WHC-EP-0435

1DB, A One-Dimensional Diffusion Code for Nuclear Reactor Analysis

Prepared for the U.S. Department of Energy
Assistant Secretary for Nuclear Energy

Westinghouse Hanford Company Richland, Washington

Hanford Operations and Engineering Contractor for the
U.S. Department of Energy under Contract DE-AC06-87RL10930

Approved for Public Release

WHC-EP--0435

DE92 003020

1DB, A One-Dimensional Diffusion Code for Nuclear Reactor Analysis

W. W. Little
Pacific Northwest Laboratory

Date Published
September 1991

Prepared for the U.S. Department of Energy
Assistant Secretary for Nuclear Energy

Westinghouse Hanford Company
P.O. Box 1970
Richland, Washington 99352

Hanford Operations and Engineering Contractor for the
U.S. Department of Energy under Contract DE-AC06-87RL10930

Document Title: 1DB, A ONE-DIMENSIONAL DIFFUSION CODE FOR NUCLEAR REACTOR ANALYSIS

Prepared by: Winston L. Little — 10/1/91
Winston W. Little, Jr.
Author
Date

Approved by: Leland L Carter — 10/1/91
Leland L. Carter
Approving Manager
Date

This page intentionally left blank.

1DB, A ONE-DIMENSIONAL DIFFUSION CODE FOR NUCLEAR REACTOR ANALYSIS

W. W. Little

ABSTRACT

1DB is a multipurpose, one-dimensional (plane, cylinder, sphere) diffusion theory code for use in reactor analysis. The code is designed to do the following:

- *To compute k_{eff} and perform criticality searches on time absorption, reactor composition, reactor dimensions, and buckling by means of either a flux or an adjoint model.*

- *To compute collapsed microscopic and macroscopic cross sections averaged over the spectrum in any specified zone.*

- *To compute resonance-shielded cross sections using data in the shielding factor format.*

- *To compute isotopic burnup using decay chains specified by the user.*

All programming is in FORTRAN. Because variable dimensioning is employed, no simple restrictions on problem complexity can be stated. The number of spatial mesh points, energy groups, upscattering terms, etc. is limited only by the available memory. The source file contains about 3,000 cards.

This page intentionally left blank.

CONTENTS

This page intentionally left blank.

1DB, A ONE-DIMENSIONAL DIFFUSION CODE FOR NUCLEAR REACTOR ANALYSIS

1.0 INTRODUCTION

1DB is a one-dimensional diffusion theory code designed to compute resonance-shielded group cross sections and perform burnup calculations. Resonance-shielded cross sections are calculated using data (infinite dilution cross sections and resonance-shielding factors) in the shielding factor format (Abagyan et al. 1964). Interpolation schemes are used to compute shielding factors applicable to specific compositions.

The flux profiles and eigenvalue are computed by standard source-iteration techniques. Convergence is accelerated using fission source over-relaxation. Adjoint calculations are performed by transposing the scattering matrix and fission source and inverting the group order of the input data.

Criticality searches can be performed on time absorption, material concentrations, region dimensions, and buckling. Alpha and k_{eff} can be used as parametric eigenvalues.

Isotopic burnup is computed using decay chains specified by the user. Decay, absorption, capture transfer, (n,2n) transfer, and fission product buildup are permitted. All burnup is by zone (i.e., using zone-averaged fluxes and cross sections).

The format of the input data (e.g., cross sections, flux dumps, geometry, and composition specifications) is compatible with the Los Alamos two-dimensional transport code Twodant (Alcouffe et al. 1984) and the Pacific Northwest Laboratory two-dimensional diffusion code 2DB (Little and Hardie 1968).

All input is through generalized input routines. These routines allow position-independent data to facilitate input from video terminals. Variable dimensioning is employed to make maximum use of the available memory. The 1DB code is an extensive revision to the Pacific Northwest Laboratory 1DX (Hardie and Little 1967) code. Revisions include additional flexibility in input format, enhanced burnup capabilities, activity options, inclusion of upscattering, and isotopic dependence of MeV/fission.

This page intentionally left blank.

2.0 CALCULATION OF FLUX AND EFFECTIVE MULTIPLICATION CONSTANT

2.1 FORMULATION OF DIFFERENCE EQUATIONS

The multigroup diffusion equations can be written in the form

$$D_i \nabla^2 \Phi_i - \Sigma_r^i \Phi_i + S_i = 0, \quad i = 1, 2, \ldots, IGM \tag{2.1}$$

where:

$$S_i = \frac{\chi_i}{k_{eff}} \sum_{j=1}^{IGM} (\nu\Sigma_f)_j \Phi_j + \sum_{\substack{j=1 \\ j \neq i}}^{IGM} \Sigma(j \rightarrow i) \Phi_j \tag{2.2}$$

IGM = Number of energy groups

i = Energy group index

Φ_i = Flux in group i

S_i = Source in group i

D_i = Diffusion constant for group i

$(\nu\Sigma_f)_i$ = Fission source cross section for group i

$\Sigma(j \rightarrow i)_i$ = Scattering cross section from group j to i

Σ_r^i = Removal cross section for group i, and

$$= \Sigma_a^i + \sum_{\substack{j=1 \\ j \neq i}}^{IGM} \Sigma(i \rightarrow j)$$

χ_i = Fission source fraction in group i

k_{eff} = Effective multiplication constant.

The spatial difference equations in 1DB are formulated such that the mesh point is placed in the center of the homogeneous mesh interval. Equations (2.1) and (2.2) are then integrated over the volume associated with each mesh point. For example, for mesh point k, the integration would be from $[r_k - (\delta r_k/2)]$ to $[r_k + (\delta r_k/2)]$.

The leakage terms are obtained by transforming the volume integral over the Laplacian to a surface integral using Green's theorem

$$\int D\nabla^2\phi dV = \int D\nabla\vec{\phi}\cdot d\vec{A} \qquad (2.3)$$

The flux gradients at the mesh boundary are obtained by interpolating the two contiguous flux values. Thus, volume integration of Equation (2.1) for mesh point k leads to the expression

$$\bar{D}_{k,k-1}A_{k,k-1}\frac{\phi_{k-1}-\phi_k}{r_k-r_{k-1}} + \bar{D}_{k+1,k}A_{k+1,k}\frac{\phi_{k+1}-\phi_k}{r_{k+1}-r_k} - \Sigma_r^k\phi_k V_k + S_k V_k = 0 \qquad (2.4)$$

where, for simplicity, the group indices have been omitted, and

$k = 1, 2, \ldots, IM$

IM = Number of mesh intervals

ϕ_k = Flux associated with mesh point k

r_k = Radial position of mesh point k

V_k = Volume associated with mesh point k

S_k = Source rate associated with mesh point k

Σ_r^k = Removal cross section associated with mesh point k

$A_{k,k-1}$ = Area of boundary between mesh point k and mesh point k-1

$\bar{D}_{k,k-1}$ = Effective diffusion constant between mesh point k and mesh point k-1, and

$$= \frac{D_k D_{k-1}(\delta r_k + \delta r_{k-1})}{D_k\delta r_{k-1} + D_{k-1}\delta r_k}$$

If we let

$$\alpha_k = \frac{\bar{D}_{k,k-1}A_{k,k-1}}{r_k - r_{k-1}} \qquad (2.5)$$

and

$$\beta_k = \alpha_k + \alpha_{k+1} + \Sigma_r^k V_k \tag{2.6}$$

Equation (2.4) can be recast into the convenient form

$$-\alpha_k \phi_{k-1} + \beta_k \phi_k - \alpha_{k+1} \phi_{k+1} = S_k V_k \tag{2.7}$$

where:

$k = 1, 2, \ldots, IM.$

2.2 DISCUSSION OF BOUNDARY CONDITIONS

Three boundary conditions are available in 1DB: reflective ($\nabla\Phi = 0$), vacuum ($\Phi = 0$ at $0.71\ \lambda_{tr}$), and periodic.

2.2.1 Reflective

Imagine that a pseudo mesh interval (interval 0) with the same composition and thickness of Interval 1 is added on the left side of the left boundary. If $\nabla\Phi = 0$ at the boundary, then $\Phi_0 = \Phi_1$. Because $(\Phi_0 - \Phi_1)$ vanishes, the coefficient of $\Phi_0 - \Phi_1$ [see Equation (2.4)] is immaterial, and thus α_1 can be set equal to zero.

2.2.2 Vacuum

Again, imagine that a pseudo mesh interval (interval IM+1) with the same composition as interval IM is added to the right side of the right boundary. Because the flux vanishes at $0.71\ \lambda_{tr}$, the appropriate coefficient of $\Phi_{IM} - \Phi_{IM+1}$ (where $\Phi_{IM+1} = 0$), in Equation (2.5), is

$$\alpha_{IM+1} = \frac{D_{IM} A_{IM+1,IM}}{0.5\ \delta r_{IM} + 0.71\ \lambda_{tr}} \tag{2.8}$$

As in the reflective case, there is no contribution of the pseudo flux in Equation (2.4); i.e., for $\nabla\Phi = 0$, $\alpha_1 = 0$, and for $\Phi = 0$ at $0.71\ \lambda_{tr}$, $\Phi_{IM+1} = 0$.

2.2.3 Periodic

In the periodic boundary condition option,

$$\Phi_o \equiv \Phi_{IM}$$

and

$$\Phi_{IM+1} \equiv \Phi_1$$

From Equation (2.5),

$$\alpha_1 = \alpha_{IM+1} = \frac{\overline{D}_{IM,1} A_{IM+1,IM}}{0.5(\delta r_1 + \delta r_{IM})} \qquad (2.9)$$

The pseudo mesh intervals discussed are not in any way a part of the code. They are mentioned here only for heuristic purposes.

2.3 SOLUTION OF DIFFERENCE EQUATIONS

The eigenvalue and spatial flux profiles are computed by standard source-iteration techniques; i.e., by using an initial fission source distribution, the flux profiles in each group are sequentially calculated beginning in the top (highest energy) group. The flux profiles in each group are computed by directly inverting Equation (2.7). Downscatter sources are computed using current-iteration fluxes; upscatter sources are computed using previous iteration fluxes. After the new fluxes in all groups have been calculated, a new fission source distribution is computed. The multiplication ratio, λ, is then obtained by taking the ratio of the new fission source rate to the old (previous iteration) fission source rate. This sequence of events is called an outer iteration.

Before each new outer iteration, the fission spectrum is multiplied by $1/\lambda$, so that λ approaches unity as the iteration proceeds. The effective multiplication constant is simply the product of the successive λs. Convergence is assumed when $|1 - \lambda| < \epsilon$, where ϵ, the eigenvalue convergence criterion, is an input parameter. All fluxes are renormalized before each outer iteration.

Fission source over-relaxation is employed in 1DB to accelerate convergence. The procedure is as follows. After the new fission source rate profile, $F_1^{\nu+1}$, is calculated, a second "new" value, $F_2^{\nu+1}$, is computed by magnifying the difference between the new fission source rate and the old fission source rate, by a factor of ORF which is the over-relaxation factor.

Thus,

$$F_2^{\nu+1} = F^{\nu} + ORF(F_1^{\nu+1} - F^{\nu}) \qquad (2.10)$$

Then, $F_2^{\nu+1}$ is normalized to give the same total source as $F_1^{\nu+1}$.

The adjoint form of Equation (2.1) is solved by transposing the scattering matrix, interchanging the role of χ_i and $(\nu\Sigma_f)_i$, and inverting the group order of the cross sections and fission spectrum. The calculation then proceeds as in a flux calculation. Thus, in an adjoint calculation, the group indices in the output are inverted, and the balance tables lack a direct physical significance.

This page intentionally left blank.

3.0 DISCUSSION OF SEARCH OPTIONS

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

3.1 TIME ABSORPTION (α CALCULATION)

For simplicity, consider the one-group, time dependent diffusion equation

$$\frac{1}{v}\frac{\partial\phi(\vec{r},t)}{\partial t} = D\nabla^2\phi(\vec{r},t) - \Sigma_a\phi(\vec{r},t) + \nu\Sigma_f\phi(\vec{r},t) \qquad (3.1)$$

If we assume that

$$\phi(\vec{r},t) = \phi(\vec{r})e^{\alpha t} \qquad (3.2)$$

Equation (3.1) can be rewritten in the form

$$D\nabla^2\phi(\vec{r}) - \left(\Sigma_a + \frac{\alpha}{v}\right)\phi(\vec{r}) + \nu\Sigma_f\phi(\vec{r}) = 0 \qquad (3.3)$$

In a time absorption calculation, the parameter, α, as defined and used in Equations (3.2) and (3.3), is computed as the eigenvalue. Note that α/v is effectively an absorption cross section--hence, the name "time absorption."

3.2 MATERIAL CONCENTRATION (C CALCULATION)

1DB can perform a flexible and comprehensive criticality search on material composition. Any number of materials can simultaneously be added, depleted, or interchanged in any number of zones.

The format for specifying concentration searches can best be described by a simple example. Suppose that a mixture, Mix 10, is to be composed of two materials mixed at full density, Materials 8 and 9. Further assume that Materials 8 and 9 are to be simultaneously interchanged such that they occupy

a fixed volume fraction, b, of the zone mixture. The I0, I1, and I2 vectors could then be set up as shown in the following tabulation.

