FLUX
     1ES*)
 3000 FORMAT(I2,4X,I6,5E12.5)
  500 FORMAT(23H1    ****OUTPUT DATA****)
  501 FORMAT(18H0 FINAL ITERATION=I3,9X,26HFINAL INTEGRATED FISSIONS=
     1  E13.5)
  502 FORMAT(18H0 FINAL ITERATION=I3,9X,28HFINAL MULTIPLICATION FACTOR=
     1E 13.6)
  503 FORMAT(40H0    ONE ITERATION PROBLEM =NO EIGENVALUE)
  504 FORMAT(10H1     POINT,5X,6HRADIUS,3X,8HMATERIAL,3X,7HFISSION,6X,5HG
     1ROUPI2,5X,5HGROUPI2,5X,5HGROUPI2,5X,5HGROUPI2,5X,5HGROUPI2,5X,5HGR
     2OUPI2/4X,6HNUMBER,25X,7HDENSITY,7X,4HFLUX,4(8X,4HFLUX))
  505 FORMAT(6X,I3,4X,  E11.4,3X,I2,5X,E11.4, 6(1X,E11.4))
  506 FORMAT(3X,115(1H*))
  507 FORMAT(30H1  AUXILIARY OUTPUT FOR GROUP I2)
  508 FORMAT(//34X,42HHEMISPHERE-   HEMISPHERE-   ANGULAR FLUXES/
     12X,10HPT. RADIUS,8X,59HSCALAR FLUX   CURRENT-LEFT  CURRENT-RIGHT F
     2ROM MU=1. TO -1.)
  509 FORMAT(2X,I3,   8(1X,  E13.6))
  510 FORMAT (62X,  E13.6,1X,E13.6,1X,E13.6,1X,E13.6)
  511 FORMAT(1H1,23X,39HNEUTRON BALANCE CHARACTERISTICS  GROUP I2)
  512 FORMAT(1H0,14X,10HINTEGRATED,8X,10HINTEGRATED,9X,5HTOTAL,12X)
     15HTOTAL,13X,3HNET/18X,4HFLUX,10X,12HFIXED SOURCE,5X,11HABSORPTIONS
     2,6X,10HFISSIONS/K,9X,7HLEAKAGE)
```

```
  513 FORMAT (7HOREGION,I3,5(4X,  E13.6))
  514 FORMAT(1H0,8HTOTAL***,1X,5(4X,  E13.6))
  515 FORMAT(////24X,23HREACTOR NEUTRON BALANCE)
  516 FORMAT(1H0,14X,10HINTEGRATED,8X,10HINTEGRATED,9X,5HTOTAL,12X,
     15HTOTAL,13X,3HNET/18X,4HFLUX,10X,12HFIXED SOURCE,5X,11HABSORPTIONS
     2,6X,10HFISSIONS  ,9X,7HLEAKAGE)
  518 FORMAT(//35X,44HHEMISPHERE-   HEMISPHERE-   ANGULAR ADJOINTS/
     12X,10HPT. RADIUS,8X,60HSCALAR ADJOINT CURRENT-RIGHT CURRENT-LEFT
     2FROM MU=-1. TO 1.)
  519 FORMAT(10H1    POINT,5X,6HRADIUS,3X,8HMATERIAL,3X,7HFISSION,6X,5HG
     1ROUP I2,5X,5HGROUP I2,5X,5HGROUP I2,5X,5HGROUP I2,5X,5HGROUP I2,5X,5HGR
     2OUP I2/4X,6HNUMBER,25X,7HDENSITY,6X,7HADJOINT,5(5X,7HADJOINT))
      END LINK 4
C*********************OVERLAY 5*****************************************
      PROGRAM LINK 5
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,26),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),IC),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      EQUIVALENCE(UA,UA2)
      EQUIVALENCE(BSC1,BSC),(BSC1(1),VUSIGH),(BSC1(157),SIGS1H),(BSC1(31
     13),SIGSH),(BSC1(469),SIGTH)
      DIMENSIONUA2(400),SIGCAP(26,25),SIGCAV(650),SIGFIS(26,25),SIGFIV
     1(650),ESCAP(26,41),EEAK(40,26),BSC(40,26),VUSIGH(26,6),SIGS1H(26,6
     2),SIGSH(26,6),SIGTH(26,6)
      COMMON/2/ESCAP1(1066),EEAK1(1040),BSC1(1040),BSV(40,26),SHAP(5),
     1DON(5),STRH(234,6),STR1H(26,6),NPSH,NDSH
      COMMON/A2/SB(22,26),AL7(22),U(22),V(22),X(150,22),CHIH(26,10),
     1XFLUX(150,22),SCPROD1(150),SCPROD2(150),SCPROD3(150),B1(22), 2(22)
     2,SCJF(40,26),SCJA(40,26),UA(400),SCPR1(40) ,SCPRJ(40),VUSIGH1(26,
     36),CHIH1(26),VOLH1(40),UA3(13),VOLH(10)
      EQUIVALENCE(ESCAP1,ESCAP,SIGFIS,SIGFIV),(EEAK1,EEAK,SIGCAP,SIGCAV)
      DIMENSION CCA(26),CCF(26),CCS(26),CCS1(26)
      BANK(0),/2,/A2/,XLPCH,HOMOG
      BANK(1),LINK5,/A1/
      IJOUT=0
      IF(NOT.GT.10) IJOUT=1
      IF(NOT.GT.10) NOT=NOT-6
      REWIND 4
      REWIND 3
```

```
      IF(MADJ.LT.0) MADJ=-MADJ
      IF(NOT.EQ.6.OR.NOT.EQ.7.OR.NOT.EQ.9)131,1
  131 FTII=1./12.5663706
      DO 132 I=1,MAX
      DO 132 J=1,J1
  132 X(I,J)=FTII
      DO 137 I=1,MAX
      DO 137 J=1,NGR
  137 SC(I,J)=1.0
    1 IF(UNIT,4)1,2,60
    2 BUFFER IN(4,1)(XFLUX(1,1),XFLUX(MAX,J1))
      PRINT 500
      PRINT 501
      IF(NOT.NE.5.AND.NOT.NE.8.AND.NOT.NE.10) GO TO 130
    4 IF(UNIT,3)4,5,3
    5 DO 7 I=1,NGR
    6 IF(UNIT,3)6,7,3
    7 BUFFER IN(3,1)(X(1,1),X(1,1))
  203 IF(UNIT,3) 203,204,3
  204 BUFFER IN(3,1)(SC(1,1),SC(MAX,NGR))
  202 IF(UNIT,3)202,209,3
  209 BACKSPACE 3
      BACKSPACE 3
    8 IF(UNIT,3)8,9,3
    9 BUFFER IN(3,1)(X(1,1),X(MAX,J1))
  130 I=J2-1
      DO 10 J=1,I
      M7=J1-J+1
      B2(J)=1.047197551*(EMU(J+1)-EMU(J))
      B2(M7)=B2(J)
      B1(J)=2.*B2(J)
   10 B1(M7)=B1(J)
      K1=1
   12 IF(NOT.NE.5.AND.NOT.NE.8.AND.NOT.NE.10) GO TO 206
  351 IF(UNIT,3)351,13,3
   13 BACKSPACE 3
      BACKSPACE 3
  206 IF(UNIT,4)206,207,60
  207 M7=1
  134 DO 61 K=1,JMAX
      L=II(K)
      EMM=DELR(K)/3.
      SSUM=SSOM=0.0
      DO 62 J=M7,L
      IF(J.EQ.M7) GO TO 301
      IF(IG.NE.3) GO TO 300
      AM1=XPT(J)**2
      AM2=XPT(J-1)*XPT(J)
      AM3=XPT(J-1)**2
      A1=1.25663706*(AM1+3.*AM2+6.*AM3)
      A2=1.25663706*(3.*AM1+4.*AM2+3.*AM3)
      A3=1.25663706*(6.*AM1+3.*AM2+AM3)
      GO TO 301
  300 A1=A2=A3=1.0
  301 SUM1=SOME1=SAME1=SAM1=0.0
      DO 63 M=2,J2
      M3=J1-M+1
      EM=(EMU(M)-EMU(M-1)) * 6.28318531
      EN=(EMU(M3+1)-EMU(M3))*6.28318531
      EM1=EM*U(M)
      EM2=EM*V(M)
      IF(IG.EQ.3) GO TO 120
```

```
      EN1=EN*V(M)
      EN2=EN*U(M)
      GO TO 121
  120 EN1=-EN*U(M3+1)
      EN2=-EN*V(M3+1)
  121 SUM1=SUM1+EN1*X(J,M-1)+EN2*X(J,M)
      SAM1=SAM1+EM1*X(J,M3)+EM2*X(J,M3+1)
      SOME1=SOME1+EM1*XFLUX(J,M3)+EM2*XFLUX(J,M3+1)
      SAME1=SAME1+EN1*XFLUX(J,M-1)+EN2*XFLUX(J,M)
   63 CONTINUE
      SAM2=SUM1+SAM1
      SAME2=SOME1+SAME1
      IF(J.EQ.1) GO TO 601
      SSUM=SSUM+EMM*(A3*SAM2+SAME2+.5*A2*(SAM2*EM11+SAME2*EM10)+A1*EM10*
     1EM11)
      SSOM=SSOM+EMM*(A3*SAM2+BSC(26+J)+.5*A2*(SAM2*EM8+EM10+BSC(26+J))+A
     11*EM8*EM10)
  601 EM8=BSC(26+J)
      EM10=SAM2
      EM11=SAME2
   62 BSC(26+J)=SAME2
      SCJF(K,K1)=SSUM
      SCJA(K,K1)=SSOM
   61 M7=L+1
      IF(NOT.EQ.6) GO TO 742
      M3=J2-1
      DO 721 J=1,M3
      SCPROD1(J)=(3.*EMU(J)+EMU(J+1))/24.
      SCPROD2(J)=(EMU(J)+EMU(J+1))/24.
  721 SCPROD3(J)=(3.*EMU(J+1)+EMU(J))/24.
      IF(NOT.LT.9) GO TO 740
      DO 730 J=1,M3
      J9=J+12
      SUM=1.-EMU(J)*EMU(J)
      SOME=1.-EMU(J+1)*EMU(J+1)
      SCPROD1(J9)=0.5*(EMU(J+1)*SQRTF(SOME)+ASINF(EMU(J+1))-EMU(J)*SQRT
     1(SUM)-ASINF(EMU(J)))
      SCPROD2(J9)=(SUM*SQRTF(SUM)-SOME*SQRTF(SOME))/3.
      SCPROD3(J9)=(EMU(J)*SUM*SQRTF(SUM)-EMU(J+1)*SOME*SQRTF(SOME))/4.
      SCPROD3(J9)=(SCPROD3(J9)-2.*EMU(J)*SCPROD2(J9)+(.25*EMU(J)*EMU(J)
     1*SCPROD1(J9))/((EMU(J+1)-EMU(J))*(EMU(J+1)-EMU(J)))
      SCPROD2(J9)=(SCPROD2(J9)-EMU(J)*SCPROD1(J9))/(EMU(J+1)-EMU(J))
  730 CONTINUE
  740 M7=1
      N2=1
      DO 720 J7=1,NHREG
      N3=LHREG(J7)
      SUM=SOME=0.0
      DO 710 K=N2,N3
      L=II(K)-1
      DO 700 I=M7,L
      DO 700 J=1,M3
      M5=J1-J+1
      IF(NOT.LT.7.OR.NOT.GT.8) GO TO 741
      SOME= SOME - (SCPROD1(J)*(XFLUX(I,M5)*(2.*X(I,J)+X(I+1,J))+XFLUX(I
     1+1,M5)*(2.*X(I+1,J)+X(I,J))+XFLUX(I,J)*(2.*X(I,M5)+X(I+1,M5))+XFLU
     2X(I+1,J)*(2.*X(I+1,M5)+X(I,M5))))*B1(J)*DELR(K)
      SOME=SOME - (SCPROD2(J)*(XFLUX(I,M5)*(2.*X(I,J+1)+X(I+1,J+1))+XFLU
     1X(I+1,M5)*(2.*X(I+1,J+1)+X(I,J+1))+XFLUX(I,M5-1)*(2.*X(I,J)+X(I+1,
     2J)) + XFLUX(I+1,M5-1)*(2.*X(I+1,J)+X(I,J))+XFLUX(I,J+1)*(2.*X(I,M5
     3)+X(I+1,M5)) +XFLUX(I+1,J+1)*(2.*X(I+1,M5)+X(I,M5))+XFLUX(I,J)*(2.
     4*X(I,M5-1)+X(I+1,M5-1))+XFLUX(I+1,J)*(2.*X(I+1,M5-1)+X(I,M5-1))))*
```