Mix number (I0)	Materials number (I1)	Material density (I2)
10	0	0
10	8	1.0
10	9	-1.0
10	10	0
10	8	a - 1.0
10	9	b - a + 1.0

The first row (10,0,0) instructs the code to clear the storage area for Mix 10. The second row (10,8,1.0) and third row (10,9,-1.0) cause Material 8 and Material 9 to be added to Mix 10 with densities of 1.0 and -1.0, respectively. The fourth row (10,10,0) causes the current contents of Mix 10 to be multiplied by the eigenvalue. Finally, rows five (10,8,a - 1.0) and six (10,9,b - a + 1.0) instruct the code to add Materials 8 and 9 to Mix 10 with densities of a - 1.0 and b - a + 1.0, respectively.

This can be summarized by the expression

$$\Sigma_{10} = \Sigma_8 EV - \Sigma_9 EV + (a - 1)\Sigma_8 + (b - a + 1)\Sigma_9 \qquad (3.4)$$

where:

Σ_{10} = Macroscopic cross section for Mix 10

Σ_8 = Full density cross section for Material 8

Σ_9 = Full density cross section for Material 9

EV = Eigenvalue.

For an initial eigenvalue guess of 1.0, Equation (3.4) reduces to $\Sigma_{10} = a\Sigma_8 + (b - a)\Sigma_9$. Therefore, a and (b - a) are simply the initial volume fractions of Materials 8 and 9, respectively.

3.3 ZONE DIMENSIONS (δ CALCULATION)

1DB searches on reactor dimensions by varying the dimensions of each mesh interval. Each mesh width, δr_k, is computed from the expression

$$\delta r_k = \delta r_k^o \, [1 + (\text{dimensional modifier})_k \, EV] \qquad (3.5)$$

where:

δr_k^o = Initial mesh spacing

EV = Eigenvalue.

Different dimensional modifiers can be specified for each zone.

3.4 BUCKLING (B^2 CALCULATION)

In a buckling search, the quantity $D_i \gamma B^2$ (where D_i is the zone dependent diffusion constant for group i, and γ is the zone dependent buckling modifier) is added to the i^{th} group absorption cross section. The in-group scattering cross section, σ_{gg}^i, is reduced by the same amount so that the calculated total cross section remains equal to the input total cross section. The buckling is then computed as the eigenvalue.

This page intentionally left blank.

4.0 CALCULATION OF ISOTOPIC BURNUP

The basic burnup equation for each zone is

$$\frac{dN^i}{dt} = -\lambda^i N^i - \bar{\sigma}_a^i \bar{\Phi} N^i + \sum^k \{y_d^{ki} \lambda^k N^k + y_c^{ki} \bar{\sigma}_c^k \bar{\Phi} N^k \quad (3.5)$$

$$+ y_n^{ki} \bar{\sigma}_{n,2n}^k \bar{\phi} N^k + y_f^{ki} \bar{\sigma}_f^k \bar{\phi} N^k\} \quad (4.1)$$

where:

N^i = Density of nuclide i

λ^i = Decay constant for nuclide i

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

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

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

$\bar{\sigma}_{n,2n}^i$ = Spectrum averaged (n,2n) cross section for nuclide i

y^{ki} = Fractional yield for isotope i by isotope k

$\bar{\Phi}$ = Total flux.

The last four terms in Equation (4.1) allow provision for decay, capture, (n,2n), and fission sources. The latter option, for example, could be used to compute the fission product buildup.

Each input time step is arbitrarily subdivided into 10 smaller time steps. Equation (4.1) is then solved as a march-out problem using the subdivided time intervals. Equation (4.1) can be rewritten in the form

$$\frac{d\vec{N}}{dt} = \vec{f}(\vec{N}, t) \quad (4.2)$$

The particular march-out algorithm used can be written as

$$\vec{N}_{J+1} = \vec{N}_J + \frac{\delta t}{2}(\vec{f}_J + \vec{f}_{J+1}) \tag{4.3}$$

where:

J = Index on time

δt = Fine-step time interval.

Equation (4.3) is implicit because $\vec{N}_{J+1}$ must be known in order to compute $\vec{f}_{J+1}$. Therefore, N must be iterated at each time point. This procedure leads to the algorithm

$$\vec{N}^{\nu+1}_{J+1} = \vec{N}_J + \frac{\delta t}{2}\ (\vec{f}_J + \vec{f}^{\nu}_{J+1}) \tag{4.4}$$

where:

ν = Iteration index.

4.1 REMARKS ON BURNUP EQUATIONS

The zone-averaged flux and cross sections appearing in Equation (4.1) are computed before each input time step. The total reactor power (from the burnable isotopes) and flux profile (relative zone fluxes) are held constant during the fine-step march-out described in Equation (4.4).

The mathematical model presented shows that relatively short time steps should be employed if rapid variations in isotopic concentration or flux profiles are anticipated. Such conditions, however, are rarely encountered in reactor design calculations.

5.0 CALCULATION OF MATERIAL ACTIVITY RATES

The activity distribution (i.e., reaction rate versus radius) can be computed for any number of materials and cross sections. The materials can be either input microscopic data or computed macroscopic mixes. The specific cross section to be used for each activity is specified by giving its position in the cross section file. This position order is shown below.

File position	Cross section (Group g)
. . .	. . .
1	n,2n
2	Fission
3	Absorption
4	Nu x fission
5	Transport
. . .	. . .
6	Sigma (g+2 → g) (Upscattering terms)
7	Sigma (g+1 → g)
8	Scattering, sigma (g → g)
9	Sigma (g-1 → g) (Downscattering terms)
10	Sigma (g-2 → g)
. . .	. . .

The file position is problem dependent because a variable number of terms can precede (n,2n), and also a variable number of terms can be used for upscatter and downscatter.

This page intentionally left blank.

6.0 CALCULATION OF COLLAPSED CROSS SECTIONS

1DB will collapse any IGM group cross section set (microscopic or macroscopic) to a NCR group set (IGM $\geq$ NCR $\geq$ 1) using the fluxes from any specified zone. The regrouping is determined by simply specifying the number of fine groups in each collapsed group. Thus, the energy bounds of the collapsed groups correspond to those in the original group structure. The collapsing scheme is as follows:

I = Collapsed energy group index

Φ_I = Total flux in collapsed group I

$\bar{\sigma}^I$ = Average microscopic cross section for the I^{th} collapsed group

$\bar{\sigma}(I \rightarrow J)$ = Scattering cross section from group I to group J.

For simplicity, the zone indices have been omitted. So, Φ_I is now defined by the equation

$$\Phi_I = \sum^{I} \Phi_i \tag{6.1}$$

where the symbol $\sum^{I}$ denotes a sum over the initial energy groups comprising the I^{th} collapsed group.

The fission, absorption, and neutron emission cross sections are calculated such that reaction rates are unaltered by the averaging procedure. Therefore, for $\overline{\sigma}_f$, $\overline{\sigma}_a$, $\overline{\nu\sigma}_f$, and $\overline{\sigma}_{n,2n}$,

$$\overline{\sigma}^I = \frac{\sum^{I} \sigma^i \Phi_i}{\Phi_I} \tag{6.2}$$

For the transport cross section, normalized reciprocal weighing is used. That is, Equation (6.2) is used with sigma replaced by the reciprocal of sigma, and the final value is normalized to conserve leakage.

The collapsed upscattering and downscattering cross sections are calculated by matching the transfer rates. Thus,

$$\bar{\sigma}(I \rightarrow J) = \frac{\sum^{I}\sum^{J} \sigma(i \rightarrow j)\ \Phi_i}{\Phi_I} \qquad (6.3)$$

Finally, the in-group scattering cross section is calculated from the identity

$$\bar{\sigma}^{I}_{gg} = \bar{\sigma}^{I}_{tr} - \bar{\sigma}^{I}_{a} - \sum^{J} \bar{\sigma}(I \rightarrow J) \qquad (6.4)$$

As an input option, cross sections can be collapsed over an entire cell (all zones) rather than over a single zone.

7.0 CALCULATION OF RESONANCE-SHIELDED CROSS SECTIONS FORMAT

Cross section data in the shielding factor format is given in three sections: infinite dilution cross sections, scattering matrices, and resonance self-shielding factors. A brief description of the format and treatment of this data is given in the following paragraphs.

7.1 INFINITE DILUTION CROSS SECTIONS AND SCATTERING MATRICES

For each energy group of each isotope, nine parameters are given. These are defined as follows:

σ_t = Transport cross section

σ_f = Fission cross section

ν = Neutrons per fission

σ_c = Capture cross section

σ_{in} = Total inelastic scattering cross section

σ_{el} = Total elastic scattering cross section

μ_e = Average cosine of elastic scattering angle

$\sigma_{n,2n}$ = n,2n cross section

σ_k = Kerma (heating) cross section.

The second data block contains the inelastic and elastic scattering matrices, $\sigma(i \rightarrow j)$. These matrices contain IGM x IGM terms, where IGM is the number of energy groups.

The code cross-checks a substantial portion of the above input data using the equations

$$\sigma_{in,i} = \sum_j \sigma_{in}(i \rightarrow j) \qquad (7.1)$$

and

$$\sigma_{el,i} = \sum_j \sigma_{el}(i \rightarrow j) \qquad (7.2)$$

7.2 RESONANCE SELF-SHIELDING FACTORS

Self-shielding factors are used to account for flux depression in the vicinity of large resonances. For each energy group of each isotope, self-shielding factors are listed for fission, capture, transport, elastic scattering, and kerma. In simple terms,

$$\bar{\sigma} = f(T, \sigma_o)\ \sigma \tag{7.3}$$

where:

$\bar{\sigma}$ = Resonance-shielded cross section

σ = Infinite dilution cross section

f = Resonance self-shielding factor

T = Temperature

σ_o = Sum of total cross sections of all other elements in the medium per atom of the isotope under discussion.

The resonance self-shielding factors are compiled for discrete values of σ_o and temperature. A representative format of the input data (for a temperature dependent isotope) is pictured in Table 7-1. Up to 10 rows of shielding factors can be given (if the f factors are temperature dependent), and up to 10 columns of discrete σ_o values can be specified.

Table 7-1. Format for Resonance Self-Shielding Factors.

Temperature, K	σ_0^1	σ_0^2	σ_0^3
T_a	${}^1f_a$	${}^2f_a$	${}^3f_a$
T_b	${}^1f_a$	${}^2f_b$	${}^3f_b$
T_c	${}^1f_c$	${}^2f_c$	${}^3f_c$

The next two sections describe the interpolation equations employed to compute shielding factors for any temperature and total cross section.

7.3 TEMPERATURE INTERPOLATION EQUATION

For each σ_o column, the shielding factor corresponding to the current temperature is computed by the equation

$$f(T) = \alpha + \beta \ln T \tag{7.4}$$

where α and ß are determined using the two nearest points. Although other interpolation schemes may be preferable in some instances, Equation (7.4) appears to be satisfactory for most reactor analyses.

7.4 σ_o INTERPOLATION EQUATION

The interpolation scheme for σ_o is similar to the interpolation scheme for temperature. That is,

$$f(\sigma_o) = \alpha + \beta \ln \sigma_o \tag{7.5}$$

where α and ß are determined using the nearest two points. If σ_o is greater than the largest tabulated total cross section, σ_o^L, or less than the smallest cross section, σ_o^S, the following approximations are employed:

$$\sigma_o = 10\ \sigma_o^L \tag{7.6}$$

$$\sigma_o = 0.1\sigma_o^S \tag{7.7}$$

7.5 ITERATION TO OBTAIN TOTAL CROSS SECTIONS

Because σ_o of each isotope cannot be obtained until the total cross section of all other isotopes is known, iteration is required to obtain both the shielding factors and the total cross sections.

An initial estimate of σ_o for each energy group of the k^{th} isotope is

$$\sigma_{o,k}^1 = \frac{1}{N_k} \sum_{j \neq k} PSF_j\ N_j \sigma_{t,j} \tag{7.8}$$

where:

$\sigma_{o,k}$ = Total cross section (excluding itself) per atom of the k^{th} isotope

$\sigma_{t,j}$ = Total cross section of j^{th} isotope

N_k = Atom density of k^{th} isotope

PSF_j = Potential scattering factor (approximately 1.0 for light isotopes, about 0.2 for uranium).

Subsequent approximations to the total cross section are obtained using the algorithm

$$\sigma_{o,k}^{\nu+1} = \frac{1}{N_k} \sum_{j \neq k} PSF\ N_j\ f_{t,j}^{\nu}\ \sigma_{t,j} \qquad (7.9)$$

where ν is the iteration index. The symbol f^{ν} indicates that f has been evaluated using the ν^{th} value of σ_o. Typical problems converge in just two iterations; hence, the number of iterations on $\sigma_{o,k}$ has been set equal to two. The effect of heterogeneity can be approximated by using modified values of the potential scattering factor, PSF, which is user input.

7.6 EQUATIONS FOR MULTIGROUP CROSS SECTIONS

The 1DB code uses cross sections in the "DTF" format:

$$\ldots\ \bar{\sigma}_{n,2n}^{i}\ \bar{\sigma}_{f}^{i}\ \bar{\sigma}_{a}^{i}\ \nu\bar{\sigma}_{f}^{i}\ \bar{\sigma}_{tr}^{i}\ \ldots\bar{\sigma}(i+1\rightarrow i)\ \bar{\sigma}(i\rightarrow i)\ \bar{\sigma}(i-1\rightarrow i)\ \ldots$$

The bar indicates that the cross sections include resonance self-shielding. Any number of special purpose cross sections can precede the (n,2n) cross section.

These cross sections are computed from data in the "shielding factor" format using the following equations:

$$\bar{\sigma}_{f}^{i} = f_{f}^{i}\ \sigma_{f}^{i} \qquad (7.10)$$

$$\bar{\sigma}_{a}^{i} = f_{f}^{i}\ \sigma_{f}^{i} + f_{c}^{i}\ \sigma_{c}^{i} \qquad (7.11)$$

$$(\overline{\nu\sigma_f})^i = \nu^i \, f_f^i \, \sigma_f^i \tag{7.12}$$

$$\overline{\sigma}_{tr}^i = f_{tr}^i \, \sigma_{tr}^i \tag{7.13}$$

$$\overline{\sigma}(i \pm k \rightarrow i) = f(i \pm k) \, \sigma_{el} \, (i \pm k \rightarrow i) + \sigma_{in}(i \pm k \rightarrow i) \tag{7.14}$$

$$\overline{\sigma} \, (i \rightarrow i) = \overline{\sigma}_{tr}^i - \overline{\sigma}_a^i - \sigma_{in} - \overline{\sigma}_{el} \tag{7.15}$$

This page intentionally left blank.

8.0 REFERENCES

Abagyan, L. P., N. O. Bazazyants, I. I. Bondarenko, M. N. Nikolaev, 1964, *Group Constant for Nuclear Reactor Calculations*, Consultants Bureau, New York.

Alcouffe, R. E., et al., 1984, *User's Guide for Twodant: A Code Package for Two-Dimensional, Diffusion-Accelerated, Neutral-Particle, Transport*, LA-10049-M, Los Alamos, New Mexico.

Hardie, R. W., and W. W. Little, 1967, *1DX, A One-Dimensional Diffusion Code for Generating Effective Nuclear Cross Sections*, BNWL-954, Pacific Northwest Laboratory, Richland, Washington.

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

This page intentionally left blank.

APPENDIX A

INPUT INSTRUCTIONS

This page intentionally left blank.

APPENDIX A

INPUT INSTRUCTIONS

The following pages describe the input data for 1DB. Most input--except hollerith and binary data files--is read in via generalized input routines. The following conventions are used to expedite input:

R n v = Repeat value (v) n times (e.g., R3 3.1 = 3.1 3.1 3.1)

I n v = Interpolate n points between previous value and v (e.g., if previous value were 0, then: I2 3 = 1 2 3)

C n m = Cycle, n times, previous m values (e.g., if last two values had been 80 and 90, then: C3 2 = 80 90 80 90 80 90)

F = Fill array with previous value

T = Terminate array

/ = Ignore remaining columns on card.

Two special formats should be noted. First, the integer control parameters (Card 2) and floating control parameters (Card 3) are input by name. That is, the parameter name followed by the value. If a name is not used, a default value is assumed. Second, for the Material Mix card, Card 11, the format is Isotope Number (integer) followed by density (floating).

Any entries beyond Column 80 are ignored. Following convention, the input is assumed to be on Unit 5.

For each array, if the correct number of data fields is not present, the code will stop with an error message. The usual assumptions for integer and floating formats apply.

Variable	Field	Description
CARD 1: Hollerith		To run a series of cases, repeat from this card.
ID		Identification card.
CARD 2: Integer		
A02		Problem type: = 0, regular calculation, = 1, adjoint calculation.

Variable	Field	Description
NEV		Eigenvalue type: = 1, keff, = 2, time absorption, = 3, concentration, = 4, zone thickness, = 5, buckling (B^2).
SO2		Parametric eigenvalue type: = 0, none, = 1, keff, = 2, alpha.
IGM		Number of energy groups.
NXCM		Number of downscattering terms.
NXCN		Number of upscattering terms.
NST		Number of added cross section slots (before n,2n).
ML		Number of input cross section materials from Unit 15.
NFX		Initial flux guess option: = 0, none, = 1, fluxes from Unit 14.
NPRT		Print option: = 0, mini print, = 1, print all fluxes, = 2, print all cross sections, = 3, print everything.
NPUN		Flux dump option: = 0, none, = 1, write fluxes to Tape 16.
NOIT		Maximum number of outer iterations.
NRCX		Number of generated cross sections from shielding factors (0 for DTF format).
NRCM		Number of mixes used in generating resonance shielded cross sections. If NRCM = 0, DTF format.
NMIX		Total number of mixes.

Variable	Field	Description
IGE		Geometry option: = 0, plane, = 1, cylinder, = 2, sphere.
IM		Number of space intervals (>=3).
IZM		Number of material zone.
BO1		Left boundary condition: = 0, vacuum, = 1, reflective, = 2, periodic.
BO2		Right boundary condition.
NACT		Number of activations (if 0, no activations).
NCG		Number of collapsed groups (if 0, no group collapsing). If IGM group data is desired as output, set NCG equal to IGM and continue as in a group collapsing calculation.
NXINP		Number of collapsed downscattering terms. 1DB also calculates this quantity, NXCMP. If NXINP < 0, NXCMP is used.
NXIMP		Number of collapsed upscattering terms. If NXIMP < 0, code value is used.
NCM		Number of collapsed materials.
NOXS		Collapsed cross section output option: = 0, print, = 1, write binary data to Unit 13. = 2, write ASCII data to Unit 13.
NIXS		Cross Section input option: = 0, cards, = 1, read binary data from Unit 15, = 2, read BCD data from Unit 15.
MAXE		Maximum number of cross section error flags. Print is suppressed if number exceeds MAXE.
CARD 3: Floating		
EV		Initial eigenvalue guess (used only in search calculations).

Variable	Field	Description
EVM		Initial eigenvalue modifier. This value should decrease keff--i.e., EV + EVM should produce a lower keff than EV. Since EV and EVM are completely problem dependent, no representative values can be given. However, these parameters are important, so some thought should be given to estimating a reasonable value. (Used only in search calculations.)
S03		Parametric eigenvalue.
BUCK		Buckling (cm^{-2}). Caution--search calculations that include a buckling term cannot be performed using input cross sections (mixes) directly in zones. Furthermore, a given input mix cannot be used directly in two or more zones in keff or search problems that have a buckling term. These problems can be avoided by mixing with a density of 1.0. If searching on buckling, BUCK should be zero.
LAL		Lower limit on ABS(lambda - 1), where (lambda - 1) is, in essence, the predicted change in the current reactivity. After LAL is reached, the eigenvalue slope is no longer altered. LAL is used only in search calculations. Recommended value: 0.005.
LAH		Upper limit on ABS(lambda - 1). If ABS(lambda - 1) is greater than LAH, LAH rather than ABS(lambda - 1) is used in predicting the new eigenvalue. LAH is used only in search calculations. Recommended value: 0.5.
EPS		Convergence criterion on the total fission source rate.
EPSA		Parametric eigenvalue convergence criterion. The eigenvalue is recalculated when the difference between the last two estimates is less than EPSA. EPSA is only used in search calculations only. Recommended value: 10 x EPS.
POD		Parameter oscillation damper. Ratio of the computed eigenvalue change to the predicted eigenvalue change. POD can be used to accelerate convergence or damp oscillations. Recommended value: 1.0. (Used only in search calculations.)

Variable	Field	Description
ORF		Over-relaxation factor. If instabilities arise, reduce ORF. Recommended value: 1.3.
SOI		If SOI is negative, the total power is normalized to ABS(SOI) Mwt. If positive, SOI = total neutron source/keff.
TMAX		Maximum time (minutes). If TMAX = .0, no time limit is imposed.
CARD 4: Floating (ML values)		
FEF	1	Mev per fission for first input material.
FEF	2	Mev per fission for second input material.
...		
CARD 5: Floating (IM+1 values)		
RO	1	Spatial position (cm) of first mesh boundary (0.0).
RO	2	Spatial position of second mesh boundary.
...		
CARD 6: Integer (IM values)		
MO	1	Zone number for first mesh interval.
MO	2	Zone number for second mesh interval.
...		
CARD 7: Integer (IZM values)		
M2	1	Material number for first zone.
M2	2	Material number for second zone.
...		
CARD 8: Floating (IZM values) Optional--required if BUCK≠0 or if I04=5.		
GAM	1	Buckling modifier for first zone.
GAM	2	Buckling modifier for second zone.
...		
CARD 9: Floating (IGM values)		
K7	1	Fission fraction (spectrum) in first energy group.

Variable	Field	Description
K7	2	Fission fraction in second energy group.
...		
CARD 10: Floating (IGM values) Optional--required if SO2 = 2 or NEV = 2.		
V7	1	Neutron velocity for first energy group (cm/sec).
V7	2	Neutron velocity for second energy group.
CARD 11: Integer and floating Optional--required if NMIX>0.		NMIX pairs of entries needed.
I1	1	Material number of Mix 1 (integer).
I2	2	Concentration of Mix 1 (floating).
...		
I1	3	Material number of Mix 2.
...		
CARD 12: Floating (IZM values) Optional--required if NEV=4.		
R3	1	Dimensional modifier for first zone (if zero, zone width is held constant).
R3	2	Dimensional modifier for second zone.
...		
CARD 13: Integer (NACT values) Optional--required if NACT>0.		
ACTM	1	Material number for first activation.
ACTM	2	Material number for second activation.
...		
CARD 14: Integer (NACT values) Optional--required if NACT>0.		
ACTP	1	Cross section position for first activation (1 = n,2n, 2 = fission, etc.). If ACTM = 0, ACTP becomes group flux number. And if ACTP<0, cumulative flux is computed.
ACTP	2	Cross section position for second activation.
...		

Variable	Field	Description
CARD 15: Integer (NCG values) Optional--required if NCG>0.		
NPN	1	Number of original groups in first collapsed group.
NPN	2	Number of original groups in second collapsed group.
...		
CARD 16: Integer (NFGM values) Optional--required if NCG>0		
NFP	1	Material number of first material to be collapsed. If NFP negative, cell averaged cross section for material number abs (NFP) computed.
NFP	2	Material number of second material to be collapsed.
...		
CARD 17: Integer (NCM values) Optional--required if NCG>0.		
NZN	1	Zone number of fluxes used for first collapsed material.
NZN	2	Zone number of fluxes used for second collapsed material.
...		
CARD 18: Integer (NRCM + NRCX values) Optional--required if NRCM>0.		
J1	1	Not used (0).
J1	2	Number of first input isotope in first cross section mix ($\leq$ML).
J1	3	Number of second input isotope in first cross section mix....
J1	x	Not used (0).
J1	x+1	Number of first input isotope in second cross section mix ($\leq$ML).
...		

Variable	Field	Description
CARD 19: Floating (NRCM + NRCX values) Optional--required if NRCM>0.		
ATEM	1	Not used (.0).
ATEM	2	Temperature of first input isotope in first cross section mix (degrees Kelvin).
ATEM	3	Temperature of second input isotope in first cross section mix.
...		
CARD 20: Floating (NRCM + NRCX values) Optional--required if NRCM>0.		
PSF	1	Not used (0).
PSF	2	Potential scattering factor for first input isotope in first cross section mix. Recommended value about 1.0 for light isotopes, and about .2 for heavy isotopes.
PSF	3	Potential scattering factor for second isotope in first cross section mix.
...		
CARD 21 (a): Hollerith (Columns 1-8) Optional--required if NIXS = 0.		
HOLN	1	Name of first input isotope.
CARD 21(b): Floating (1 value)		
ATW	1	Atomic weight of first isotope.
CARD 21(c): Floating (NST+NXCM+NXCN+5 values)		
	1 through NST	NST entries of additional (optional) processes.
C	NST+1	n,2n cross section (barns)--for first group of first isotope (if NST=0).
C	NST+2	Fission cross section (barns)--for first group of first isotope.
C	NST+3	Absorption cross section.
C	NST+4	Nu * fission cross section.

Variable	Field	Description
C	NST+5	Transport cross section.
	...	
C	NST+6	Upscatter, (i+1->i) ...
C	NST+7	In-group scatter, (i -> i)
C	NST+8	Downscatter, (i-1 -> i) ...
...		

Repeat through group IGM, beginning each new group on a new card. Repeat from Card 21(a) for ML isotopes.

CARD 22: Integer (ML values)
Optional--required if NRCM>0.

NUT	1	Sequence number on cross section tape of first input isotope. If negative, isotope read from file named XS _____.DAT, where _____ = abs (NUT). For example, NUT = -3 would correspond to file name XS0003.DAT.
NUT	2	Sequence number on cross section tape of second input isotope.
...		

CARD 23: Floating

TEMP	1	Program control: = 0, end problem = N, read burnup data for N isotopes, = -N, take time step of |N| days.