```
     5B1(J)*DELR(K)
      SOME=SOME+ (SCPROD3(J)*(XFLUX(I,M5-1)*(2.*X(I,J+1)+X(I+1,J+1))+XFL
     1UX(I+1,M5-1)*(2.*X(I+1,J+1)+X(I,J+1))+XFLUX(I,J+1)*(2.*X(I,M5-1)+X
     2(I+1,M5-1))+XFLUX(I+1,J+1)*(2.*X(I+1,M5-1)+X(I,M5-1))))+B1(J)*DELR
     3(K)
741  IF(NOT.LT.9) GO TO 732
      J9=J+12
      SOME=SOME+(XFLUX(I+1,M5)*(2.*X(I+1,J)+X(I,J))+XFLUX(I,M5)*(X(I+1,J
     1)+2.*X(I,J))+XFLUX(I+1,J)*(2.*X(I+1,M5)+X(I,M5))+XFLUX(I,J)*(X(I+1
     2,M5)+2.*X(I,M5)))*SCPROD1(J9)*0.66666667*DELR(K)
      SOME=SOME+((2.*XFLUX(I+1,M5)+XFLUX(I,M5))*(X(I+1,J+1)-X(I+1,J))+(2
     1.*XFLUX(I,M5)+XFLUX(I+1,M5))*(X(I,J+1)-X(I,J))+(2.*X(I+1,J)+X(I,J)
     2)*(XFLUX(I+1,M5-1)-XFLUX(I+1,M5))+(2.*X(I,J)+X(I+1,J))*(XFLUX(I,M5
     3-1)-XFLUX(I,M5))+(2.*XFLUX(I+1,J)+XFLUX(I,J))*(X(I+1,M5-1)-X(I+1,M
     45))+(2.*XFLUX(I,J)+XFLUX(I+1,J))*(X(I,M5-1)-X(I,M5))+(2.*X(I+1,M5)
     5+X(I,M5))*(XFLUX(I+1,J+1)-XFLUX(I+1,J))+(2.*X(I,M5)+X(I+1,M5))*(XF
     6LUX(I,J+1)-XFLUX(I,J)))*SCPROD2(J9)*0.66666667*DELR(K)
      SOME=SOME+(3.*(XFLUX(I,M5-1)-XFLUX(I,M5))*(X(I+1,J+1)-X(I+1,J))+3.
     1*(X(I,J+1)-X(I,J))*(XFLUX(I+1,M5-1)-XFLUX(I+1,M5))+2.*(XFLUX(I,M5)
     2-XFLUX(I,M5-1)-XFLUX(I+1,M5)+XFLUX(I+1,M5-1))*(X(I,J)-X(I,J+1)-X(I
     3+1,J)+X(I+1,J+1))+3.*(XFLUX(I,J+1)-XFLUX(I,J))*(X(I+1,M5-1)-X(I+1,
     4M5))+3.*(X(I,M5-1)-X(I,M5))*(XFLUX(I+1,J+1)-XFLUX(I+1,J))+2.*(XFLU
     5X(I,J)-XFLUX(I,J+1)-XFLUX(I+1,J)+XFLUX(I+1,J+1))*(X(I,M5)-X(I,M5-1
     6)-X(I+1,M5)+X(I+1,M5-1)))*SCPROD3(J9)*0.66666667*DELR(K)
732  IF (NOT.NE.5.AND.NOT.NE.8.AND.NOT.NE.10) GO TO 700
      SUM=SUM+(SCPROD2(J)      )*((XFLUX(I,M5)+XFLUX(I+1,M5))*(X(I+1,J+1)
     1-X(I,J+1))+(XFLUX(I,M5-1)+XFLUX(I+1,M5-1))*(X(I+1,J)-X(I,J))+(XFLU
     2X(I,J+1)+XFLUX(I+1,J+1))*(X(I,M5)-X(I+1,M5))+(XFLUX(I,J)+XFLUX(I+1
     3,J))*(X(I,M5-1)-X(I+1,M5-1)))+B2(J)*6.
      SUM=SUM+(SCPROD1(J)          )*((XFLUX(I,M5)+XFLUX(I+1,M5))*(X(I+1,J
     1)-X(I,J))+(XFLUX(I,J)+XFLUX(I+1,J))*(X(I,M5)-X(I+1,M5)))*B2(J)*6.
      SUM=SUM+(SCPROD3(J)          )*((XFLUX(I,M5-1)+XFLUX(I+1,M5-1))*(X(I
     1+1,J+1)-X(I,J+1))+(XFLUX(I,J+1)+XFLUX(I+1,J+1))*(X(I,M5-1)-X(I+1,M
     25-1)))*B2(J)*6.
700  CONTINUE
710  M7=L+1
      BSV(J7,K1)=SOME
      L6=J7+6
      BSV(L6,K1)=SUM
720  N2=N3+1
742  M7=1
      DO 17 K=1,JMAX
      L=II(K)-1
      DO 16 I=M7,L
      SOME=0.0
      SUM=0.0
      SUME=0.0
      M3=J2-1
      DO 15 J=1,M3
      M5=J1-J+1
      SOME=SOME+B1(J)*(X(I,M5)*XFLUX(I,J)+X(I,M5-1)*XFLUX(I,J+1)+
     1X(I,J)*XFLUX(I,M5)+X(I,J+1)*XFLUX(I,M5-1))
      SUM=SUM+B1(J)*(X(I+1,M5)*XFLUX(I,J)+X(I+1,M5-1)*XFLUX(I,J+1)+
     1X(I+1,J)*XFLUX(I,M5)+X(I+1,J+1)*XFLUX(I,M5-1))
      SUME=SUME+B1(J)*(X(I,M5)*XFLUX(I+1,J)+X(I,M5-1)*XFLUX(I+1,J+1)+
     1X(I,J)*XFLUX(I+1,M5)+X(I,J+1)*XFLUX(I+1,M5-1))
      SOME=SOME+B2(J)*(X(I,M5)*XFLUX(I,J+1)+X(I,M5-1)*XFLUX(I,J)+
     1X(I,J)*XFLUX(I,M5-1)+X(I,J+1)*XFLUX(I,M5))
      SUM=SUM+B2(J)*(X(I+1,M5)*XFLUX(I,J+1)+X(I+1,M5-1)*XFLUX(I,J)+
     1X(I+1,J)*XFLUX(I,M5-1)+X(I+1,J+1)*XFLUX(I,M5))
 15   SUME=SUME+B2(J)*(X(I,M5)*XFLUX(I+1,J+1)+X(I,M5-1)*XFLUX(I+1,J)+
     1X(I,J)*XFLUX(I+1,M5-1)+X(I,J+1)*XFLUX(I+1,M5))
```

```
      SCPROD1(I)=SOME
      SCPROD2(I)=SUM
      SCPROD3(I)=SUME
   16 CONTINUE
   17 M7=II(K)
      SUM=0.0
      DO 24 J=1,M3
      M5=J1-J+1
      SUM=SUM+B1(J)*(X(MAX,M5)*XFLUX(MAX,J)+X(MAX,J)*XFLUX(MAX,M5)+
     1X(MAX,M5-1)*XFLUX(MAX,J+1)+X(MAX,J+1)*XFLUX(MAX,M5-1))
   24 SUM=SUM+B2(J)*(X(MAX,M5-1)*XFLUX(MAX,J)+X(MAX,J)*XFLUX(MAX,M5-1)+
     1X(MAX,M5)*XFLUX(MAX,J+1)+X(MAX,J+1)*XFLUX(MAX,M5))
      SCPROD1(MAX)=SUM
   18 IF(K1-NGR)136,20,20
  136 IF(NOT.EQ.6.OR.NOT.EQ.7.OR.NOT.EQ.9) GO TO 201
   19 IF(UNIT,3)19,21,3
   21 BUFFER IN(3,1)(X(1,1),X(MAX,J1))
  200 IF(UNIT,3)200,201,3
  201 M7=1
  208 IF(UNIT,4)208,205,60
  205 BUFFER IN(4,1)(XFLUX(1,1),XFLUX(MAX,J1))
      INDEX=1
      GO TO 31
   20 IF(UNIT,4)20,27,60
   27 BUFFER IN(4,1)(SCFLUX(1,1),SCFLUX(MAX,NGR))
      INDEX=2
   31 M7=1
      BSC(K1,1)=0.0
      DO 30 K=1,JMAX
      L=II(K)-1
      EM=DELR(K)/3.0
      EN=EM*.5
      SUM=0.0
      DO 29 J=M7,L
      IF(IG.NE.3) GO TO 302
      AM1=XPT(J+1)**2
      AM2=XPT(J+1)*XPT(J)
      AM3=XPT(J)**2
      A1=1.256637*(AM1+3.*AM2+6.*AM3)
      A2=1.256637*(3.*AM1+4.*AM2+3.*AM3)
      A3=1.256637*(6.*AM1+3.*AM2+AM3)
      GO TO 29
  302 A1=A2=A3=1.0
   29 SUM=SUM+EM*(A1*SCPROD1(J)+.5*A2*(SCPROD2(J)+SCPROD3(J))+A3*SCPROD1
     1(J+1))
      EEAK(K,K1)=SUM
      BSC(K1,1)=BSC(K1,1)+SUM
   30 M7=II(K)
      K1=K1+1
      GO TO(12,28),INDEX
   28 CONTINUE
      NBB=1
      NAB=NGR
      NCB=NGR-6
      IF(NCB)32,32,33
   32 NAB=NGR
      GO TO 34
   33 NAB=6
   34 CONTINUE
      NBB1=NBB+1
      NBB2=NBB+2
      NBB3=NBB+3
```

```
      NBB4=NBB+4
      NBB5=NBB+5
      PRINT 502,NBB,NBB1,NBB2,NBB3,NBB4,NBB5
      DO 35 J=1,JMAX
   35 PRINT 503,J,(EEAK(J,M6),M6=NBB,NAB)
      PRINT 504,(BSC(M6,1),M6=NBB,NAB)
      DO 350 J=1,JMAX
      PRINT 520,(SCJF(J,M6),M6=NBB,NAB)
  350 PRINT 521,(SCJA(J,M6),M6=NBB,NAB)
      DO 80 J7=1,NHREG
   80 PRINT 522,(BSV(J7,M6),M6=NBB,NAB)
      DO 81 J7=1,NHREG
      L6=J7+6
   81 PRINT 522,(BSV(L6,M6),M6=NBB,NAB)
      NBB=NAB+1
      IF(NCB)37,37,36
   36 NCB=NCB-6
      IF(NCB)38,38,39
   38 NAB=NAB+NCB+6
      GO TO 34
   39 NAB=NAB+6
      GO TO 34
   37 PRINT 505
      M7=1
      IF(JMAX-6)40,40,41
   40 M6=JMAX
      GO TO 42
   41 M6=6
   42 DO 43 J=M7,M6
      SUM=0.0
      DO 44 K=1,NGR
   44 SUM=SUM+EEAK(J,K)
      CHIH1(J)=SUM
   43 CONTINUE
      M71=M7+1
      M72=M7+2
      M73=M7+3
      M74=M7+4
      M75=M7+5
      PRINT 506,M7,M71,M72,M73,M74,M75
      PRINT 507,(CHIH1(M5),M5=M7,M6)
      IF(M6-JMAX)45,46,46
   45 M7=M6+1
      IF(M6+6-JMAX)47,47,48
   47 M6=M6+6
      GO TO 42
   48 M6=JMAX
      GO TO 42
   46 SUM=0.0
      DO 49 I=1,NGR
   49 SUM=SUM+BSC(I,1)
      PRINT 508,SUM
      DO 50 I=1,NGR
      DO 50 J=1,JMAX
      M6=MIR(J)
      IF(VINV(I,M6))51,50,51
   50 CONTINUE
      PRINT 509
      GO TO 52
   51 SUME=0.0
      DO 53 I=1,NGR
      DO 53 J=1,JMAX
```