Cards 24-29 are read only if NCON>0.

CARD 24: Integer (NCON values)

MATN	1	Material sequence number for first burnable isotope.
MATN	2	Material sequence number for second burnable isotope.
...		

CARD 25: Floating (NCON values)

ALAM	1	Decay constant for first burnable isotope (days>1).
ALAM	2	Decay constant for second burnable isotope.
...		

Variable	Field	Description
CARD 26: Integer (NCON values)		
NBR	1	Category designation for first burnable isotope (0/1/2 = none/fertile/fissile). Used only in breeding ratio calculation.
NBR	2	Designation for second burnable isotope.
...		
CARD 27: Integer (NCON values)		
NID	1	Number of decay sources for first burnable isotope.
NID	2	Number of decay sources for second burnable isotope.
...		
CARD 28: Integer (Sum NID values)		
IND	1	Burnable isotope number for first decay source. If no decay sources, card not required.
IND	2	Burnable isotope number for second decay source.
...		
CARD 29: Floating (Sum NID values)		
YDN	1	Fractional yield for first decay source.
YDN	2	Fractional yield for second decay source.
...		

Repeat Cards 27-29 for n,2n, capture, and fission. Repeat Card 23 (Control).

* * * * 1DB INPUT * * * *

*IDENTIFICATION CARD

*CONTROL CARD (28 INTEGERS)

AO2	0/1=REGULAR CALCULATION/ADJOINT CALCULATION
NEV	EIGENVALUE TYPE (1/2/3/4/5=K/ALPHA/CONC/DELTA/BUCK)
SO2	PARAMETRIC EIGENVALUE TYPE (0/1/2=NONE/K/ALPHA)
IGM	NUMBER OF ENERGY GROUPS
NXCM	NUMBER OF DOWNSCATTERING TERMS
NXCN	NUMBER OF UPSCATTERING TERMS
NST	NUMBER OF ADDED SLOTS (BEFORE N,2N)
ML	NUMBER OF INPUT MATERIALS
NFX	FLUX GUESS (0/1=NONE/TAPE 15)
NPRT	PRINT OPTION (0/1/2/3=MINI/FLUXES/XS/ALL)
NPUN	FLUX DUMP (0/1=NONE/TAPE 16)
NPO2	MAXIMUM NUMBER OF OUTER ITERATIONS
NRCM	NO. OF MIXES USED IN GENERATING XS DATA (0 FOR DTF)
NRCX	NO. OF SHIELDED XS GENERATED
IGE	GEOMETRY (0/1/2=PLANE/CYLINDER/SPHERE)
IM	NUMBER OF SPACE INTERVALS
IZM	NUMBER OF MATERIAL ZONES
NMIX	TOTAL NUMBER OF MIXES
BO1	LEFT BOUNDARY (0/1/2=VACUUM/REFL./PERIODIC)
BO2	RIGHT BOUNDARY (0/1/2=VACUUM/REFL./PERIODIC)
NACT	NUMBER OF ACTIVITIES
NCG	NUMBER OF COLLAPSED GROUPS (IF 0, NO EFFECT)
NXIMP	NO. OF COLLAPSED DOWNSCATTER TERMS (IF <0, BY 1DB)
NXINP	NO. OF COLLAPSED UPSCATTER TERMS (IF <0, BY 1DB)
NCM	NUMBER OF COLLAPSED MATERIALS
NOXS	COLLAPSED XS (0/1=PRINT/UNIT 13 BINARY/UNIT 13 BCD)
NIXS	INPUT XS (CARDS/UNIT 15 BINARY/UNIT 15 BCD)
MAXE	MAXIMUM NUMBER OF ERROR FLAGS

*CONTROL CARD (12 FLOATING)

EV	FIRST EIGENVALUE GUESS
EVM	EIGENVALUE MODIFIER
SO3	PARAMETRIC EIGENVALUE
BUCK	BUCKLING (CM-2)
LAL	LAMBDA LOWER
LAH	LAMBDA UPPER
EPS	EIGENVALUE CONVERGENCE CRITERION
EPSA	PARAMETER CONVERGENCE CRITERION
POD	PARAMETER OSCILLATION DAMPER
ORF	OVER-RELAXATION FACTOR
SO1	NEG./POS.=POWER (MWT)/NEUTRON SOURCE RATE
TMAX	MAXIMUM TIME (MINUTES--IF 0, NO EFFECT)

```
*FISSION ENERGY (MEV/FISSION)
         FEF           ML TERMS (FLOATING)

*MESH BOUNDARIES
          RO           IM+1 TERMS (FLOATING)

*ZONE NUMBERS BY MESH INTERVAL
          MO           IM TERMS (INTEGER)

*MATERIAL NUMBERS BY ZONE
          M2           IZM TERMS (INTEGER)

*BUCKLING MODIFIERS BY ZONE (OPTIONAL--IF BUCK<>0 OR I04=5)
     GAM               IZM TERMS (FLOATING)

*FISSION FRACTIONS
          K7           IGM TERMS (FLOATING)

*NEUTRON VELOCITY (OPTIONAL--IF S04=2 OR NEV=2)
     V7                IGM TERMS (FLOATING)

*MIXTURE SPECIFICATIONS (MAT. NUMBERS AND DENSITIES FOR MIX--OPTIONAL--
                         IF NMIX)
        I1 AND I2  (INTEGER AND FLOATING)

*MATERIAL NUMBERS FOR ACTIVATION (OPTIONAL--IF NACT>0)
   ACTM              NACT TERMS (INTEGER)

*CROSS SECTION POSITION FOR ACTIVATION (OPTIONAL--IF NACT>0)
        ACTP         NACT TERMS (INTEGER)

*NUMBER OF GROUPS PER CRUNCHED GROUP (OPTIONAL--IF NCG)
         NPN         NCG TERMS (INTEGER)

*MATERIAL NUMBERS TO BE CRUNCHED (OPTIONAL--IF NCG)
         NFP         NCM TERMS (INTEGER)

*ZONE NUMBERS OF FLUXES USED FOR CRUNCH CALCULATION (OPTIONAL--IF NCG)
         NZN         NCM TERMS (INTEGER)

*MATERIAL NUMBERS FOR EACH MIX (OPTIONAL--IF NRCM)
          J1         NRCM = NRCX TERMS (INTEGER)

*TEMPERATURE OF EACH MATERIAL FOR EACH MIX (OPTIONAL--IF NRCM)
   ATEM              NRCM + NRCX TERMS (FLOATING)

*POTENTIAL SCATTERING FACTOR OF EACH MATERIAL (OPTIONAL--IF NRCM)
         PSF         NRCM + NRCX TERMS (FLOATING)

*CROSS SECTIONS (OPTIONAL--IF NIXS=0--REPEAT SEQUENCE FOR ML ISOTOPES)
  NAME               1 TERM (8 CHARACTERS)
  WEIGHT             1 TERM (FLOATING)
  VALUES             ITL TERMS (FLOATING)--REPEAT FOR EACH ENERGY GROUP
```

```
*CROSS SECTION SEQUENCE NUMBERS ON DATA TAPE (OPTIONAL--IF NRCM>0)
        PUP       ML TERMS (INTEGER)

*CONTROL CARD
        NCON            1 TERM (FLOATING)
                        =  0, END PROBLEM
                        = +N, READ BURNUP DATA FOR N ISOTOPES
                        = -N, TAKE TIME STEP OF ABS(N) DAYS

BURNUP DATA (ALL OPTIONAL--IF NCON > 0):

*MATERIAL SEQUENCE NUMBERS FOR BURNABLE ISOTOPES
        MATN            NCON TERMS (INTEGER)

*DECAY CONSTANT (DAYS-1) FOR BURNABLE ISOTOPES
        ALAM            NCON TERMS (FLOATING)

*BURNABLE ISOTOPE CATEGORY (0/1/2=NONE/FERTILE/FISSILE)
         NBR            NCON TERMS (INTEGER)

*NUMBER OF DECAY SOURCES FOR EACH BURNABLE ISOTOPE
         NID            NCON TERMS (INTEGER)
*BURNABLE ISOTOPE NUMBERS FOR DECAY SOURCES
                        SUM NID TERMS (INTEGER)
*FRACTIONAL YIELDS FOR DECAY SOURCES
                        SUM NID TERMS (FLOATING)

*NUMBER OF N,2N SOURCES FOR EACH BURNABLE ISOTOPE
         NIN            NCON TERMS (INTEGER)
*BURNABLE ISOTOPE NUMBERS FOR N,2N SOURCES
                        SUM NIN TERMS (INTEGER)
*FRACTIONAL YIELDS FOR N,2N SOURCES
                        SUM NIN TERMS (FLOATING)

*NUMBER OF CAPTURE SOURCES FOR EACH BURNABLE ISOTOPE
         NIC            NCON TERMS (INTEGER)
*BURNABLE ISOTOPE NUMBERS FOR CAPTURE SOURCES
                        SUM NIC TERMS (INTEGER)
*FRACTIONAL YIELDS FOR CAPTURE SOURCES
                        SUM NIC TERMS (FLOATING)

*NUMBER OF FISSION SOURCES FOR EACH BURNABLE ISOTOPE
         NIF            NCON TERMS (INTEGER)

*BURNABLE ISOTOPE NUMBERS FOR FISSION SOURCES
                        SUM NIF TERMS (INTEGER)
*FRACTIONAL YIELDS FOR FISSION SOURCES
                        SUM NIF TERMS (FLOATING)

*CONTROL CARD (SEE ABOVE)
```

This page intentionally left blank.

APPENDIX B

SAMPLE PROBLEM

This page intentionally left blank.

```
1DB SAMPLE PROBLEM  (1 GROUP, 2 ISOTOPE, 2 ZONE, 100 DAY BURNUP)
IGE 1  IGM 1    ML 2   NMIX 2   NPRT 0   NIXS 0   NOIT 30
IM 10  IZM 2    B01 1  B02 0    NCG 1    NCM 2    T   CONTROL
EPS .0001       S01 -1.0                          T   CONTROL
R2 200. T     MEV/FISSION
.0 I9 20. T   RADII
R5 1 R5 2 T   ZONE BY MESH
3 4 T         MATERIALS BY ZONE
1.0 T         CHI'S
1 .01 2 .02 T MIX SPECIFICATIONS
2 .04 T
1 T           GROUP COLLAPSING DATA:
1 2 T         MATERIALS
1 2 T            ZONES
PU-239        CROSS SECTIONS  MATERIAL 1
239.   T
0.1 2.5 2.0 7.0 3. 1.0 T
NA-23         CROSS SECTIONS  MATERIAL 2
23.    T
0.0 .0 .1 .0 1.1 1. T
2. T          CONTROL--READ DATA FOR TWO BURNABLE ISOTOPES
1 2 T         MATERIAL NUMBERS
.0 .0 T       DECAY CONSTANTS (DAY-1)
2 0 T         CATEGORY (0/1/2=NONE/FERTILE/FISSILE)
0 0 T         NUMBER OF DECAY SOURCES
0 0 T         NUMBER OF CAPTURE SOURCES
0 0 T         NUMBER OF N,2N SOURCES
0 0 T         NUMBER OF FISSION SOURCES
/             NOTE: ID'S FOR EACH SOURCE NOT REQUIRED
/             NOTE: YIELDS FOR EACH SOURCE NOT REQUIRED
-100. T       TIME STEP (DAYS)
0 T           CONTROL--END PROBLEM
```

```
          * * *  1 D B  (PC VERSION 7/24/90)  * * *

1DB SAMPLE PROBLEM  (1 GROUP, 2 ISOTOPE, 2 ZONE, 100 DAY BURNUP)

    0  A02   0/1=REGULAR CALCULATION/ADJOINT CALCULATION
    1  NEV   EIGENVALUE TYPE (1/2/3/4/5=K/ALPHA/CONC/DELTA/BUCK)
    0  S02   PARAMETRIC EIGENVALUE TYPE (0/1/2=NONE/K/ALPHA)
    1  IGM   NUMBER OF ENERGY GROUPS
    0 NXCM   NUMBER OF DOWNSCATTERING TERMS
    0 NXCN   NUMBER OF UPSCATTERING TERMS
    0  NST   NUMBER OF ADDED SLOTS (BEFORE N,2N)
    2   ML   NUMBER OF INPUT MATERIALS
    0 NRCM   NO. OF MIXES USED IN GENERATING SF XS (0 FOR DTF)
    2 NMIX   TOTAL NUMBER OF MIXES

    1  IGE   GEOMETRY (0/1/2=PLANE/CYLINDER/SPHERE)
   10   IM   NUMBER OF SPACE INTERVALS
    2  IZM   NUMBER OF MATERIAL ZONES
    1  B01   LEFT BOUNDARY  (0/1/2=VACUUM/REFL./PERIODIC)
    0  B02   RIGHT BOUNDARY (0/1/2=VACUUM/REFL./PERIODIC)
    0 NPRT   PRINT OPTION (0/1/2/3=MINI/FLUXES/XS/ALL)
   30 NOIT   MAXIMUM NUMBER OF OUTER ITERATIONS
    0 MAXE   MAXIMUM NUMBER OF ERROR FLAGS

    0 NACT   NUMBER OF ACTIVITIES
    1  NCG   NUMBER OF COLLAPSED GROUPS (IF 0, NO EFFECT)
    0 NCCM   NO. OF COLLAPSED DOWNSCATTER TERMS (IF <0, BY 1DB)
    0 NCCN   NO. OF COLLAPSED UPSCATTER TERMS (IF <0, BY 1DB)
    2  NCM   NUMBER OF COLLAPSED MATERIALS
    0  NFX   FLUX GUESS (0/1=NONE/UNIT 14)
    0 NPUN   FLUX DUMP (0/1=NONE/UNIT 16)
    0 NIXS   INPUT XS  (0/1/2=CARDS/UNIT 15 BINARY/UNIT 15 BCD)
    0 NOXS   OUTPUT XS (0/1/2=PRINT/UNIT 13 BINARY/UNIT 13 BCD)

 0.0000E+00   EV   FIRST EIGENVALUE GUESS
 0.0000E+00  EVM   EIGENVALUE MODIFIER
 0.0000E+00  S03   PARAMETRIC EIGENVALUE
 0.0000E+00 BUCK   BUCKLING (CM-2)
 1.0000E-03  LAL   LAMBDA LOWER
 1.0000E-01  LAH   LAMBDA UPPER

 1.0000E-04  EPS   EIGENVALUE CONVERGENCE CRITERION
 1.0000E-03 EPSA   PARAMETER CONVERGENCE CRITERION
 1.0000E+00  POD   PARAMETER OSCILLATION DAMPER
 1.3000E+00  ORF   OVER-RELAXATION FACTOR
-1.0000E+00  S01   NEG./POS.=POWER (MWT)/NEUTRON SOURCE RATE
 0.0000E+00 TMAX   MAXIMUM TIME (MINUTES--IF 0, NO EFFECT)

 LAST =      352
```

```
FISSION ENERGY (MEV/FISSION)
        FEF     2
 .20000E+03  .20000E+03

MESH POINTS
        RO    11
 .00000E+00  .20000E+01  .40000E+01  .60000E+01  .80000E+01  .10000E+02
 .12000E+02  .14000E+02  .16000E+02  .18000E+02  .20000E+02

ZONE NUMBERS BY MESH INTERVAL
        MO    10
    1     1     1     1     1     2     2     2     2     2

MATERIAL NUMBERS BY ZONE
        M2     2
    3     4

FISSION FRACTIONS
        K7     1
 .10000E+01

MIXTURE SPECIFICATIONS FOR EACH MIX
        MIX (MAT. #, DENSITY) FOR MATERIAL      3,      2 VALUES
    1   .1000E-01     2   .2000E-01
        MIX (MAT. #, DENSITY) FOR MATERIAL      4,      1 VALUES
    2   .4000E-01

NUMBER OF GROUPS PER CRUNCHED GROUP
        NPN     1
    1

NUMBERS OF MATERIALS TO BE CRUNCHED (NEG. FOR CELL)
        NFP     2
    1     2

ZONE NUMBERS OF FLUXES USED FOR CRUNCH CALCULATION
        NZN     2
    1     2


1DB SAMPLE PROBLEM  (1 GROUP, 2 ISOTOPE, 2 ZONE, 100 DAY BURNUP)


THE FOLLOWING NUCLIDES ARE IN THE DTF FORMAT

 1      PU-239
 2      NA-23


                    T I M E  =      0.000E+00 D A Y S
```

MIXTURE SPECIFICATIONS

MIXTURE NUMBER	MIX COMMAND	MATERIAL DENSITY	NAME
1	3	0	0.00000E+00	
2	3	1	1.00000E-02	PU-239
3	3	2	2.00000E-02	NA-23
4	4	0	0.00000E+00	
5	4	2	4.00000E-02	NA-23

1DB SAMPLE PROBLEM (1 GROUP, 2 ISOTOPE, 2 ZONE, 100 DAY BURNUP)

TIME (MINUTES)	OUTER ITERATIONS	EIGENVALUE SLOPE	EIGENVALUE	LAMBDA
.128E-01	0	.00000000E+00	.00000000E+00	.00000000E+00
.139E-01	1	.00000000E+00	.63535860E+00	.63535860E+00
.139E-01	2	.00000000E+00	.64401470E+00	.10136240E+01
.139E-01	3	.00000000E+00	.64107440E+00	.99543440E+00
.139E-01	4	.00000000E+00	.64157060E+00	.10007740E+01
.139E-01	5	.00000000E+00	.64148160E+00	.99986120E+00
.146E-01	6	.00000000E+00	.64149820E+00	.10000260E+01

FINAL NEUTRON BALANCE TABLE

GROUP	FISS. SOURCE	IN-SCATTER	OUT-SCATTER	ABSORPTION	LEFT LEAK.	RIGHT LEAK.
1	1.3623E+17	0.0000E+00	3.3255E+09	3.6056E+16	0.0000E+00	1.0018E+17
2	1.3623E+17	0.0000E+00	3.3255E+09	3.6056E+16	0.0000E+00	1.0018E+17

ZONE		MAT.	RADIUS (CM)	AVG. RADIUS (CM)	TOTAL FLUX (N/CM2-SEC)	POWER (MWT/LITER)	FISSION SOURCE
1	1	3	0.000E+00	1.000E+00	4.706E+15	3.769E+00	3.294E+14
2	1	3	2.000E+00	3.000E+00	4.578E+15	3.667E+00	3.204E+14
3	1	3	4.000E+00	5.000E+00	4.327E+15	3.466E+00	3.029E+14
4	1	3	6.000E+00	7.000E+00	3.964E+15	3.175E+00	2.775E+14
5	1	3	8.000E+00	9.000E+00	3.503E+15	2.806E+00	2.452E+14
6	2	4	1.000E+01	1.100E+01	3.005E+15	0.000E+00	0.000E+00
7	2	4	1.200E+01	1.300E+01	2.630E+15	0.000E+00	0.000E+00
8	2	4	1.400E+01	1.500E+01	2.313E+15	0.000E+00	0.000E+00
9	2	4	1.600E+01	1.700E+01	2.041E+15	0.000E+00	0.000E+00
10	2	4	1.800E+01	1.900E+01	1.803E+15	0.000E+00	0.000E+00
11			2.000E+01				

```
1DB SAMPLE PROBLEM  (1 GROUP, 2 ISOTOPE, 2 ZONE, 100 DAY BURNUP)


MATERIAL INVENTORY (KILOGRAMS) FOR EACH ZONE


MATERIAL   ATOMIC WT.    ZONE  1    ZONE  2

  1 PU-239    239.00    .125E+01   .000E+00
  2 NA-23      23.00    .240E+00   .144E+01


MATERIAL   ATOMIC WT.    TOTAL

  1 PU-239    239.00    .125E+01
  2 NA-23      23.00    .168E+01


ZONE NUMBER    VOLUME (LITERS)

        1         3.142E-01
        2         9.425E-01


BURNUP DATA



MATERIAL SEQUENCE NUMBERS FOR BURNABLE ISOTOPES
       MATN     2
    1     2

DECAY CONSTANT (DAYS-1) FOR BURNABLE ISOTOPES
       ALAM     2
 .00000E+00  .00000E+00

BURNABLE ISOTOPE TYPE (0/1/2=NONE/FERTILE/FISSILE)
        NBR     2
    2     0

NUMBER OF DECAY SOURCES FOR EACH BURNABLE ISOTOPE
        NID     2
    0     0

NUMBER OF N,2N SOURCES FOR EACH BURNABLE ISOTOPE
        NIN     2
    0     0

NUMBER OF CAPTURE SOURCES FOR EACH BURNABLE ISOTOPE
        NIC     2
    0     0
```

NUMBER OF FISSION SOURCES FOR EACH BURNABLE ISOTOPE

NIF	2
0	0

Z O N E 1 FLUX =3.9740E+15 VOLUME =3.1416E-01 LITERS

BURNABLE ISOTOPE	MAT. NO.	NAME	ATOM DENSITY	FISSION RATE	ABSORPTION RATE	SIGMA FISSION	SIGMA ABSORPTION
1	1	PU-239	1.00E-02	3.12E+16	2.50E+16	2.50E+00	2.00E+00
2	2	NA-23	2.00E-02	0.00E+00	2.50E+15	0.00E+00	1.00E-01

Z O N E 2 FLUX =2.2786E+15 VOLUME =9.4248E-01 LITERS

BURNABLE ISOTOPE	MAT. NO.	NAME	ATOM DENSITY	FISSION RATE	ABSORPTION RATE	SIGMA FISSION	SIGMA ABSORPTION
1	1	PU-239	0.00E+00	0.00E+00	0.00E+00	2.50E+00	2.00E+00
2	2	NA-23	4.00E-02	0.00E+00	8.59E+15	0.00E+00	1.00E-01

BREEDING RATIO = .0000

T I M E = 1.000E+02 D A Y S

MIXTURE SPECIFICATIONS

MIXTURE NUMBER	MIX COMMAND	MATERIAL DENSITY	NAME
1	0	0.00000E+00	
2	1	9.31572E-03	PU-239
3	2	1.99292E-02	NA-23
4	0	0.00000E+00	
5	2	3.99188E-02	NA-23

1DB SAMPLE PROBLEM (1 GROUP, 2 ISOTOPE, 2 ZONE, 100 DAY BURNUP)

TIME (MINUTES)	OUTER ITERATIONS	EIGENVALUE SLOPE	EIGENVALUE	LAMBDA
.184E-01	0	.00000000E+00	.64149820E+00	.00000000E+00
.184E-01	1	.00000000E+00	.64149430E+00	.64149430E+00
.193E-01	2	.00000000E+00	.64149590E+00	.10000030E+01

FINAL NEUTRON BALANCE TABLE

GROUP	FISS. SOURCE	IN-SCATTER	OUT-SCATTER	ABSORPTION	LEFT LEAK.	RIGHT LEAK.
1	1.4624E+17	1.7180E+10	3.5697E+09	3.8704E+16	0.0000E+00	1.0753E+17
2	1.4624E+17	1.7180E+10	3.5697E+09	3.8704E+16	0.0000E+00	1.0753E+17

ZONE	MAT.		RADIUS (CM)	AVG. RADIUS (CM)	TOTAL FLUX (N/CM2-SEC)	POWER (MWT/LITER)	FISSION SOURCE
1	1	3	0.000E+00	1.000E+00	5.051E+15	3.769E+00	3.536E+14
2	1	3	2.000E+00	3.000E+00	4.914E+15	3.667E+00	3.440E+14
3	1	3	4.000E+00	5.000E+00	4.645E+15	3.466E+00	3.251E+14
4	1	3	6.000E+00	7.000E+00	4.255E+15	3.175E+00	2.979E+14
5	1	3	8.000E+00	9.000E+00	3.760E+15	2.806E+00	2.632E+14
6	2	4	1.000E+01	1.100E+01	3.225E+15	0.000E+00	0.000E+00
7	2	4	1.200E+01	1.300E+01	2.823E+15	0.000E+00	0.000E+00
8	2	4	1.400E+01	1.500E+01	2.483E+15	0.000E+00	0.000E+00
9	2	4	1.600E+01	1.700E+01	2.191E+15	0.000E+00	0.000E+00
10	2	4	1.800E+01	1.900E+01	1.936E+15	0.000E+00	0.000E+00
11			2.000E+01				

1DB SAMPLE PROBLEM (1 GROUP, 2 ISOTOPE, 2 ZONE, 100 DAY BURNUP)

MATERIAL INVENTORY (KILOGRAMS) FOR EACH ZONE

MATERIAL		ATOMIC WT.	ZONE 1	ZONE 2
1	PU-239	239.00	.116E+01	.000E+00
2	NA-23	23.00	.239E+00	.144E+01

MATERIAL	ATOMIC WT.	TOTAL
1 PU-239	239.00	.116E+01
2 NA-23	23.00	.168E+01

ZONE NUMBER	VOLUME (LITERS)
1	3.142E-01
2	9.425E-01

ZONE AVERAGED FLUXES

ZONE	FLUX (N/CM2-SEC)	VOLUME (LITERS)
1	4.26582E+15	3.14159E-01
2	2.44589E+15	9.42477E-01

COLLAPSED FISSION FRACTIONS AND ZONE AVERAGED FLUXES BY GROUP

GROUP	FISS. FCT.	ZONE 1	ZONE 2
1	1.000E+00	4.266E+15	2.446E+15

COLLAPSED CROSS SECTIONS

GROUP 1

MATERIAL	ZONE	N,2N	SIGF	SIGA	NUSIGF	SIGTR...	GXG...
PU-239	1	1.00E-01	2.50E+00	2.00E+00	7.00E+00	3.00E+00	1.00E+00
NA-23	2	0.00E+00	0.00E+00	1.00E-01	0.00E+00	1.10E+00	1.00E+00