```
      M6=MIR(J)
   53 SUME=SUME+EEAK(J,I)*VINV(I,M6)
      PRINT 510,SUME
   52 IF(UNIT,4)52,57,60
   57 M7=1
      DO 56 K=1,JMAX
      M6=MIR(K)
      L=II(K)
      DO 55 J=M7,L
      M5=K+J-1
      SUM=0.0
      SOME=0.0
      DO 54 I=1,NGR
      SUM=SUM+CHI(I,M6)*SC(J,I)
   54 SOME=SOME+VUSIG(I,M6)*SCFLUX(J,I)
      UA(M5)=SUM
      UA(M5+200)=SOME
   55 CONTINUE
   56 M7=II(K)
      M7=1
      SUM=0.0
      DO 59 I=1,JMAX
      L=II(I)-1
      EM=DELR(I)/3.
      EN=EM*.5
      DO 58 J=M7,L
      IF(IG.NE.3) GO TO 65
      AM1=XPT(J+1)**2
      AM2=XPT(J+1)*XPT(J)
      AM3=XPT(J)**2
      A1=1.256637*(AM1+3.*AM2+6.*AM3)
      A2=1.256637*(3.*AM1+4.*AM2+3.*AM3)
      A3=1.256637*(6.*AM1+3.*AM2+AM3)
      GO TO 67
   65 A1=A2=A3=1.0
   67 K=I+J-1
      SUM=SUM+EM*(A1*UA(K)+UA(K+200)+.5*A2*(UA(K)*UA(K+201)+UA(K+1)*
     XUA(K+200))+A3*UA(K+1)*UA(K+201))
   58 CONTINUE
   59 M7=II(I)-1
      PRINT 511,SUM
      SOME=SUME/(SUM*EIGEN1)
      PRINT 512,SOME
      CALL HOMOG
      IF(IJOUT.EQ.1) NOT=NOT+6
      RETURN
    3 PRINT 513
      CALL Q8QERROR(0,4HBUG.)
   60 PRINT 514
      CALL Q8QERROR(0,4HBUG.)
   64 RETURN
  500 FORMAT(*1AUXILIARY OUTPUT FOR PERTURBATION ANALYSIS*)
  501 FORMAT(/*0INTEGRAL OF (FLUX(R,MU) TIMES ADJOINT(R,MU))DMU DR*)
  502 FORMAT(/10H    REGION,6X,5HGROUP I3,5(6X,5HGROUP I3))
  503 FORMAT(8X,I2,6(1X,E13.6))
  504 FORMAT(/10H    TOTAL =,6(1X,E13.6))
  505 FORMAT(*0REGION-WISE TOTAL INTEGRAL OF ANGULAR FLUX TIMES ADJOINT
     1*)
  506 FORMAT(/11H      REGION,I3,5(6X,6HREGION,I3))
  507 FORMAT(6(2X,E13.6))
  508 FORMAT(//48H TOTAL INTEGRAL OVER ANGLE, SPACE, AND ENERGY = E13.6)
  509 FORMAT(//* NO INVERSE VELOCITIES PROVIDED, LIFETIME NOT COMPLTED*)
```

```
510 FORMAT(//56H INTEGRAL OF ((1/V(E))PHI(E,R,MU)PHI*(E,R,MU))DMUDRDE
   1= E13.6)
511 FORMAT(//66H INTEGRAL OF ((CHI(I)PHI*(I,R))(NUSIGF(I,R)PHI(I,R),
   1I=1,NGR)DR = E13.6)
512 FORMAT(//34H SYSTEM PROMPT NEUTRON LIFETIME = E13.6)
513 FORMAT(*0EOF OR PARITY ERROR ON TAPE 3*)
514 FORMAT(*0EOF OR PARITY ERROR ON TAPE 4*)
515 FORMAT(//* NO ENERGY HOMOGENIZATION REQUESTED, OUTPUT FOR ONE HOMO
   1GENIZED GROUP IS PROVIDED*)
516 FORMAT(//32H NUMBER OF HOMOGENIZED GROUPS = I2,/25H FINE GROUP BOU
   1NDARIES = 10(I3))
517 FORMAT(//* ERROR, NO. OF HOMOGENIZED GROUPS REQUESTED IS TOO LARGE
   1, ONLY ONE IS COMPUTED*)
518 FORMAT(//*NO SPACE HOMOGENIZATION REQUESTED, OUTPUT FOR ONE HOMOGE
   1NIZED REGION IS PROVIDED*)
520 FORMAT(10H..CURRENT, 6(1X,E13.6))
521 FORMAT(10H,ADJ.CURNT, 6(1X,E13.6))
522 FORMAT(10H INTEGRAL ,6(1X,E13.6))
    END LINK 5
    SUBROUTINE HOMOG
    COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
   1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
   2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
   3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
   4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJK1,
   5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
    DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
   1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
   2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
   3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
   426,25),BETA(2,26),LRATE(25)
    EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
   1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
   2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
   3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
   420),KIT1),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),D6),(LL
   5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
   6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
   7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
    EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
   14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),IN),(
   2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
   3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
   410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
   5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
   612522),SVM)
    EQUIVALENCE(UA,UA2)
    EQUIVALENCE(BSC1,BSC),(BSC1(1),VUSIGH),(BSC1(157),SIGS1H),(BSC1(31
   13),SIGSH),(BSC1(469),SIGTH)
    DIMENSIONUA2(400),SIGCAP(26,25),SIGCAV(650),SIGFIS(26,25),SIGFIV
   1(650),ESCAP(26,41),EEAK(40,26),BSC(40,26),VUSIGH(26,6),SIGS1H(26,6
   2),SIGSH(26,6),SIGTH(26,6)
    COMMON/2/ESCAP1(1066),EEAK1(1040),BSC1(1040),BSV(40,26),SHAP(5),
   1DON(5),STRH(234,6),STR1H(26,6),NPSH,NDSH
    COMMON/A2/SB(22,26),ALZ(22),U(22),V(22),X(150,22),CHIH(26,10),
   1XFLUX(150,22),SCPROD1(150),SCPROD2(150),SCPROD3(150),B1(22),E2(22)
   2,SCJF(40,26),SCJA(40,26),UA(400),SCPR1(40) ,SCPRJ(40),VUSIGH (26,
   36),CHIH1(26),VOLH1(40),UA3(13),VOLH(10)
    EQUIVALENCE(ESCAP1,ESCAP,SIGFIS,SIGFIV),(EEAK1,EEAK,SIGCAP,SIGCAV)
    DIMENSION CCA(26),CGF(26),CCS(26),CCS1(26),SITRH(26,6),CCD(26),
   1VINVH(26,6)
    EQUIVALENCE(UA2(1),SITRH),(UA2(157),CCD),(UA2(183),VINVH)
```

```
      BANK(0),/2/,/A2/,XLPCH,HOMOG
      BANK(1),LINK5,/A1/
      IF(NHGP)116,116,117
  116 IF(NHREG)118,118,119
  118 PRINT 529
      PRINT 530
  529 FORMAT(* NO ENERGY OR SPACE HOMOGENIZATION REQUESTED*)
  530 FORMAT(* HOMOGENIZATION DONE OVER ALL REGIONS AND ALL GROUPS*)
      NHGP=1
      NGREG=1
      LHGP(1)=NGR
      LHREG(1)=JMAX
      GO TO 120
  119 PRINT 531,NGR
  531 FORMAT(37H NO ENERGY HOMOGENIZATION REQUESTED, I2,16H GROUPS PROVI
     1DED)
      NHGP=NGR
      DO 121 I=1,NGR
  121 LHGP(I)=I
      GO TO 129
  117 IF(NHREG)122,122,123
  122 PRINT 532,JMAX
  532 FORMAT(36H NO SPACE HOMOGENIZATION REQUESTED, I2,17H REGIONS PROVI
     1DED)
      NHREG=JMAX
      DO 124 I=1,JMAX
  124 LHREG(I)=I
      GO TO 129
  123 IF(NHGP-NGR)126,126,127
  127 PRINT 533
  533 FORMAT(*  ERROR, NUMBER OF HOM. GROUPS REQUESTED IS TOO LARGE*)
      PRINT 534,NGR
  534 FORMAT(4X,I2,28H HOMOGENIZED GROUPS PROVIDED)
      NHGP=NGR
      DO 128 I=1,NGR
  128 LHGP(I)=I
  126 IF(NHREG-JMAX)129,129,130
  130 PRINT 535
  535 FORMAT(*  ERROR, NUMBER OF HOM. REGIONS REQUESTED IS TOO LARGE*)
      PRINT 536,JMAX
  536 FORMAT(4X,I2,29H HOMOGENIZED REGIONS PROVIDED)
      NHREG=JMAX
      DO 131 I=1,JMAX
  131 LHREG(I)=I
  129 M=NHREG-1
      DO 132 I=1,M
      IF(LHREG(I+1)-LHREG(I))133,132,132
  133 PRINT 537
  537 FORMAT(* ERROR, HOMOGENIZED REGION BOUNDARIES NOT IN ASCENDING ORD
     1ER*)
      GO TO 114
  132 CONTINUE
      M=NHGP-1
      DO 134 I=1,M
      IF(LHGP(I+1)-LHGP(I))135,135,134
  135 PRINT 538
  538 FORMAT(* ERROR, HOMOGENIZED GROUP BOUNDARIES NOT IN ASCENDING ORDE
     1R*)
      GO TO 114
  134 CONTINUE
      IF(LHREG(NHREG)-JMAX)136,136,137
  137 PRINT 539
```