APPENDIX C

CODE LISTING

This page intentionally left blank.

```
C     1DB.INC
      COMMON      NCR1,     NFF,    NINP,    NOUT, NSCRAT,  NSCR1,  NSCR2,
     1             NSM,     NSR,     B07,     CNT,    CVT,  NXCMP,  NXCMR,
     3            IGEP,     IGP,     IGV,     IHS,   NOXS,   NIXS,
     4             IHT,      IP,    IRED,   ITEMP, ITEMP1, ITEMP2,    ITL,
     5            ITLP,   KPAGE,      ME,    NRCX,    P02,    R02,    T06,
     6              NB,      NE,   NGOTO,    NOIT,   NRED,  NSIGO,  NXCNP
      COMMON      MAXE,     A02,     NEV,     S02,    IGM,   NXCM,   NXCN,
     1              ML,     NFX,    NPRT,    NPUN,   NRCM,   NIFF,   MM01,
     2             IGE,      IM,     IZM,      MT,    M01,    B01,    B02,
     3             NCG,    NCCN,    NCCM,    NACT,    NCM,   NMIX,    NST
      COMMON       LRO,     LMO,     LM2,    LGAM,    LK7,    LV7,    LIO,
     1             LI1,     LI2,     LR3,     LNO,   LATW,  LHOLN,   LNUT,
     2           LMPUP,     LNT,    LNGB,    LNGE,  LNPFF,  LNDSN,    LSF,
     3             LSR,     LSM,     LSE,     LFF,  LATEM,   LPSF,   LTEM,
     4            LNXS,    LFSS,  LSIGO,    LNPN,
     5            LNFP,    LNZN,     LNV,     LC2,  LALPH,  LALPS,   LPHJ,
     6            LPHI,     LCO,     LA0,     LR1,    LR4,    LR5,    LV0,
     7             LN2,     LF0,     LF2,     LI3,    LK6,    LS2,   LCXS,
     8            LVOL,   LMASS,     LHA,     LPA,    LEB,   LMAT,  LICHI,
     9              LC,    LAST,   LIDUM,    LTXS,   LEXS,   LFEF
      COMMON      LMATN,    LNBR,  LALAM,  LMASSP,
     1            LNID,     LNIN,    LNIC,    LNIF,
     2            LIND,     LINN,    LINC,    LINF,
     3            LYDN,     LYNN,    LYCN,    LYFN,
     4           LPHIB,     LAXS,    LFXS,   LMACT,  LLACT
      COMMON       SK7,      T7,     T11,     PI2,    TEMP,  TEMP1,  TEMP2,
     1           TEMP3,    TEMP4,     DAY,     V11,   NCON,   DELT,   TMAX,
     2            PBAR,     SBAR,     LAP,    LAPP,    LAR,    LAL,    LAH,
     3             S03,     BUCK,     EPS,    EPSA,    POD,    ORF,    S01,
     4              EV,      EVM,     GLH,     ALA
      COMMON     /CHAR/ID(20)
      INTEGER      A02,     B01,     B02,     B07,    CNT,    CVT,    P02,
     1             R02,     S02,     T06
      REAL         LAH,     LAL,     LAP,    LAPP,    LAR,     N0,     N2,
     1              I2,      I3,      K6,      K7,   MASS,  MASSP
```

```
C     PROGRAM 1DB
C
C     * * * * *  DESCRIPTION OF SUBROUTINES  * * * * *
C
C    INP         CONTROLS READING AND PRINTING OF ALL INPUT DATA (EXCEPT
C                CROSS SECTION DATA).
C
C    FXINP       READS CROSS SECTION DATA.
C
C    ERRO2       PRINTS ERROR MESSAGES.
C
C    CLEAR       SETS AN ARRAY OF A GIVEN LENGTH EQUAL TO A SPECIFIED
C                CONSTANT.
C    REAG2       READS FLOATING POINT DATA IN GENERALIZED FORMAT.
C
C    REAI2       READS INTEGER DATA IN GENERALIZED FORMAT.
C
C    REAIG2      READS MIXTURE SPECIFICATIONS (FLOATING AND INTEGER)
C
C    RCINP1      READS CROSS SECTION DATA IN THE RUSSIAN FORMAT FROM
C                CARDS.
C
C    RCPUP       READS CROSS SECTION DATA IN THE RUSSIAN FORMAT FROM TAPE.
C
C    RCPRT1      PRINTS CROSS SECTION DATA IN THE RUSSIAN FORMAT.
C
C    RCCHK       CHECKS CROSS SECTION DATA IN THE RUSSIAN FORMAT FOR
C                CONSISTENCY.
C
C    RCSTUP      COMPUTES F-FACTORS FOR TEMPERATURE DEPENDENT ISOTOPES
C                AND WRITES THE CROSS SECTION DATA ON TAPE IN THE PROPER
C                ORDER.
C
C    RCCAL1      CALCULATES SIGO.
C
C    RCCAL2      CALCULATES F-FACTORS.
C
C    RCCSS       CALCULATES RESONANCE SHIELDED CROSS SECTIONS.
C
C    RECS        READS CROSS SECTIONS, PERFORMS ADJOINT REVERSALS IF
C                REQUIRED, AND WRITES CROSS SECTION TAPE.
C
C    INIT        PERFORMS ADJOINT REVERSALS ON V7 AND K7, MIXES AND PRINTS
C                CROSS SECTIONS, MODIFIES GEOMETRY, AND CALCULATES AREAS,
C                VOLUMES, EFFECTIVE FISSION SPECTRUM AND FISSION RATE.
C
C    FISCAL      CALCULATES FISSION SUMS AND LAMBDA AND NORMALIZES FLUX
C                AND FISSION SOURCE RATE.
C
C    MONPR       MONITOR PRINT--PRINTS TIME, ITERATION NUMBER, EIGENVALUE
C                SLOPE, EIGENVALUE, AND LAMBDA AFTER EACH OUTER ITERATION.
C
C    OUTER       PERFORMS AN OUTER ITERATION--CALCULATES THE SOURCE INTO
C                EACH GROUP AND THE NEW FISSION AND FISSION SOURCE RATE.
C
C    INNER1      CALCULATES COEFFICIENTS FOR THE FLUX EQUATION.
C
C    INNER       CALCULATES THE FLUX FOR SPECIFIED GROUP.
C
C    CNNP        PERFORMS CONVERGENCE TESTS AND CALCULATES NEW PARAMETERS
C                FOR SEARCH OPTIONS.
C
C    FINPR       FINAL PRINT--PRINTS RADII, AREAS, VOLUMES, FLUXES,
C                POWER, AND FISSION SOURCE RATE.
C
C    NBAL        COMPUTES AND PRINTS BALANCE TABLES.
C
C    GRAM        CALCULATES AND PRINTS MATERIAL INVENTORIES FOR EACH
C                ZONE.
C
C    CRUNCH      CALCULATES, PRINTS, AND PUNCHES COLLAPSED CROSS SECTIONS.
C
C    MARCH       COMPUTES TIME DEPENDENT ISOTOPIC CONCENTRATION
C
C    INPB        READS AND PRINTS INPUT DATA FOR BURNUP
C
```

```
C   AVERAG     CALCULATES ZONE AVERAGED FLUXES AND CROSS SECTIONS
C
C     * * * * *  INTERNAL VARIABLES  * * * * *
C
C     (TAPE 13)     OUTPUT FOR COLLAPSED CROSS SECTIONS
C     (TAPE 14)     INPUT FOR FLUX RESTART
C     (TAPE 15)     INPUT FOR CROSS SECTIONS
C     (TAPE 16)     OUTPUT FOR FLUX DUMP
C     NCR1          SCRATCH CROSS SECTION TAPE (3)
C     NFF           SCRATCH SHIELDING FACTOR TAPE (8)
C     NINP          INPUT TAPE (5)
C     NOUT          OUTPUT TAPE (6)
C     NSCRAT        SCRATCH TAPE (4)
C     NSCR1         SCRATCH TAPE (11) --SWITCHED WITH NSR
C     NSCR2         SCRATCH TAPE (12) --SWITCHED WITH NSM
C     NSM           SCRATCH TAPE FOR SCATTERING MATRIX (10)
C     NSR           SCRATCH TAPE FOR INFINITE DILUTION XS (9)
C     ALA           LAMBDA
C     B07           USED FOR INTERNAL COMPUTATION IN FISCAL AND INIT
C     CNT           CONVERGENCE TRIGGER FOR LAMBDA
C     CVT           CONVERGENCE TRIGGER
C     DELT          TIME STEP IN DAYS
C     E01           TEMPORARY
C     E02           TEMPORARY
C     E03           TEMPORARY
C     EQ            EIGENVALUE SLOPE
C     EVP           PREVIOUS EIGENVALUE
C     EVPP          EIGENVALUE FOR TWO ITERATIONS BACK
C     GBAR          GROUP INDICATOR FOR TAPE MOTION IN OUTER
C     GLH           INITIAL TIME IN SECONDS
C     IGEP          IGE + 1
C     IGP           IGM + 1
C     IGV           GROUP INDICATOR FOR INNER AND OUTER
C     IHS           POSITION OF SIGMA SELF SCATTER
C     IHT           POSITION OF SIGMA TRANSPORT
C     IP            IM + 1
C     IRED          FIRST PASS THROUGH RCCAL1  (1/2=YES/NO)
C     ITEMP         TEMPORARY
C     ITEMP1        TEMPORARY
C     ITEMP2        TEMPORARY
C     ITL           CROSS SECTION TABLE LENGTH
C     ITLP          CRUNCHED CROSS SECTION TABLE LENGTH
C     KPAGE         PAGE COUNTER FOR MONITOR PRINT
C     LAP           LAMBDA FOR PREVIOUS EIGENVALUE
C     LAPP          LAMBDA FOR TWO ITERATIONS BACK
C     LAR           LAMBDA FOR PREVIOUS ITERATION
C     ME            CROSS SECTION DATA INCONSISTENCY COUNT
C     MM01          NO. MIX SPECIF. FOR GENERATING SHIELDED XS
C     MT            TOTAL NUMBER OF MATERIALS INCLUDING MIXES
C     M01           NUMBER OF MIXTURE SPECIFICATIONS
C     NRCX          NO. OF GENERATED SHIELDED XS (MM01 - NRCM)
C     NB            TEMPORARY
C     NCON          BURNUP CONTROL (-1/0/N=TIME STEP/END/DATA)
C     NE            TEMPORARY
C     NGOTO         FLAG SET EQUAL TO ONE IF PROBLEM IS FINISHED
C     NOIT          MAXIMUM NUMBER OF OUTER ITERATIONS
C     NRED          SHIELDING FACTOR CROSS SECTIONS (0/1=NO/YES)
C     NSIGO         SIGO ITERATION COUNT
C     NXCMP         NUMBER OF DOWNSCATTERING TERMS FOR CRUNCHED GROUPS
C     NXCNP         NUMBER OF UPSCATTERING TERMS FOR CRUNCHED GROUPS
C     NXCMR         NXCM + 1
C     P02           OUTER ITERATION COUNT
C     PBAR          TEMPORARY
C     PI2           6.28318
C     SBAR          TEMPORARY
C     SK7           SUM OF K7 OVER ALL GROUPS
C     T06           0/1=NOT DELTA/DELTA CALCULATION
C     T7            ALPHA/VELOCITY
C     T11           PREVIOUS FISSION TOTAL
C     TEMP          TEMPORARY
C     TEMP1         TEMPORARY
C     TEMP2         TEMPORARY
C     TEMP3         TEMPORARY
C     TEMP4         TEMPORARY
C     TI            TIME
C     V11           TOTAL SOURCE FOR THE GROUP
```

```
C
C     * * * * *  INPUT VARIABLES (CARDS 1-5)  * * * * *
C
C     ID(20)         IDENTIFICATION CARD
C     A02            0/1=FLUX CALCULATION/ADJOINT CALCULATION