```
539 FORMAT(* ERROR, HOM. REGION BOUNDARIES TOO LARGE*)
    GO TO 114
136 IF(LHGP(NHGP)-NGR)120,120,138
138 PRINT 540
540 FORMAT(* ERROR, HOM. GROUP BOUNDARIES TOO LARGE*)
    GO TO 114
120 IF(LFREG.LT.1)LFREG=1
    M7=1
    DO141 I=1,NHREG
    L=II(LHREG(I))
    VOLH(I)=XPT(L)-XPT(M7)
    IF(IG.EQ.3)VOLH(I)=4.1887902*(XPT(L)**3-XPT(M7)**3)
141 M7=L
    NPSH=0
    NN6=1
    NDSH=0
    N2=1
    AM1=0.0
    DO 72 J7=1,NHREG
    MATNAM=LFREG+J7-1
    N5=1
    N3=LHREG(J7)
    VOL1=VOLH(J7)
    SUMN1=0.0
    DO143 I4=1,NHGP
    N6=LHGP(I4)
    N7=1
    SUMN2=0.0
    DO 65 I5=1,NHGP
 28 SUM2=SUM5=SUM6=SUM7=0.0
 30 N8=LHGP(I5)
    EM1=EM2=EM3=EM4=EM6=EM1A=0.0
    DO142 I=N5,N6
    DO 66 I1=N7,N8
 25 M1=NN6
    DO 667 J=N2,N3
    SOME=SUME=SOM=0.0
    M2=II(J)-1
    IF(IG.EQ.3) GO TO 31
    VOL5= XPT(M2+1)-XPT(M1)
    GO TO 26
 31 VOL5=4.1887902*(XPT(M2+1)**3-XPT(M1)**3)
 26 EM=DELR(J)
    EN=EM/3.0
    DO 68 L=M1,M2
    A1=A2=A3=1.0
    A11=A12=.5*EM
    IF(IG.NE.3) GO TO 10
    AM1=XPT(L+1)**2
    AM2=XPT(L+1)*XPT(L)
    AM3=XPT(L)**2
    A11=1.0471976*EM*(AM1+2.*AM2+3.*AM3)
    A12=1.0471976*EM*(3.*AM1+2.*AM2+AM3)
    A1=1.256637*(AM1+3.*AM2+6.*AM3)
    A2=1.256637*(3.*AM1+4.*AM2+3.*AM3)
    A3=1.256637*(6.*AM1+3.*AM2+AM3)
 10 SOME=SOME+A11*SCFLUX(L,I)+A12*SCFLUX(L+1,I)
    SOM=SOM+A11*SCFLUX(L,I1)+A12*SCFLUX(L+1,I1)
 68 SUME=SUME+EN*(A1*SC(L,I1)*SCFLUX(L,I)+.5*A2*(SC(L,I1)*SCFLUX(L+1,I
   1)+SC(L+1,I1)*SCFLUX(L,I))+A3*SC(L+1,I1)*SCFLUX(L+1,I))
    I2=MIR(J)
    EM2=EM2+CHI(I1,I2)*VUSIG(I,I2)*SUME
```

```
      IF(I.NE.N5)GO TO 21
      IF(NOT.EQ.5.OR.NOT.EQ.6) SUM5=SUM5+.5*EEAK(J,I1)
      SUM7=SUM7+SOM
   21 IF(I1.NE.N7)GO TO 20
      SUM2=SUM2+SOME
      IF(NOT.EQ.5.OR.NOT.EQ.6) SUM6=SUM6+.5*EEAK(J,I)
   20 IF(I1-I)67,70,69
   70 EM1=EM1+SIGT(I,I2)*EEAK(J,I)
      EM3=EM3+SIGS(I,I2)*SUME
      EM4=EM4+SIGS1(I,I2)*SCJF(J,I)
      EM6=EM6+VINV(I,I2)*SOME
      GO TO 67
   69 M4=I1-I-1
      IF(I1-I.GT.NDS)GO TO 67
      M5=25
      M6=0
      DO 139 J5=1,M4
      M6=M6+M5
  139 M5=M5-1
      M4=I+M6
      EM3=EM3+STR(M4,I2)*SUME
      IF(I-I1+1)67,79,67
   79 EM4=EM4+STR1(I,I2)*SCJA(J,I+1)
   67 CONTINUE
      IF(ITOUT.NE.40)GO TO 667
      PRINT 521,SOME,SOM,SUME,EEAK(J,I)
      PRINT 389,J,J7,I,I4,I1,I5,SUM2,SUM5,SUM6,SUM7
  389 FORMAT(6I5,4E18.6)
  667 M1=M2+1
      IF(I.NE.N5) GO TO 66
      IF(NOT.NE.5.AND.NOT.NE.6) SUM5=SUM5+BSV(J7,I1)
   66 CONTINUE
      IF(NOT.NE.5.AND.NOT.NE.6) SUM6=SUM6+BSV(J7,I)
      L6=J7+6
      IF(NOT.EQ.5.OR.NOT.EQ.8.OR.NOT.EQ.10) EM1A=EM1A+BSV(L6,I)
      PRINT 360,J7,I,SUM5,SUM6,EM1A,I4,I5
  360 FORMAT(4H J7=,I2,5X,2HI=,I2,5X,5HSUM5=,E18.6,5X,5HSUM6=,E18.6,5X,5
     1HEM1A=,E18.6,2X,3HI4=,I2,2X,3HI5=,I2)
  142 CONTINUE
      SUMN2=SUMN2+EM2
      EN1=N8-N7+1
      EM1=EM1+EN1
      EM5=EM5+EN1
      EM2=EM2+EN1
      EM3=EM3+EN1
      EM4=EM4+EN1
      VUSIGH1(I5,J7)=(EM2*SUM7)/(SUM5*SUM2*EN1*25.132741)
      IF(I5-I4) 65,318,73
  318 VINVH(I4,J7)=EM6/SUM2
      SIGTH(I4,J7)=EM1/(2.0*SUM6*EN1)
      IF(NOT.EQ.5.OR.NOT.EQ.8.OR.NOT.EQ.10) SIGTH(I4,J7)=SIGTH(I4,J7)-EM
     1A/(2.*SUM6)
      SIGSH(I4,J7)=(EM3*SUM7)/(SUM5*SUM2*EN1*25.132741)
  319 SIGS1H(I4,J7)=(EM4*SUM7)/(SUM5*SUM2*6.2831853*EN1)
      GO TO 65
   73 M9=25
      M10=I5-I4-1
      IF(I5-I4.GT.NDS)GO TO 65
      IF(I5-I4.GT.NDSH)NDSH=I5-I4
      M11=0
      DO 74 J5=1,M10
      M11=M11+M9
```

```
 74 M9=M9+1
    M10=I4+M11
    STRH(M10,J7)=(EM3*SUM7)/(SUM5*SUM2*EN1*25.132741)
    IF(I5-I4-1)65,80,65
 80 STR1H(I4,J7)=(EM4*SUM7)/(SUM5*SUM2*EN1*6.2831853)
    IF(STR1H(I4,J7).NE.0.)NPSH=1
 65 N7=N8+1
    SUM=0.0
    DO 75 J5=1,NHGP
 75 SUM=SUM+VUSIGH1(J5,J7)
    DO 76 J5=1,NHGP
 76 CHIH1(J5)=VUSIGH1(J5,J7)/SUM
    VUSIGH(I4,J7)=SUM
    PRINT 520,I4,J7
    PRINT 521,(CHIH1(J5),J5=1,NHGP)
520 FORMAT(/40H FISSION SPECTRUM FROM HOMOGENIZED GROUP I3,20H HOMOGEN
   1IZED REGION I2)
521 FORMAT(7(3X,E12.5))
    IF(MIK.NE.2) GO TO 304
    XNQ=NHGP/5.
    NNQ=XNQ
    FNQ=NNQ
    IF(XNQ-FNQ)308,308,307
307 NNQ=NNQ+1
308 PUNCH 306,NNQ,I4,J7
306 FORMAT (14H,THE FOLLOWING ,I2,50H, ENERGY DEPEND FISS SPEC FRAC CA
   1RDS ARE FOR GROUP ,I2,9H,, REGION ,I2)
    NCARD=1
    NCRDTP=2
    MQ=1
    NQ=5
303 IF(NHGP.GE.NQ) GO TO 301
    NQ=NHGP
301 PUNCH 302,ISET,MATNAM,NCRDTP,I4,NCARD,(CHIH1(J5),J5=MQ,NQ)
    NCARD=NCARD+1
302 FORMAT(A5,I6,1X,I1,2I3,1X,5E12.6)
    IF(NHGP-NQ)304,304,305
305 MQ=MQ+5
    NQ=NQ+5
    GO TO 303
304 CONTINUE
    IF(SUMN2.LT.SUMN1) GO TO 143
    SUMN1=SUMN2
    DO 720 J5=1,NHGP
720 CHIH(J5,J7)=CHIH1(J5)
143 N5=N6+1
    N2=N3+1
    IF(ISAVE.EQ.0)GO TO 72
    J8=J7+LFREG-1
    DO 721 J5=1,NHGP
    SIGT(J5,J8)=SIGTH(J5,J7)
    SIGS(J5,J8)=SIGSH(J5,J7)
    SIGS1(J5,J8)=SIGS1H(J5,J7)
    VUSIG(J5,J8)=VUSIGH(J5,J7)
    CHI(J5,J8)=CHIH(J5,J7)
    VINV(J5,J8)=VINVH(J5,J7)
    IF(J5.EQ.NHGP) GO TO 721
    STR1(J5,J8)=STR1H(J5,J7)
721 CONTINUE
    DO 722 J5=1,234
722 STR(J5,J8)=STRH(J5,J7)
 72 NN6=II(N3)
```

```
      REWIND 2
      REWIND 8
C     NOW PRINT AND PUNCH HOM X-SECTS
      K7=0
      PRINT 522
      PRINT 542,(LHGP(I),I=1,NHGP)
      PRINT 543,(LHREG(I),I=1,NHREG)
  542 FORMAT(1H0,4X,24HLOWER GROUP BOUNDARIES = 26I4)
  543 FORMAT(1H0,4X,25HUPPER REGION BOUNDARIES = 6I6)
      DO 100 I=1,NHREG
      PRINT 523,I
  522 FORMAT(1H1,2X,30HHOMOGENIZED CROSS SECTION DATA)
  523 FORMAT(///4X,11H***********/4X,11HHOMOGENIZED,10X,7H*SIGMA*,7X,9H*
     1SIGMA S*,5X,9H*SIGMA S*,5X,10H*NU-SIGMA*,4X,9H*FISSION*,3X,11H*DIF
     2FUSION*,5X,9H*INVERSE*/4X,7H*REGION,I3,1H*,10X,7H*TOTAL*,7X,9H* ZE
     3RO  *,5X,9H*  ONE  *,5X,9H*FISSION*,5X,10H*SPECTRUM*,2X,13H*COEFFI
     4CIENT*,3X,10H*VELOCITY*/4X,11H***********)
  524 FORMAT(6X,9HHOM GROUP I3,3X,7(1X,E13.6))
  525 FORMAT(1H0,3X,15HTRANSFER MATRIX)
  526 FORMAT(9X,9HDOWN**** I2/(5E18.6))
  527 FORMAT(7X,11HP1 DOWN*** I2/(5E18.6))
      DO 101 J=1,NHGP
      SUM=1./(3.*SIGTH(J,I))
  101 PRINT 524,J,SIGTH(J,I),SIGSH(J,I),SIGS1H(J,I),VUSIGH(J,I),CHIH(J,I
     1),SUM,VINVH(J,I)
      IF(NDSH)103,103,102
  102 PRINT 525
      M=1
      DO 104 I1=1,NDSH
      L=NHGP-I1-1
      M7=L+M
      PRINT 526,I1,(STRH(M6,I),M6=M,M7)
  104 M=26+M-I1
  103 IF(NPSH)106,106,105
  105 M7=NHGP-1
      PRINT 527,I,(STR1H(M6,I),M6=1,M7)
C     NOW PUNCH HOM X-SECTS
  528 FORMAT(I2,I10,5E12.5)
  106 IF(MIK.EQ.2) CALL XLPCH
      IF(MIK.NE.1) GO TO 100
      PUNCH 544,(PROBT(M8),M8=1,12)
  544 FORMAT(12A6)
      M=(LFREG+I-2)*26
      M1=M+50
      M2=M+700
      M3=M+1350
      M4=M+2000
      M7=M+9775
      M8=M+2650
      M5=(LFREG+I-2)*234+3300
      M6=M+9152-LFREG-I
      K1=5
      DO 108 K=1,NHGP,5
      MM1=M1+K-1
      MM2=M2+K-1
      MM3=M3+K-1
      MM4=M4+K-1
      MM6=M6+K-1
      MM7=M7+K-1
      MM8=M8+K-1
      IF((NHGP-K).LT.5)K1=NHGP-K+1
      N11=K1+K-1
```

```
      PUNCH 528,K1,MM1,(SIGTH(N10,I),N10=K,N11)
      PUNCH 528,K1,MM2,(SIGSH(N10,I),N10=K,N11)
      PUNCH 528,K1,MM3,(SIGS1H(N10,I),N10=K,N11)
      PUNCH 528,K1,MM4,(VUSIGH(N10,I),N10=K,N11)
      PUNCH 528,K1,MM7,(VINVH(N10,I),N10=K,N11)
      PUNCH 528,K1,MM8,(CHIH(N10,I),N10=K,N11)
      IF(N11.EQ.NHGP) K1=K1-1
      IF(K1.LT.1) GO TO 200
      N11=K1+K-1
108   PUNCH 528,K1,MM6,(STR1H(N10,I),N10=K,N11)
200   IF(NDSH)109,109,110
110   K2=5
      DO 111 K=1,234,5
      MM5=M5+K-1
      IF((234-K).LT.5)K2=234-K
      K3=K+K2-1
111   PUNCH 528,K2,MM5,(STRH(K4,I),K4=K,K3)
109   PUNCH 545
545   FORMAT(72H*************************************************************
     1****************)
100   CONTINUE
114   AC69=(TIMEBEG-TIMELEFT(AC69))*.001
      PRINT 541,AC69
541   FORMAT(45H0 TOTAL TIME INCLUDING INTEGRATION EDIT TOOK F9.3,8H SEC
     1ONDS)
      RETURN
      END
      SUBROUTINE XLPCH
      COMMON/A1/LL(250),E(17435),CC(6),NN(21),NR,LC,NA,NOF,LF,NAF,I,J,
     1K,L,M,J1,J2,J3,J4,J5,M1,M2,M3,M4,M5,M6,M7,SOME,SUM,AJ,S1,S2,VOL,
     2LGO,NCE,AN,IT,XPT(150),IDIM,IUDIM,ILDIM,NPI,EIGM1,EIGEN,EIGEN1,EIG
     3EN2,NEXT,K3,MM5,III,MM4,K1,K2,NCTR,POWR2(189),POWR3(189),SC(150,26
     4),XK(3),VX(3),TIMEBEG,OUTCON,SEARCON,EIGM3,SCFLUX(150,26),IJM1,
     5EM1,MATNO(20),ISET,PROBT(12),ISAVE,ELOWER(27)
      DIMENSION II(40),MIR(40),DELR(40),POWR1(189),EMU(22),ALPHA(2,26),
     1DELTA(22,26),SVM(189,26),SIGT(26,25),SIGS(26,25),SIGS1(26,25),VUSI
     2G(26,25),CHI(26,25),STR(234,25),STR1(25,25),VINV(26,25),LHGP(26),
     3LHREG(6),CONC(40),MIX(40),MTIX(20),NTMIX(20),GAMMA(10,2,26),BUCK(
     426,25),BETA(2,26),LRATE(25)
      EQUIVALENCE(LL(1),MAX),(LL(2),JMAX),(LL(3),NGR),(LL(4),N),(LL(5),
     1NDS),(LL(6),NPS),(LL(7),NOT),(LL(8),ITOUT),(LL(9),LCO),(LL(10),MIK
     2),(LL(11),LPG),(LL(12),IDP),(LL(13),MUTEST),(LL(14),JSP),(LL(15),N
     3MIX),(LL(16),MMIX),(LL(17),KREG),(LL(18),NSOS),(LL(19),NFOS),(LL(
     420),KITI),(LL(21),IBUK),(LL(22),MADJ),(LL(23),MFR),(LL(24),BG),(LL
     5(25),IEXOP),(LL(26),II),(LL(66),MIR),(LL(106),MIX),(LL(146),MTIX),
     6(LL(166),NTMIX),(LL(186),NHGP),(LL(187),NHREG),(LL(188),LHREG),
     7(LL(198),LHGP),(LL(224),LFREG),(LL(225),NRATE),(LL(226),LRATE)
      EQUIVALENCE(LL(220),NFREG),(E(1),EPS1),(E(2),EPS2),(E(3),EPS3),(E(
     14),FAC),(E(5),THETA),(E(6),SEN),(E(7),SGES),(E(8),RR),(E(9),XIN),(
     2E(10),DELR),(E(50),SIGT),(E(700),SIGS),(E(1350),SIGS1),(E(2000),VU
     3SIG),(E(2650),CHI),(E(3300),STR),(E(9150),STR1),(E(9775),VINV),(E(
     410425),ALPHA),(E(10477),BETA),(E(10529),GAMMA),(E(11049),DELTA),(E
     5(11621),EMU),(E(11643),CONC),(E(11683),POWR1),(E(11872),BUCK),(E(
     612522),SVM)
      EQUIVALENCE(UA,UA2)
      EQUIVALENCE(BSC1,BSC),(BSC1(1),VUSIGH),(BSC1(157),SIGS1H),(BSC1(31
     13),SIGSH),(BSC1(469),SIGTH)
      DIMENSIONUA2(400),SIGCAP(26,25),SIGCAV(650),SIGFIS(26,25),SIGFIV
     1(650),ESCAP(26,41),EEAK(40,26),BSC(40,26),VUSIGH(26,6),SIGS1H(26,6
     2),SIGSH(26,6),SIGTH(26,6)
      COMMON/2/ESCAP1(1066),EEAK1(1040),BSC1(1040),BSV(40,26),SHAP(5),
     1DON(5),STRH(234,6),STR1H(26,6),NPSH,NDSH
```

```
      COMMON/A2/SB(22,26),ALZ(22),U(22),V(22),X(150,22),CHIH(26,10),
     1XFLUX(150,22),SCPROD1(150),SCPROD2(150),SCPROD3(150),B1(22),B2(22)
     2,SCJF(40,26),SCJA(40,26),UA(400),SCPR1(40) ,SCPRJ(40),VUSIGH1(26,
     36),CHIH1(26),VOLH1(40),UA3(13),VOLH(10)
      EQUIVALENCE(ESCAP1,ESCAP,SIGFIS,SIGFIV),(EEAK1,EEAK,SIGCAP,SIGCAV)
      DIMENSION CCA(26),CCF(26),CCS(26),CCS1(26),SITRH(26,6),CCD(26),
     1VINVH(26,6)
      EQUIVALENCE(UA2(1),SITRH),(UA2(157),CCD),(UA2(183),VINVH)
      BANK(0),/2/,/A2/,XLPCH,HOMOG
      BANK(1),LINK5,/A1/
      NDSH=12
      IF(NHGP.LT.12)NDSH=NHGP-1
      MATNAM=LFREG+I-1
C     PUNCH TYPE-PLUS CARD
      ADEN=1.0
      PUNCH 500,ISET  ,MATNAM,ADEN,(PROBT(J),J=1,6),I
  500 FORMAT(A5,I6,1X,1H+,E12.5,5HTESS ,2X,6A6,7HHOM MAT,I3)
      NTYPE=0
      M1=0
      M=1
C     PUNCH ZERO-TH GROUP TYPE 0 CARD, E OPT BL,  D OPT=1, TR OPT=1
      PUNCH 501,ISET  ,MATNAM,NTYPE,M1,ELOWER(1)
  501 FORMAT(A5,I6,1X,I1,I3,4X,E12.5,12X,E12.5)
      DO 2 J=1,NHGP
      M7=LHGP(J)+1
      E9=1./VINVH(J,I)
    2 PUNCH 501,ISET  ,MATNAM,NTYPE,J,ELOWER(M7),E9
C     NOW PUNCH TYPE 1 CARDS
C     BUT FIRST COMPUTE ABS AND TOT SCATTERING X-SECTS
C     NOTE THAT CAP=0, SIGFIS=ABS AND NU=NUSIGFIS/ABS  IN TYPE 1 CARDS
      M5=NHGP-1
      DO 10 J=1,M5
   10 CCS1(J)=STR1H(J)+SIGS1H(J,I)
      CCS1(NHGP)=SIGS1H(NHGP,I)
      DO 3 J=1,NHGP
      M3=0
      SUM=0.0
      M2=26
      IF(J-NHGP)12,11,12
   12 DO 4 K=1,NDSH
      K1=J+M3
      SUM=SUM+STRH(K1,I)
      M2=M2-1
      IF(K+J-NHGP)4,11,11
    4 M3=M3+M2
   11 CCS(J)=SUM+SIGSH(J,I)
      CCD(J)=1./(3.*(SITRH(J,I)-CCS1(J)))
      CCA(J)=SIGTH(J,I)-CCS(J)
    3 CCF(J)=VUSIGH(J,I)/CCA(J)
C     NOW PUNCH TYPE 1 CARDS
      CAPT=0.0
      NTYPE=1
      DO 5 J=1,NHGP
    5 PUNCH 503,ISET  ,MATNAM,NTYPE,J,CCD(J),SIGTH(J,I),CAPT,CCA(J),CCF(
     1J)
  503 FORMAT(A5,I6,1X,I1,I3,4X,5E12.5)
C     NOW PUNCH TYPE 2 CARDS IF REQUIRED-THIS GIVES FISSION SPECTRUM
      NTYPE=2
      DO 6 J=1,NHGP
      IF(CHIH(J,I))6,6,7
    6 CONTINUE
      GO TO 8
```

```
    7 M=0
      M8=0
      DO 9 J=1,NHGP,5
      M8=M8+1
      N1=J+4
      IF(J+4.GT.NHGP)N1=NHGP
    9 PUNCH 506,ISET  ,MATNAM,NTYPE,M,M8,(CHIH(K,I),K=J,N1)
  506 FORMAT(A5,I6,1X,I1,I3,I3,1X,5F12.5)
C     NOW PUNCH SCATTERING X-SECTS. AS TYPE 3(ELASTIC)
C     FIRST WE MUST FIND TOTAL P-1 SCATTERING X-SECTS
    8 NTYPE=3
      M0=0
      M1=1
      DO 13 J=1,NHGP
C     DETERMINE NO. OF POSSIBLE DOWNSCATTER GROUPS =IG
      KG=NDSH
      IF(NHGP-J.LT.NDSH) KG=NHGP-J
C     NO UPSCATTER ALLOWED
      NF=0
C     ONLY 1 P-1 DOWNSCATTER GROUP
      IG1=NPSH
      IF(J.EQ.NHGP)IG1=0
      NF1=0
      PUNCH 504,ISET  ,MATNAM,NTYPE,J,M0,CCS(J),NF,KG
      IF((CCS1(J).EQ.0.0).AND.(IG1.EQ.0)) GO TO 13
      PUNCH 504,ISET  ,MATNAM,NTYPE,J,M1,CCS1(J),NF1,IG1
  504 FORMAT(A5,I6,1X,I1,I3,3X,I1,E12.5,I3,I3)
   13 CONTINUE
C     NOW PREPARE TO PUNCH THE TRANSFER MATRIX
      NTYPE=4
      NNGP=NHGP-1
      DO 14 K=1,NNGP
      CON=CCS(K)
      M3=0
      M4=0
      M2=26
      IF(K-NHGP)18,16,18
   18 DO 15 J=1,NDSH
      M4=J
      K1=K+M3
      CCA(J+1)=STRH(K1)/CON
      M2=M2-1
      IF(J+K-NHGP)15,16,16
   15 M3=M3+M2
C     THE P-0 TRANSFER VECTOR IS NOW COMPUTED FOR GROUP K
   16 CCA(1)=SIGSH(K)/CON
C     NOW PUNCH P-0 VECTOR
      M5=M4+1
      N5=1
      DO 17 J=1,M5,5
      N4=J+4
      IF(J+4.GT.M5)N4=M5
      PUNCH 505,ISET  ,MATNAM,NTYPE,K,N5,M0,(CCA(K1),K1=J,N4)
  505 FORMAT(A5,I6,1X,I1,I3,I3,I1,5E12.5)
   17 N5=N5+1
C     NOW PREPARE P-1 VECTOR
      IF(NPSH.EQ.0) GO TO 14
      N5=1
      CON=CCS1(K)
      CCF(1)=SIGS1H(K)/CON
      CCF(2)=STR1H(K)/CON
      IF(K.EQ.NHGP)CCF(2)=0.0
      PUNCH 505,ISET  ,MATNAM,NTYPE,K,N5,M1,CCF(1),CCF(2)
   14 CONTINUE
    1 CONTINUE
      STOP
      END
```

APPENDIX F

Sample Problem

Shown below are the listing of the input card deck and the output print for a very simple two-group, two-region, anisotropic-scatter, sample problem. Although the problem is not a practical one, it uses many of the features and options of TESS and hence would serve as a good test case. Part 1 calculates the real and adjoint fluxes, then spatially weights the macroscopic cross sections using the $\sqrt{1 - \mu^2}$ bilinear weighting option. The cross sections are on cards, so the problem is completely self-contained.

The change-case capability is used to follow Part 1 with a second pass which spatially homogenizes the cross sections using the $\sqrt{1 - \mu^2}$ flux-weighting option. Rather than repeat the flux iterations, the fluxes from the cell calculation in Part 1, retained in memory, are used.

1. Listing of Sample-problem Input Deck