C     NEV            EIGENVALUE TYPE (1/2/3/4/5=KEFF/ALPHA/CONCENTRATION/
C                    DELTA/BUCKLING)
C     S02            PARAMETRIC EIGENVALUE TYPE  (0/1/2=NONE/KEFF/ALPHA)
C     IGM            NUMBER OF GROUPS
C     NXCM           NUMBER OF DOWNSCATTERING TERMS
C     NXCN           NUMBER OF UPSCATTERING TERMS
C     NST            NUMBER OF ADDED SLOTS (BEFORE N,2N)
C     ML             NUMBER OF INPUT MATERIALS
C     NFX            FLUX GUESS  (0/1=NONE/UNIT 14)
C     NPRT           PRINT OPTION (0/1/2/3=MINI/FLUXES/XS/ALL)
C     NPUN           FLUX DUMP  (0/1=NONE/UNIT 16)
C     NRCM           NO. MIXES USED IN GENERATING SHIELDED XS  (IF 0,
C                    DATA IS IN DTF FORMAT)
C     NMIX           TOTAL NUMBER OF MIXES
C     IGE            GEOMETRY  (0/1/2=PLANE/CYLINDER/SPHERE)
C     IM             NUMBER OF SPACE INTERVALS
C     IZM            NUMBER OF MATERIAL ZONES
C     B01            LEFT BOUNDARY CONDITION  (0/1/2=VACUUM/REFL/PER)
C     B02            RIGHT BOUNDARY CONDITION  (0/1/2=VACUUM/REFL/PER)
C     NACT           NUMBER OF COMPUTED ACTIVITY RATES
C     NCG            NUMBER OF CRUNCHED GROUPS
C     NCCM           NO. COLLAPS. DOWNSCAT. TERMS  (IF 0, CALC. BY 1DB)
C     NCCN           NO. COLLAPS. UPSCAT. TERMS  (IF 0, CALC. BY 1DB)
C     NCM            NO. COLLAPSED MATERIALS
C     NIXS           XS INPUT  (0/1/2=CARDS/UNIT 15 BINARY/UNIT 15 BCD)
C     NOXS           XS OUTPUT (0/1/2=PRINT/UNIT 13 BINARY/UNIT 13 BCD)
C     MAXE           MAXIMUM NUMBER OF ERROR FLAGS
C
C     EV             FIRST EIGENVALUE GUESS
C     EVM            EIGENVALUE MODIFIER
C     S03            PARAMETRIC EIGENVALUE
C     BUCK           BUCKLING
C     LAL            LAMBDA LOWER
C     LAH            LAMBDA UPPER
C     EPS            EIGENVALUE CONVERGENCE CRITERION
C     EPSA           PARAMETRIC EIGENVALUE CONVERGENCE CRITERION
C     POD            PARAMETER OSCILLATION DAMPER
C     ORF            FISSION SOURCE OVER-RELAXATION FACTOR
C     S01            NEG/POS=POWER (MWT)/NEUTRON SOURCE RATE
C     TMAX           MAXIMUM TIME (MINUTES--IF 0, NO EFFECT)
C     NCON           0/N/-N=END OF PROBLEM/BURNUP FOR N ISOTOPES/
C                    TAKE TIME STEP
C     DELT           TIME STEP (DAYS)
C
C     * * * * *  SUBSCRIPTED VARIABLES  * * * * *
C
C     FEF(ML)        MEV PER FISSION FOR EACH INPUT ISOTOPE
C     RO(IP)         INITIAL RADII
C     MO(IM)         ZONE NUMBERS
C     M2(IZM)        MATERIAL NUMBERS BY ZONE
C     K7(IGM)        FISSION SPECTRUM  (INPUT)
C     V7(IGM)        NEUTRON VELOCITIES
C     I0(M01)        MIX NUMBER
C     I1(M01)        MATERIAL NUMBER FOR MIX
C     I2(M01)        MATERIAL DENSITY
C     R3(IZM)        RADIAL ZONE NUMBERS  (DELTA CALCULATION ONLY)
C     NO(IM,IGM)     FLUX  (OLD)
C     ATW(ML)        MATERIAL ATOMIC WEIGHT
C     MAT(ML)        MATERIAL ID NUMBER (ID ONLY, NOT USED BY CODE)
C     HOLN(ML,2)     MATERIAL NAME
C     ICHI(ML)       FLAG ON CHI INPUT BY ISOTOPE (0/1=NO/YES)
C     NUT(ML)        SEQUENCE NUMBER ON TAPE FOR DATA IN RUSSIAN FORMAT
C     MPUP(ML)       ISOTOPE NOS. OF MATERIALS IN ORDER READ FROM TAPE
C     NT(ML)         ISOTOPE IS TEMPERATURE DEPENDENT  (0/1=NO/YES)
C     NGB(ML)        GROUP NUMBER F-FACTORS BEGIN AT
C     NGE(ML)        GROUP NUMBER F-FACTORS END AT
C     NPFF(5,ML)     NO. OF SIGO COLUMNS (1/2/3/4/5=FISSION/CAPTURE/
C                    TRANS/ELASTIC/KERMA F-FACTORS)
C     FF(5,IGM,10,10)  INPUT SELF-SHIELDING FACTORS
C                        TYPE/GROUP/NO. OF TEMP/NO. OF SIGMAS
C     SF(5,ML,10)    DISCRETE SIGO VALUES FOR INPUT F-FACTORS
```

```
C     SR(8,IGM)      INFINITE DILUTION CROSS SECTIONS (1/2/3/4/5/6/7/8=
C                    (TRANS/FISS/NU/CAP/INELAS/ELAS/N,2N/KERMA)
C     SM(IGM,IGM)    INELASTIC SCATTERING MATRIX (SM(I,J), I TO J)
C     SE(IGM,IGM)    ELASTIC SCATTERING MATRIX
C     ATEM(MM01)     ISOTOPE TEMPERATURE IN XS MIX VECTOR
C     PSF(MM01)      POTENTIAL SCATTERING FACTOR
C     TEM(10,ML)     DISCRETE TEMPERATURE VALUES FOR INPUT F-FACTORS
C     NXS(NRCM)      NO. OF MIX SPECIFICATIONS IN EACH XS MIX
C     FSS(5,NRCX,IGM)CALCULATED SELF-SHIELDING FACTORS
C                    (1/2/3/4/5=FISSION/CAPTURE/TRANS/ELASTIC/KERMA)
C     SIGO(NRCX,IGM) CALCULATED SIGO VALUES
C     NPN(NCG)       NUMBER OF GROUPS PER CRUNCHED GROUP
C     NFP(NCM)       MATERIAL NUMBERS TO BE CRUNCHED (NEGATIVE FOR CELL)
C     NZN(NCM)       ZONE NUMBERS OF FLUXES USED FOR CRUNCH CALCULATION
C     NV(IGM)        COLLAPSED GROUP NO. FOR EACH FINE GROUP
C     C2(IZM,IGM)    ZONE MACROSCOPIC TRANSPORT CROSS SECTIONS
C     ALPH(IZM,IGM) FINE-GROUP TRANSPORT CROSS SECTION WEIGHTING COEFF.
C     ALPS(IZM,NCG) ALPH(IZM,IGM) SUMMED OVER A COLLAPSED GROUP
C     PHJ(IZM,NCG)   ZONE FLUXES FOR EACH COLLAPSED GROUP
C     PHI(IZM,IGP)   ZONE FLUXES FOR EACH FINE GROUP (AND TOTAL)
C     CO(ITL,MT)     CROSS SECTION ARRAY FOR CURRENT GROUP
C     MACT(NACT)     MATERIAL NUMBER FOR ACTIVITIES (IF = 0, FLUX)
C     LACT(NACT)     XS POSITION NUMBER FOR ACTIVITIES
C     A0(IP)         AREA ELEMENTS
C     R1(IP)         CURRENT RADII
C     R4(IM)         AVERAGE RADII
C     R5(IM)         DELTA-R
C     V0(IM)         VOLUME ELEMENTS
C     N2(IM,IGM)     FLUX  (NEW)
C     F0(IM)         FISSION SOURCE (OLD)
C     F2(IM)         FISSION SOURCE (NEW)
C     I3(M01)        MATERIAL DENSITIES USED IN GRAM
C     K6(IGM)        FISSION SPECTRUM  (EFFECTIVE)
C     S2(IM)         SOURCE
C     CXS(IP,IGM,2) CONSTANTS INVOLVING CROSS SECTIONS FOR FLUX CALC.
C     VOL(IZM)       ZONE VOLUME (LITERS)
C     MASS(ML,IZM)   MATERIAL INVENTORY IN EACH ZONE
C     HA(IM)         TEMP STORAGE FOR FLUX CALCULATION
C     PA(IM)         TEMP STORAGE FOR FLUX CALCULATION
C     EB(8,IGP)      1/2/3/4/5/6/7/8=FISSION SOURCE/IN-SCATTER/
C                    OUT-SCATTER/ABSORPTIONS/LEFT LEAK./RIGHT LEAK./
C                    TOTAL LEAK./FISSION RATE*CONVERSION FACTORS
C     GAM(IZM)       BUCKLING MODIFIERS FOR EACH ZONE
C     ALAM(ML)       DECAY CONSTANT (DAYS-1)
C     MATN(ML)       MATERIAL NUMBER FOR EACH BURNABLE ISOTOPE
C     NBR(ML)        BURNABLE ISOTOPE TYPE (0/1/2)=NONE/FERTILE/FISSILE)
C     NID(ML)        NUMBER DECAY SOURCES FOR EACH BURNABLE ISOTOPE
C     NIN(ML)        NUMBER N,2N SOURCES FOR EACH BURNABLE ISOTOPE
C     NIC(ML)        NUMBER CAPTURE SOURCES FOR EACH BURNABLE ISOTOPE
C     NIF(ML)        NUMBER FISSION SOURCES FOR EACH BURNABLE ISOTOPE
C     IND(ML*ML)     ISOTOPE NUMBER FOR DECAY BY SOURCE
C     INN(ML*ML)     ISOTOPE NUMBER FOR N,2N BY SOURCE
C     INC(ML*ML)     ISOTOPE NUMBER FOR CAPTURE BY SOURCE
C     INF(ML*ML)     ISOTOPE NUMBER FOR FISSION BY SOURCE
C     YDN(ML*ML)     DECAY YIELD BY SOURCE
C     YNN(ML*ML)     N,2N YIELD BY SOURCE
C     YCN(ML*ML)     CAPTURE YIELD BY SOURCE
C     YFN(ML,ML)     FISSION YIELD BY SOURCE
C     FXS(ML,IZM)    SPECTRUM AVERAGE FISSION XS
C     AXS(ML,IZM)    SPECTRUM AVERAGE ABSORPTION XS
C     TXS(ML,IZM)    SPECTRUM AVERAGE N,2N XS
C     EXS(IGM,MT)    ENERGY RELEASE XS FOR EACH MATERIAL
C
$INCLUDE:'1DB.INC'
      COMMON/PACKED/A(50000)
C     READ INPUT DATA
      CALL TIMEX (GLH)
1     CALL INP
      IF (NRCM .LE. 0) GO TO 20
C     CROSS SECTION DATA IS IN RUSSIAN FORMAT
      CALL RCINP1 (A(LHOLN),A(LATW),A(LMAT),A(LICHI),A(LNT),A(LNGB),
     1             A(LNGE),A(LNPFF),A(LFF),A(LSF),A(LSR),A(LSM),
     2             A(LSE),A(LTEM),A(LATEM),A(LI0),A(LI1),
     3             A(LPSF),A(LNUT),A(LMPUP))
      CALL RCSTUP (A(LNPFF),A(LNT),A(LFF),A(LTEM),A(LATEM),A(LNGB),
     1             A(LNGE),A(LI1),A(LSR),A(LSM),A(LSE),A(LMPUP))
```

```
      DO 15  NSIGO = 1,2
      CALL RCCAL1 (A(LNXS),A(LI1),A(LSR),A(LPSF),A(LI2),A(LFSS),
     1             A(LSIGO))
15    CALL RCCAL2 (A(LNXS), A(LI1), A(LNGB), A(LNGE), A(LNPFF),
     1             A(LFF), A(LSIGO), A(LSF), A(LFSS), A(LHOLN))
      R02 = R02 + 1
      NRED = 0
      CALL RCCSS (A(LNXS),A(LI1),A(LSR),A(LSM),A(LSE),A(LN2),A(LFSS))
C     READ DATA IN DTF FORMAT, PERFORM ADJ REVERSALS, AND MAKE XS TAPE
20    CALL RECS (A(LN2), A(LATW), A(LHOLN))
      DO 25 I=LN2,LAST
25    A(I)=0.0
40    CALL INIT (A(LK6), A(LK7), A(LIO), A(LI1), A(LI2), A(LM0), A(LM2),
     1           A(LN0), A(LR0), A(LR1), A(LR3), A(LR4), A(LR5), A(LA0),
     2           A(LF0), A(LC0), A(LV0), A(LV7), A(LFEF), A(LEXS),
     3           A(LGAM), A(LHOLN))
C     PERFORM FISSION CALCULATION
      CALL FISCAL (A(LN0), A(LF0), A(LV0), A(LC0), A(LK6),
     2             A(LM0), A(LM2), A(LEB), A(LEXS))
C     CALL MONITOR PRINT
50    CALL MONPR
      IF (NGOTO .EQ. 1) GO TO 100
C     PERFORM AN OUTER ITERATION
      CALL OUTER (A(LA0), A(LC0), A(LF0), A(LK6), A(LM0), A(LM2),
     1            A(LN0), A(LN2), A(LS2), A(LV0), A(LV7), A(LF2),