```
  TESS TWO GROUP-TWO REGION TEST PROBLEM,NOT EQUALS 10,SPACE COLLAPSE
16          1  7  2  2  4  1  1 10 50  1  0  0  0  1  0  4  0
 5         23  1  1  0  5  7
 2         66  3  4
 4        106  3  1  4  2
 3        186  2  1  2
 2   1    198  1  2
 1          1 .0001
 2         10 2.25          0.5
 2         50 3.0185        6.963
 2         76  10.           20.
 2        700  2.9852        6.963
 2        726  9.5           14.
 2       1350  0.01          0.02
 2       1376   0.1          0.2
 2       2026 0.0           6.0
 2       2676 1.0           0.0
 1       3300 0.33333E-01
 1       9150 0.001   E-01
 2       9775 7.23    E-10 2.29    E-09
 2       9801 7.23    E-10 2.29    E-09
 4      11643 0.0           0.05          0.0           0.01
 1       3534 0.5
 1   1   9175 0.001
   NOT EQUALS 15,SPACE COLLAPSE
 1   1       7 15
 1   1       1 .0001
```

2. Sample-problem Output

```
  TESS TWO GROUP-TWO REGION TEST PROBLEM,NOT EQUALS 10,SPACE COLLAPSE   -0

PROGRAM TESS 69A

    SLAB GEOMETRY
FLUX CALCULATION OF DUAL FLUX-ADJOINT OPTION
 INPUT DATA **************
    GEOMETRY INDICATOR ****  1
    NUMBER OF POINTS ******  7
    NUMBER OF REGIONS *****  2
    NUMBER OF GROUPS ******  2
    DOWNSCATTER GROUPS ****  1
    ANGULAR APPROXIMATION *  1
    ANGULAR INTERVALS *****  4
    POWER GUESS OPTION ****  0
    CONVERGENCE OPTION ****  1
    OUTPUT OPTION ********* 10
    ELEMS. IN MIX. VECT. **  4
    INPUT PRINT OPTION ****  0
    ITERATION MAXIMUM ***** 50
    P1-DOWNSCATTER ********  1
    TAPE ELEMENTS *********  0
    SEARCH OPTION *********  0
    SEARCH ZONE ***********  0
    SEARCH POS, IN MIX*****  0
     FILL POS. IN MIX*****   0
    BUCKLING INPUT OPTION *  0
    NO. OF REACTION RATES *  0
    HOMOGENIZED GROUPS ****  2
    HOMOGENIZED REGIONS ***  1

    EPSILON *************** 1.00000-004
    INITIAL RADIUS ******** 0.00000+000
    EXTRAPOLATION FACTOR ** 0.00000+000
    NORMALIZATION FACTOR ** 1.00000+000
    SECOND GUESS ********** 0.00000+000
    EIGENVALUE DESIRED **** 1.00000+000
    SEARCH RATIO ********** 0.00000+000


REGION DATA

      REGION        REGION        MAXIMUM        DELTA       OUTER
       NO.         MATERIAL      POINT INDEX       R         RADIUS
        1             3              5        2.25000+000 9.00000+000
        2             4              7        5.00000-001 1.00000+001


 ANGULAR DATA
     MU 1*-1.00000+000
     MU 2*-5.00000+001
     MU 3* 0.00000+000
     MU 4* 0.00000+000
     MU 5* 5.00000+001
     MU 6* 1.00000+000

 MIXTURE DATA
     1    3    0.00000+000*      MIX.
     2    1    5.00000+002*
     3    4    0.00000+000*      MIX.
     4    2    1.00000+002*

 NO FIXED SOURCE INPUT


 BOUNDARY CONDITION SPECIFICATION
        PERIODIC BOUNDARY CONDITION
   ALPHA FOR ALL GROUPS = 1.00000+000 LEFT, AND  1.00000+000 RIGHT


 CROSS SECTION DATA

 ***********
 *MATERIAL  3*     *SIGMA*      *SIGMA S*    *SIGMA S*    *NU-SIGMA*   *FISSION*    *BUCKLING*
 ***********      *TOTAL*      * ZERO  *    *  ONE  *    *FISSION*    *SPECTRUM*
    GROUP  1     1.50925-001  1.49960-001  5.00000-004  0.00000+000  0.00000+000  0.00000+000
    GROUP  2     3.48150-001  3.48150-001  1.00000-003  0.00000+000  0.00000+000  0.00000+000
```

```
TRANSFER MATRIX
    DOWN****  1
  1.666650-003
    P-1 DOWNSCATTER
  5.000000-006


 **********
 *MATERIAL  4*      *SIGMA*       *SIGMA S*     *SIGMA S*     *NU-SIGMA*    *FISSION*     *BUCKLING*
 **********         *TOTAL*       * ZERO  *     *  ONE  *     *FISSION*     *SPECTRUM*
    GROUP  1      1.00000-001  9.50000-002  1.00000-003  0.00000+000  1.00000+000  0.00000+000
    GROUP  2      2.00000-001  1.40000-001  2.00000-003  6.00000+002  0.00000+000  0.00000+000

 TRANSFER MATRIX
    DOWN****  1
  5.000000-003
    P-1 DOWNSCATTER
  1.000000-005

 END OF INPUT PRINT
    TIME    0.725
NO. OF POINTS PER READ =      7   NO. OF READS =     1
 NO. OF LOWER AND UPPER WORDS FIRST READ =   321 AND   354
 NO. OF LOWER AND UPPER WORDS LAST READ =   321 AND   399
TIME1=    1.347 TIME2=    1.718 GROUP  1
TIME1=    1.974 TIME2=    2.586 GROUP  2
ITERATION BEGUN AT TIME =     5.107

ITER,  EIGENVALUE      EMAX      POINT     EMIN     POINT     CAL.-EPS.     INP.-EPS.
  1    1.00074+000  1.00060+000   8    9.93399-001   7    7.39170+004  1.00000-004
TIME1=    5.455 TIME2=    5.464 TIME3=    6.271 TIME4=    6.271 TIME5=    6.271 TIME6=    6.291

ITER,  EIGENVALUE      EMAX      POINT     EMIN     POINT     CAL.-EPS.     INP.-EPS.
  2    1.00074+000  1.00000+000   6    1.00000+000   8    2.90823+011  1.00000-004
TIME1=    6.512 TIME2=    6.520 TIME3=    7.377 TIME4=    7.377 TIME5=    7.377 TIME6=    7.399

ITER,  EIGENVALUE      EMAX      POINT     EMIN     POINT     CAL.-EPS.     INP.-EPS.
  3    1.00074+000  1.00000+000   6    1.00000+000   8    0.00000+000  1.00000-004
TIME1=    7.790 TIME2=    7.798 TIME3=    8.491 TIME4=    8.496 TIME5=    8.496 TIME6=    8.516


  ****OUTPUT DATA****
 FINAL ITERATION=  3          FINAL MULTIPLICATION FACTOR= 1.00074+000


  POINT     RADIUS    MATERIAL   FISSION      GROUP 1      GROUP 2      GROUP 3      GROUP 4      GROUP 5      GROUP 6
  NUMBER                         DENSITY       FLUX         FLUX         FLUX         FLUX         FLUX         FLUX
     1    0.0000+000     3      0.0000+000  5.0645+001  1.6789+001
     2    2.2500+000     3      0.0000+000  4.9593+001  1.8404+001
     3    4.5000+000     3      0.0000+000  4.9420+001  1.8290+001
     4    6.7500+000     3      0.0000+000  4.9593+001  1.8404+001
     5    9.0000+000     3      0.0000+000  5.0645+001  1.6789+001
 ***************************************************************************************************************
     5    9.0000+000     4      1.0066+000  5.0645+001  1.6789+001
     6    9.5000+000     4      9.9340+001  5.0775+001  1.6569+001
     7    1.0000+001     4      1.0066+000  5.0645+001  1.6789+001
 ***************************************************************************************************************


 AUXILIARY OUTPUT FOR GROUP  1

                                          HEMISPHERE-   HEMISPHERE-   ANGULAR FLUXES
 PT, RADIUS            SCALAR FLUX        CURRENT-LEFT  CURRENT-RIGHT FROM MU=1. TO -1.
   1  0.000000+000  5.064544+001  -1.237569+001  1.274891+001  4.089008+000  4.045565+000  4.378840+000  3.959286+000
                                                                3.943439+000  3.943747+000
   2  2.250000+000  4.959343+001  -1.234468+001  1.253015+001  3.950311+000  4.000805+000  3.905262+000  3.879437+000
                                                                3.985981+000  3.943579+000
   3  4.500000+000  4.942022+001  -1.240786+001  1.240786+001  3.963003+000  3.952898+000  3.862147+000  3.862147+000
                                                                3.952898+000  3.963003+000
   4  6.750000+000  4.959343+001  -1.253015+001  1.234468+001  3.943579+000  3.925981+000  3.879437+000  3.905262+000
                                                                4.000805+000  3.990311+000
   5  9.000000+000  5.064544+001  -1.274891+001  1.237569+001  3.943747+000  3.933439+000  3.952286+000  4.378840+000
                                                                4.045565+000  4.009008+000
   6  9.500000+000  5.077464+001  -1.257629+001  1.257629+001  3.977171+000  3.991786+000  4.201329+000  4.201329+000
                                                                3.991786+000  3.977171+000
   7  1.000000+001  5.064544+001  -1.237569+001  1.274891+001  4.089008+000  4.045565+000  4.378840+000  3.959286+000
                                                                3.943439+000  3.943747+000
```

AUXILIARY OUTPUT FOR GROUP 2

PT.	RADIUS	SCALAR FLUX	HEMISPHERE-CURRENT-LEFT	HEMISPHERE-CURRENT-RIGHT	ANGULAR FLUXES FROM MU=1. TO -1.			
1	0.000000+000	1.678909+001	-4.489111+000	4.115516+000	1.384397+000	1.310882+000	1.032836+000	1.368846+000
					1.420956+000	1.438517+000		
2	2.250000+000	1.840374+001	-4.628044+000	4.442396+000	1.395233+000	1.395967+000	1.616732+000	1.499433+000
					1.478471+000	1.462914+000		
3	4.500000+000	1.829048+001	-4.568406+000	4.568406+000	1.442278+000	1.464718+000	1.450324+000	1.450324+000
					1.484718+000	1.442278+000		
4	6.750000+000	1.840374+001	-4.442396+000	4.628044+000	1.462914+000	1.478471+000	1.492433+000	1.616732+000
					1.395967+000	1.395233+000		
5	9.000000+000	1.678909+001	-4.115516+000	4.489111+000	1.438512+000	1.430956+000	1.368846+000	1.032836+000
					1.310882+000	1.384397+000		
6	9.500000+000	1.656890+001	-4.276917+000	4.276917+000	1.399646+000	1.365997+000	1.142405+000	1.142405+000
					1.365997+000	1.399646+000		
7	1.000000+001	1.678909+001	-4.489111+000	4.115516+000	1.384397+000	1.310882+000	1.032836+000	1.368846+000
					1.420956+000	1.438517+000		

NEUTRON BALANCE CHARACTERISTICS GROUP 1

	INTEGRATED FLUX	INTEGRATED FIXED SOURCE	TOTAL ABSORPTIONS	TOTAL FISSIONS/K	NET LEAKAGE
REGION 1	4.483182+002	0.000000+000	-7.397251-004	0.000000+000	-7.464498-001
REGION 2	5.071004+001	0.000000+000	-1.383615-010	0.000000+000	7.464498-001
TOTAL***	4.990282+002	0.000000+000	-7.397253-004	0.000000+000	-9.313226-010

NEUTRON BALANCE CHARACTERISTICS GROUP 2

	INTEGRATED FLUX	INTEGRATED FIXED SOURCE	TOTAL ABSORPTIONS	TOTAL FISSIONS/K	NET LEAKAGE
REGION 1	1.617459+002	0.000000+000	0.000000+000	0.000000+000	7.471895-001
REGION 2	1.667900+001	0.000000+000	1.000740+000	1.000000+000	-7.471895-001
TOTAL***	1.784249+002	0.000000+000	1.000740+000	1.000000+000	-3.492460-010

REACTOR NEUTRON BALANCE

TOTAL***	6.774531+002	0.000000+000	1.000000+000	1.000000+000	-1.280569-009

THIS CASE CONSUMED 11.907SECONDS OF COMPUTER TIME

ADJOINT CALCULATION, CORRESPONDING TO PRECEEDING FLUX CALC

TESS TWO GROUP-TWO REGION TEST PROBLEM,NOT EQUALS 10,SPACE COLLAPSE

```
NO. OF POINTS PER READ =      7   NO. OF READS =     1
 NO. OF LOWER AND UPPER WORDS FIRST READ =   321 AND   354
 NO. OF LOWER AND UPPER WORDS LAST READ =   321 AND   399
 IN ON 2FROM     1TO   321
OUT ON 3FROM     1TO   321
 IN ON 2FROM  3001TO  3354
OUT ON 3FROM  3001TO  3354
 IN ON 8FROM     1TO   321
OUT ON 2FROM     1TO   321
 IN ON 8FROM  3001TO  3354
OUT ON 2FROM  3001TO  3354
 IN ON 3FROM     1TO   321
OUT ON 8FROM     1TO   321
 IN ON 3FROM  3001TO  3354
OUT ON 8FROM  3001TO  3354
ITERATION BEGUN AT TIME =    26.113
ITER,  EIGENVALUE      EMAX     POINT     EMIN    POINT    CAL.-EPS.    INP.-EPS.
  1    1.00074+000  1.00000+000   6    1.00000+000   7    7.39170E+004  1.00000-004
TIME1=   26.366 TIME2=   26.374 TIME3=   27.278 TIME4=   27.278 TIME5=   27.279 TIME6=   27.301
```

```
ITER,  EIGENVALUE      EMAX      POINT      EMIN     POINT    CAL.-EPS,    INP.-EPS.
  2    1.00074+000  1.00000+000   6    1.00000+000   7    0.00000+000  1.00000-004
TIME1=   27.474 TIME2=   27.482 TIME3=   28.389 TIME4=   28.389 TIME5=   28.389 TIME6=   28.409

ITER,  EIGENVALUE      EMAX      POINT      EMIN     POINT    CAL.-EPS,    INP.-EPS.
  3    1.00074+000  1.00000+000   6    1.00000+000   7    0.00000+000  1.00000-004
TIME1=   28.633 TIME2=   28.642 TIME3=   29.498 TIME4=   29.502 TIME5=   29.502 TIME6=   29.524
```

****OUTPUT DATA****

FINAL ITERATION= 3 FINAL MULTIPLICATION FACTOR= 1.000740+000

POINT NUMBER	RADIUS	MATERIAL	FISSION DENSITY	GROUP 1 ADJOINT	GROUP 2 ADJOINT	GROUP 3 ADJOINT	GROUP 4 ADJOINT	GROUP 5 ADJOINT	GROUP 6 ADJOINT
1	0.0000+000	3	0.0000+000	1.0007+000	1.0000+000				
2	2.2500+000	3	0.0000+000	1.0007+000	1.0000+000				
3	4.5000+000	3	0.0000+000	1.0007+000	1.0000+000				
4	6.7500+000	3	0.0000+000	1.0007+000	1.0000+000				
5	9.0000+000	3	0.0000+000	1.0007+000	1.0000+000				
5	9.0000+000	4	1.0000+000	1.0007+000	1.0000+000				
6	9.5000+000	4	1.0000+000	1.0007+000	1.0000+000				
7	1.0000+001	4	1.0000+000	1.0007+000	1.0000+000				

AUXILIARY OUTPUT FOR GROUP 2

PT,	RADIUS	SCALAR ADJOINT	HEMISPHERE-CURRENT-RIGHT	HEMISPHERE-CURRENT-LEFT	ANGULAR ADJOINTS FROM MU=-1. TO 1.			
1	0.000000+000	1.000000+000	-2.500000-001	2.500000-001	7.957747-002	7.957747-002	7.957747-002	7.957747-002
					7.957747-002	7.957747-002		
2	2.250000+000	1.000000+000	-2.500000-001	2.500000-001	7.957747-002	7.957747-002	7.957747-002	7.957747-002
					7.957747-002	7.957747-002		
3	4.500000+000	1.000000+000	-2.500000-001	2.500000-001	7.957747-002	7.957747-002	7.957747-002	7.957747-002
					7.957747-002	7.957747-002		
4	6.750000+000	1.000000+000	-2.500000-001	2.500000-001	7.957747-002	7.957747-002	7.957747-002	7.957747-002
					7.957747-002	7.957747-002		
5	9.000000+000	1.000000+000	-2.500000-001	2.500000-001	7.957747-002	7.957747-002	7.957747-002	7.957747-002
					7.957747-002	7.957747-002		
6	9.500000+000	1.000000+000	-2.500000-001	2.500000-001	7.957747-002	7.957747-002	7.957747-002	7.957747-002
					7.957747-002	7.957747-002		
7	1.000000+001	1.000000+000	-2.500000-001	2.500000-001	7.957747-002	7.957747-002	7.957747-002	7.957747-002
					7.957747-002	7.957747-002		

AUXILIARY OUTPUT FOR GROUP 1

PT,	RADIUS	SCALAR ADJOINT	HEMISPHERE-CURRENT-RIGHT	HEMISPHERE-CURRENT-LEFT	ANGULAR ADJOINTS FROM MU=-1. TO 1.			
1	0.000000+000	1.000740+000	-2.501864-001	2.501846-001	7.963647-002	7.963629-002	7.963463-002	7.963675-002
					7.963684-002	7.963679-002		
2	2.250000+000	1.000745+000	-2.501866-001	2.501857-001	7.963656-002	7.963651-002	7.963698-002	7.963711-002
					7.963688-002	7.963679-002		
3	4.500000+000	1.000746+000	-2.501863-001	2.501863-001	7.963669-002	7.963674-002	7.963719-002	7.963719-002
					7.963674-002	7.963669-002		
4	6.750000+000	1.000745+000	-2.501857-001	2.501866-001	7.963679-002	7.963688-002	7.963711-002	7.963698-002
					7.963651-002	7.963656-002		
5	9.000000+000	1.000740+000	-2.501846-001	2.501864-001	7.963679-002	7.963684-002	7.963675-002	7.963463-002
					7.963629-002	7.963647-002		
6	9.500000+000	1.000739+000	-2.501854-001	2.501854-001	7.963662-002	7.963655-002	7.963551-002	7.963551-002
					7.963655-002	7.963662-002		
7	1.000000+001	1.000740+000	-2.501864-001	2.501846-001	7.963647-002	7.963629-002	7.963463-002	7.963675-002
					7.963684-002	7.963679-002		

NEUTRON BALANCE CHARACTERISTICS GROUP 1

	INTEGRATED FLUX	INTEGRATED FIXED SOURCE	TOTAL ABSORPTIONS	TOTAL FISSIONS/K	NET LEAKAGE
REGION 1	9.006697+000	0.000000+000	-1.486105-005	0.000000+000	6.184564-011
REGION 2	1.000740+000	0.000000+000	-2.730502-012	0.000000+000	-7.275958-011
TOTAL***	1.000744+001	0.000000+000	-1.486106-005	0.000000+000	-2.910383-011

NEUTRON BALANCE CHARACTERISTICS GROUP 2

	INTEGRATED FLUX	INTEGRATED FIXED SOURCE	TOTAL ABSORPTIONS	TOTAL FISSIONS/K	NET LEAKAGE
REGION 1	9.000000+000	0.000000+000	0.000000+000	0.000000+000	3.698595-006
REGION 2	1.000000+000	0.000000+000	6.000000-002	5.995565-002	-3.698624-006
TOTAL***	1.000000+001	0.000000+000	6.000000-002	5.995565-002	-1.091394-011

REACTOR NEUTRON BALANCE

TOTAL***	2.000744+001	0.000000+000	5.998514-002	5.995565-002	-4.001777-011

THIS CASE CONSUMED 32.835SECONDS OF COMPUTER TIME

AUXILIARY OUTPUT FOR PERTURBATION ANALYSIS

INTEGRAL OF (FLUX(R,MU) TIMES ADJOINT(R,MU))DMU DR

REGION	GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP 5	GROUP 6
1	3.570258+001	1.287133+001				
2	4.038362+000	1.327272+000				
TOTAL =	3.974094+001	1.419860+001				
CURRENT	1.122347+002	4.036761+001				
ADJ.CURNT	0.000000+000	1.121512+002				
CURRENT	1.257862+001	4.289615+000				
ADJ.CURNT	0.000000+000	1.256930+001				
INTEGRAL	1.987268+001	7.096112+000				
INTEGRAL	3.680017-007	1.776715-013				

REGION-WISE TOTAL INTEGRAL OF ANGULAR FLUX TIMES ADJOINT

REGION 1	REGION 2	REGION 3	REGION 4	REGION 5	REGION 6
4.857390+001	5.365634+000				

TOTAL INTEGRAL OVER ANGLE, SPACE, AND ENERGY = 5.393954+001

INTEGRAL OF ((1/V(E))PHI(E,R,MU)PHI*(E,R,MU))DMUDRDE = 6.124749+008

INTEGRAL OF ((CHI(I)PHI*(I,R))(NUSIGF(I,R)PHI(I,R)), I=1,NGR)DR = 1.169495+000

SYSTEM PROMPT NEUTRON LIFETIME = 5.233217-008

J7= 1	I= 1	SUM5=	1.987268+001	SUM6=	1.987268+001	EM1A=	3.680017-007 I4= 1 I5= 1
J7= 1	I= 1	SUM5=	7.096112+000	SUM6=	1.987268+001	EM1A=	3.680017-007 I4= 1 I5= 2

FISSION SPECTRUM FROM HOMOGENIZED GROUP 1 HOMOGENIZED REGION 1

0.00000+000 0.00000+000

J7= 1	I= 2	SUM5=	1.987268+001	SUM6=	7.096112+000	EM1A=	1.776715-013 I4= 2 I5= 1
J7= 1	I= 2	SUM5=	7.096112+000	SUM6=	7.096112+000	EM1A=	1.776715-013 I4= 2 I5= 2

FISSION SPECTRUM FROM HOMOGENIZED GROUP 2 HOMOGENIZED REGION 1

1.00000+000 0.00000+000

```
HOMOGENIZED CROSS SECTION DATA

   LOWER GROUP BOUNDARIES =   1   2
0  UPPER REGION BOUNDARIES =      2

 **********
 HOMOGENIZED          *SIGMA*       *SIGMA S*     *SIGMA S*     *NU-SIGMA*    *FISSION*     *DIFFUSION*    *INVERSE*
 *REGION  1*          *TOTAL*       * ZERO  *     *  ONE  *     *FISSION*     *SPECTRUM*    *COEFFICIENT*  *VELOCITY*
 **********
   HOM GROUP  1     1.457339-001  1.437303*001  5.501675-004  0.000000*000  1.000000*000  2.287274*000  7.230000-010
   HOM GROUP  2     3.344513-001  3.288400*001  1.097803-003  5.608099-003  0.000000*000  9.966574-001  2.990000-009

 TRANSFER MATRIX
     DOWN****  1
   2.006278-003
    P1 DOWN***  1
   5.504759-006

TOTAL TIME INCLUDING INTEGRATION EDIT TOOK    34.688 SECONDS



   NOT EQUALS 15,SPACE COLLAPSE                                          -0

PROGRAM TESS 69A

    SLAB GEOMETRY
FLUX CALCULATION OF DUAL FLUX-ADJOINT OPTION
* CROSS SECTION HOMOGENIZATION USING FLUXES AND ADJOINTS FROM PRECEEDING PROBLEM *
 INPUT DATA *************
    GEOMETRY INDICATOR ****  1
    NUMBER OF POINTS ******  7
    NUMBER OF REGIONS *****  2
    NUMBER OF GROUPS ******  2
    DOWNSCATTER GROUPS ****  1
    ANGULAR APPROXIMATION *  1
    ANGULAR INTERVALS *****  4
    POWER GUESS OPTION ****  0
    CONVERGENCE OPTION ****  1
    OUTPUT OPTION ********* 15
    ELEMS. IN MIX. VECT. **  4
    INPUT PRINT OPTION ****  0
    ITERATION MAXIMUM ***** 50
    P1-DOWNSCATTER ********  1
    TAPE ELEMENTS *********  0
    SEARCH OPTION *********  0
    SEARCH ZONE ***********  0
    SEARCH POS. IN MIX*****  0
     FILL POS. IN MIX*****   0
    BUCKLING INPUT OPTION *  0
    NO. OF REACTION RATES *  0
    HOMOGENIZED GROUPS ****  2
    HOMOGENIZED REGIONS ***  1

    EPSILON ************** 1.00000-004
    INITIAL RADIUS ******** 0.00000*000
    EXTRAPOLATION FACTOR ** 0.00000*000
    NORMALIZATION FACTOR ** 1.00000*000
    SECOND GUESS ********** 0.00000*000
    EIGENVALUE DESIRED **** 1.00000*000
    SEARCH RATIO ********** 0.00000*000


 REGION DATA

      REGION        REGION        MAXIMUM        DELTA        OUTER
       NO.         MATERIAL      POINT INDEX       R          RADIUS
       1              3              5        2.25000*000  9.00000*000
       2              4              7        5.00000-001  1.00000*001


 ANGULAR DATA
     MU 1**1.00000*000
     MU 2**5.00000-001
     MU 3* 0.00000*000
     MU 4* 0.00000*000
     MU 5* 5.00000-001
     MU 6* 1.00000*000

 MIXTURE DATA
```

```
  1   3   0.00000+000*      MIX.
  2   1   5.00000-002*
  3   4   0.00000+000*      MIX.
  4   2   1.00000-002*

NO FIXED SOURCE INPUT


BOUNDARY CONDITION SPECIFICATION
        PERIODIC BOUNDARY CONDITION
  ALPHA FOR ALL GROUPS = 1.00000+000 LEFT, AND  1.00000+000 RIGHT


CROSS SECTION DATA

 ***********
*MATERIAL  3*     *SIGMA*      *SIGMA S*    *SIGMA S*    *NU-SIGMA*   *FISSION*    *BUCKLING*
 ***********      *TOTAL*      * ZERO  *    * ONE  *     *FISSION*    *SPECTRUM*
    GROUP  1     1.50925-001  1.49260+001  5.00000-004  0.00000+000  0.00000+000  0.00000+000
    GROUP  2     3.48150-001  3.48150+001  1.00000-003  0.00000+000  0.00000+000  0.00000+000

 TRANSFER MATRIX
      DOWN****  1
   1.666650-003
     P=1 DOWNSCATTER
   5.000000-006


 ***********
*MATERIAL  4*     *SIGMA*      *SIGMA S*    *SIGMA S*    *NU-SIGMA*   *FISSION*    *BUCKLING*
 ***********      *TOTAL*      * ZERO  *    * ONE  *     *FISSION*    *SPECTRUM*
    GROUP  1     1.00000-001  9.50000+002  1.00000-003  0.00000+000  1.00000+000  0.00000+000
    GROUP  2     2.00000-001  1.40000+001  2.00000-003  6.00000+002  0.00000+000  0.00000+000

 TRANSFER MATRIX
      DOWN****  1
   5.000000-003
     P=1 DOWNSCATTER
   1.000000-005

 END OF INPUT PRINT
    TIME    0.575


AUXILIARY OUTPUT FOR PERTURBATION ANALYSIS

INTEGRAL OF (FLUX(R,MU) TIMES ADJOINT(R,MU))DMU DR

   REGION        GROUP  1        GROUP  2        GROUP  3        GROUP  4        GROUP  5        GROUP  6
      1   3.567603+001   1.287133+001
      2   4.035377+000   1.327272+000

  TOTAL =  3.971141+001   1.419860+001
.,.CURRENT  1.121512+002   4.036761+001
.ADJ.CURNT  4.036761+001   1.121512+002
.,.CURRENT  1.256930+001   4.289615+000
.ADJ.CURNT  4.289615+000   1.256930+001
 INTEGRAL   1.985791+001   7.096112+000
 INTEGRAL   0.000000+000   0.000000+000

REGION-WISE TOTAL INTEGRAL OF ANGULAR FLUX TIMES ADJOINT

   REGION  1       REGION  2       REGION  3       REGION  4       REGION  5       REGION  6
 4.854736+001    5.362649+000


TOTAL INTEGRAL OVER ANGLE, SPACE, AND ENERGY =  5.391000+001

INTEGRAL OF ((1/V(E))PHI(E,R,MU)PHI*(E,R,MU))DMUDRDE =  6.122614+008

INTEGRAL OF ((CHI(I)PHI*(I,R))(NUSIGF(I,R)PHI(I,R)),  I=1,NGR)DR = 1.168631+000
```

```
SYSTEM PROMPT NEUTRON LIFETIME =  5.235263-008
J7= 1     I= 1     SUM5=      1.985791+001     SUM6=      1.985791+001     EM1A=      0.000000+000  I4= 1  I5= 1
J7= 1     I= 1     SUM5=      7.096112+000     SUM6=      1.985791+001     EM1A=      0.000000+000  I4= 1  I5= 2

FISSION SPECTRUM FROM HOMOGENIZED GROUP  1 HOMOGENIZED REGION  1
  0.00000+000    0.00000+000
J7= 1     I= 2     SUM5=      1.985791+001     SUM6=      7.096112+000     EM1A=      0.000000+000  I4= 2  I5= 1
J7= 1     I= 2     SUM5=      7.096112+000     SUM6=      7.096112+000     EM1A=      0.000000+000  I4= 2  I5= 2

FISSION SPECTRUM FROM HOMOGENIZED GROUP  2 HOMOGENIZED REGION  1
   1.00000+000    0.00000+000



  HOMOGENIZED CROSS SECTION DATA

    LOWER GROUP BOUNDARIES =   1   2
0   UPPER REGION BOUNDARIES =     2


   ***********
   HOMOGENIZED          *SIGMA*      *SIGMA S*     *SIGMA S*     *NU*SIGMA*    *FISSION*    *DIFFUSION*     *INVERSE*
   *REGION  1*          *TOTAL*      * ZERO  *     * ONE  *      *FISSION*     *SPECTRUM*   *COEFFICIENT*   *VELOCITY*
   ***********
     HOM GROUP  1     1.457339-001  1.437302-001  5.501675-004  0.000000+000  1.000000+000  2.287274+000  7.930000-010
     HOM GROUP  2     3.344513-001  3.288400-001  1.097803-003  3.608121-003  0.000000+000  9.966974-001  2.990000-009

   TRANSFER MATRIX
        DOWN****  1
     2.006278-003
      P1 DOWN***  1
     5.504759-006

  TOTAL TIME INCLUDING INTEGRATION EDIT TOOK      4.374 SECONDS
```

ACKNOWLEDGMENTS

The basic development work for TESS was done by G. E. Putnam (presently at Aerojet Nuclear Corporation) in his MIST code. He also provided many suggestions that were used in extending MIST to the present TESS code. We wish to thank Mr. Putnam for many helpful consultations. Mr. R. K. Disney of Westinghouse Astronuclear wrote the search routine, and his work in this respect has been incorporated intact. We also acknowledge helpful suggestions of R. J. Wagner of Aerojet Nuclear Corporation on efficiently handling the large coefficient arrays.

It is with great pleasure that we acknowledge the diligent and meticulous efforts of Lois Meyer (AP) who provides the link between the Idaho branch and the CDC-3600 at Argonne, Illinois. Without her efforts in assisting in debugging, making a multitude of card changes, and keeping abreast of the various monitor system changes, this program could not have been completed.

Thanks go to Arne P. Olson for suggesting the source-modification leakage treatment, as well as to R. B. Nicholson for much discussion pertaining to the cross-section homogenization formulas he developed. R. G. Palmer contributed several suggestions for improving the code from the point of view of the user.

REFERENCES

1. H. Lawroski, *Zero Power Plutonium Reactor Facility*, Nucl. News *11*, 47-49 (1968).
2. G. E. Putnam and D. M. Shapiro, *MIST (Multigroup Internuclear Slab Transport)*, IDO-16856 (May 10, 1963).
3. A. M. Weinberg and E. P. Wigner, *The Physical Theory of Neutron Chain Reactors*, University of Chicago Press, pp. 272-275 (1958).
4. Ibid, pp. 238-241.
5. R. B. Nicholson, "Cross Section Averaging Schemes for Group Collapsing and Cell Homogenization in Neutron Transport Calculations for Critical Assemblies," *Reactor Physics Division Annual Report: July 1, 1968 to June 30, 1969*, ANL-7610, pp. 488-498 (Jan 1970).
6. G. E. Putnam, private communication (July 1967).
7. L. A. Hageman and C. J. Pfeifer, *A Utilization of the Neutron Diffusion Program PDQ-5*, WAPD-TM-395 (Jan 1965).
8. D. Meneghetti, *Discrete Ordinate Quadratures for Thin Slab Cells*, Nucl. Sci. Eng. *11*, 295-303 (1962).
9. D. Meneghetti, *Table of TRIM Discrete Quadrature Parameters for Thin Binary Slab Cells*, ANL-6613 (Sept 1962).
10. G. J. Duffy, H. Greenspan, S. D. Sparck, J. V. Zapatka, and M. K. Butler, *SNARG-1D, A One-dimensional, Discrete-ordinate, Transport-theory Program for the CDC-3600*, ANL-7221 (June 1966).
11. B. J. Toppel, ed., *The Argonne Reactor Computation (ARC) System*, ANL-7332 (Nov 1967).
12. S. D. Sparck, *XLIBIT: ANL Cross-section Library Code*, ANL-7112 (Feb 1966).

ARGONNE NATIONAL LAB WEST
3 4444 00011304 3