     2            A(LCXS), A(LR5), A(LR4), A(LHA), A(LPA), A(LEB),
     3            A(LEXS))
C     PERFORM FISSION CALCULATION
      CALL FISCAL (A(LN2), A(LF0), A(LV0), A(LC0), A(LK6),
     2             A(LM0), A(LM2), A(LEB), A(LEXS))
C     PERFORM CONVERGENCE AND NEW PARAMETER CALCULATIONS
      CALL CNNP (A(LF2),A(LK6),A(LEB))
      IF (NGOTO .EQ. 2) GO TO 50
      IF (NGOTO .EQ. 3) GO TO 40
C     FINAL PRINT
100   CALL FINPR (A(LF2), A(LPA), A(LC0), A(LM0), A(LM2), A(LN2),
     1            A(LR1), A(LR4), A(LV0), A(LF0), A(LEB), A(LEXS),
     2            A(LCXS), A(LPHI), A(LC2), A(LGAM))
      IF (NACT .LE. 0) GO TO 150
      CALL ACTIVE (A(LF2), A(LC0), A(LM0), A(LM2), A(LN2), A(LR4),
     1             A(LMACT), A(LLACT), A(LHOLN), A(LV0), A(LVOL))
C     CALCULATE MATERIAL INVENTORY
150   CALL GRAM (A(LMASS), A(LVOL), A(LATW), A(LHOLN), A(LM0),
     1           A(LM2),A(LV0),A(LIO),A(LI1),A(LI2),A(LI3))
      CALL INPB (A(LMATN),A(LNBR),A(LNID),A(LNIN),A(LNIC),A(LNIF),
     1           A(LIND), A(LINN), A(LINC), A(LINF),
     2           A(LYDN), A(LYNN), A(LYCN), A(LYFN),
     3           A(LALAM), A(LHOLN))
      IF (NCON .EQ. 0) GO TO 200
      CALL AVERAG (A(LPHIB),A(LAXS),A(LFXS),A(LTXS),A(LMATN),A(LMASS),
     1             A(LATW),A(LVOL),A(LC0),A(LN2),A(LM0),A(LV0),A(LHOLN),
     2             A(LNBR))
      CALL MARCH (A(LPHIB),A(LMATN),A(LFXS),A(LAXS),A(LTXS),A(LVOL),
     1            A(LMASS),A(LMASSP),A(LALAM),
     2            A(LNID),A(LNIN),A(LNIC),A(LNIF),
     3            A(LIND),A(LINN),A(LINC),A(LINF),
     4            A(LYDN),A(LYNN),A(LYCN),A(LYFN),
     5            A(LIO), A(LI1), A(LI2), A(LM2))
      GO TO 40
200   IF (NCG .LE.0) GO TO 240
C     COLLAPSE CROSS SECTIONS
      CALL CRUNCH (A(LPHI),A(LV7),A(LPHJ),A(LNPN),A(LK7),
     1             A(LN2),A(LC0),A(LNFP),A(LNZN),A(LHOLN),
     2             A(LATW),A(LNV),A(LVOL),A(LC2),A(LALPH),A(LALPS))
240   STOP 'END 1DB'
      END

      SUBROUTINE INP
$INCLUDE:'1DB.INC'
      COMMON/PACKED/A(50000)
C     THIS SUBROUTINE CONTROLS THE READING OF INPUT DATA EXCEPT XS DATA
      MEMORY = 50000
      NCR1 = 3
      NSCRAT = 4
      NINP = 5
      NOUT = 6
```

```
      NFF = 8
      NSR = 9
      NSM = 10
      NSCR1 = 11
      NSCR2 = 12
C     UNIT 5=INPUT
C     UNIT 6=OUTPUT
C     UNIT 13=CROSS SECTION OUTPUT
C     UNIT 14=FLUX INPUT
C     UNIT 15=CROSS SECTION INPUT
C     UNIT 16=FLUX OUTPUT
      OPEN (5,  FILE=' ')
      OPEN (6,  FILE=' ')
      OPEN (15, FILE=' ', STATUS='UNKNOWN')
      OPEN (13, FILE=' ', STATUS='UNKNOWN')
C     OPEN (14, FILE=' ', FORM='UNFORMATTED')
C     OPEN (16, FILE=' ', FORM='UNFORMATTED')
      OPEN (4,  STATUS='SCRATCH', FORM='UNFORMATTED')
      OPEN (8,  STATUS='SCRATCH', FORM='UNFORMATTED')
      OPEN (9,  STATUS='SCRATCH', FORM='UNFORMATTED')
      OPEN (10, STATUS='SCRATCH', FORM='UNFORMATTED')
      OPEN (11, STATUS='SCRATCH', FORM='UNFORMATTED')
      OPEN (12, STATUS='SCRATCH', FORM='UNFORMATTED')
      OPEN (3,  STATUS='SCRATCH', FORM='UNFORMATTED')
      REWIND NCR1
      REWIND NSCRAT
      REWIND NFF
      REWIND NSR
      REWIND NSM
      REWIND NSCR1
      REWIND NSCR2
      WRITE(NOUT,10)
10    FORMAT (12X,'*  *  *  1 D B  (PC VERSION 7/24/90)  *  *  *'//)
      READ(NINP,20,END=25) (ID(I),I=1,20)
20    FORMAT (20A4)
      GO TO 26
25    STOP
26    CONTINUE
      WRITE(NOUT,30) (ID(I),I=1,20)
30    FORMAT (1X,20A4/)
      CALL REIC2 (A02, NEV, S02, IGM, NXCM, NXCN, NST, ML, NFX, NPRT,
     1            NPUN, NOIT, NRCM, NMIX, IGE, IM, IZM, B01, B02,
     2            NACT, NCG, NCCM, NCCN, NCM, NOXS, NIXS, MAXE)
      CALL REGC2 (EV, EVM, S03, BUCK, LAL, LAH, EPS, EPSA, POD, ORF,
     1            S01, TMAX)
      WRITE (NOUT,60) A02, NEV, S02, IGM, NXCM, NXCN, NST
60    FORMAT(
     1 I6,'  A02   0/1=REGULAR CALCULATION/ADJOINT CALCULATION'/
     2 I6,'  NEV   EIGENVALUE TYPE (1/2/3/4/5=K/ALPHA/CONC/DELTA/BUCK)'/
     3 I6,'  S02   PARAMETRIC EIGENVALUE TYPE (0/1/2=NONE/K/ALPHA)'/
     4 I6,'  IGM   NUMBER OF ENERGY GROUPS'/
     5 I6,' NXCM   NUMBER OF DOWNSCATTERING TERMS'/
     6 I6,' NXCN   NUMBER OF UPSCATTERING TERMS'/
     7 I6,'  NST   NUMBER OF ADDED SLOTS (BEFORE N,2N)')
      WRITE (NOUT,70) ML, NRCM, NMIX
70    FORMAT(
     9 I6,'   ML   NUMBER OF INPUT MATERIALS'/
     5 I6,' NRCM   NO. OF MIXES USED IN GENERATING SF XS (0 FOR DTF)'/
     6 I6,' NMIX   TOTAL NUMBER OF MIXES'/)
      WRITE (NOUT,80) IGE, IM, IZM, B01, B02, NPRT, NOIT, MAXE
80    FORMAT(
     1 I6,'  IGE   GEOMETRY (0/1/2=PLANE/CYLINDER/SPHERE)'/
     2 I6,'   IM   NUMBER OF SPACE INTERVALS'/
     3 I6,'  IZM   NUMBER OF MATERIAL ZONES'/
     5 I6,'  B01   LEFT BOUNDARY  (0/1/2=VACUUM/REFL./PERIODIC)'/
     1 I6,'  B02   RIGHT BOUNDARY (0/1/2=VACUUM/REFL./PERIODIC)'/
     1 I6,' NPRT   PRINT OPTION (0/1/2/3=MINI/FLUXES/XS/ALL)'/
     4 I6,' NOIT   MAXIMUM NUMBER OF OUTER ITERATIONS'/
     7 I6,' MAXE   MAXIMUM NUMBER OF ERROR FLAGS'/)
      WRITE (NOUT,90) NACT, NCG, NCCM, NCCN, NCM,
     1                NFX, NPUN, NIXS, NOXS
90    FORMAT(
     1 I6,' NACT   NUMBER OF ACTIVITIES'/
     2 I6,'  NCG   NUMBER OF COLLAPSED GROUPS (IF 0, NO EFFECT)'/
     3 I6,' NCCM   NO. OF COLLAPSED DOWNSCATTER TERMS (IF <0, BY 1DB)'/
     4 I6,' NCCN   NO. OF COLLAPSED UPSCATTER TERMS (IF <0, BY 1DB)'/
```

```
     5 I6,'  NCM   NUMBER OF COLLAPSED MATERIALS'/
     1 I6,'  NFX   FLUX GUESS (0/1=NONE/UNIT 14)'/
     3 I6,' NPUN   FLUX DUMP (0/1=NONE/UNIT 16)'/
     6 I6,' NIXS   INPUT XS  (0/1/2=CARDS/UNIT 15 BINARY/UNIT 15 BCD)'/
     6 I6,' NOXS   OUTPUT XS (0/1/2=PRINT/UNIT 13 BINARY/UNIT 13 BCD)'/)
      WRITE (NOUT,100) EV, EVM, S03, BUCK, LAL, LAH
100   FORMAT(
     1 1X,1PE11.4,'   EV   FIRST EIGENVALUE GUESS'/
     2 1X,1PE11.4,'  EVM   EIGENVALUE MODIFIER'/
     3 1X,1PE11.4,'  S03   PARAMETRIC EIGENVALUE'/
     4 1X,1PE11.4,' BUCK   BUCKLING (CM-2)'/
     5 1X,1PE11.4,'  LAL   LAMBDA LOWER'/
     6 1X,1PE11.4,'  LAH   LAMBDA UPPER'/)
      WRITE (NOUT,120) EPS, EPSA, POD, ORF, S01, TMAX
120   FORMAT(
     1 1X,1PE11.4,'  EPS   EIGENVALUE CONVERGENCE CRITERION'/
     2 1X,1PE11.4,' EPSA   PARAMETER CONVERGENCE CRITERION'/
     3 1X,1PE11.4,'  POD   PARAMETER OSCILLATION DAMPER'/
     4 1X,1PE11.4,'  ORF   OVER-RELAXATION FACTOR'/
     5 1X,1PE11.4,'  S01   NEG./POS.=POWER (MWT)/NEUTRON SOURCE RATE'/
     6 1X,1PE11.4,' TMAX   MAXIMUM TIME (MINUTES--IF 0, NO EFFECT)'/)
      KPAGE = 1000
      ME = 0
      IHT = 5 + NST
      IHS = IHT + NXCN + 1
      ITL = NXCM + IHS
      IP = IM + 1
      IGP = IGM + 1
      IGEP = IGE + 1
      EQ = .0
      LAP = .0
      LAPP = .0
      DAY = .0
      LAR = .0
      ALA = .0
      P02 = 0
      CVT = 0
      CNT = 0
      NXCMP = NXCM
      NXCNP = NXCN
      IF (NXCMP .GT. (NCG-1)) NXCMP = NCG-1
      IF (NXCNP .GT. (NCG-1)) NXCNP = NCG-1
      ITLP = NXCMP + IHS
      T06 = 0
      IF (NEV .EQ. 4) T06 = 1
      R02 = 0
      IRED = 1
      NRED = 0
      MM01 = 0
      NRCX = 0
      IF (NRCM .LE. 0) GO TO 316
      NRED = 1
      R02 = -1
      NRCX = ML
      MM01 = NRCM + NRCX
316   NXCMR = NXCM + 1
C     ESTIMATE OF M01 AND MM01
      M01 = NMIX * (ML + 1)
      MT = ML + NMIX
C     COMPUTE DIMENSION POINTERS
      LR0 = 1
      LM0 = LR0 + IP
      LM2 = LM0 + IM
      LGAM = LM2 + IZM
      LK7 = LGAM + IZM
      LV7 = LK7 + IGM
      LI0 = LV7 + IGM
      LI1 = LI0 + M01
      LI2 = LI1 + M01
      LR3 = LI2 + M01
      LN0 = LR3 + IZM*T06
      LATW = LN0 + IM*IGM
      LMAT = LATW + ML
      LICHI = LMAT + ML
      LHOLN = LICHI + ML
      LNUT = LHOLN + ML * 2
```

```
      LMPUP = LNUT + ML
      LNT = LMPUP + ML
      LNGB = LNT + ML*NRED
      LNGE = LNGB + ML*NRED
      LNPFF = LNGE + ML*NRED
      LNDSN = LNPFF + 5*ML*NRED
      LFF = LNDSN + ML*NRED
      LSF = LFF + 5*IGM*100*NRED
      LSR = LSF + 5*ML*10*NRED
      LSM = LSR + 8*IGM*NRED
      LSE = LSM + IGM*IGM*NRED
      LATEM = LSE + IGM*IGM*NRED
      LPSF = LATEM + MM01*NRED
      LTEM = LPSF + MM01*NRED
      LNXS = LTEM + 10*ML*NRED
      LFSS = LNXS + NRCM*NRED
      LSIGO = LFSS + 5*NRCX*IGM*NRED
      LNPN = LSIGO + NRCX*IGM*NRED
      LNFP = LNPN + NCG
      LNZN = LNFP + NCM
      LNV = LNZN + NCM
      LC2 = LNV + IGM
      LALPH = LC2 + IZM*IGM
      LALPS = LALPH + IZM*IGM
      LPHJ = LALPS + IZM*NCG
      LPHI = LPHJ + IZM*NCG
      LCO = LPHI + IZM*IGP
      LMACT = LCO + ITL*MT
      LLACT = LMACT + NACT
      LAO = LLACT + NACT
      LR1 = LAO + IP
      LR4 = LR1 + IP
      LR5 = LR4 + IM
      LVO = LR5 + IM
      LFEF = LVO + IM
      LEXS = LFEF + ML
      LMATN = LEXS + MT * IGM
      LNBR = LMATN + ML
      LNID = LNBR + ML
      LNIN = LNID + ML
      LNIC = LNIN + ML
      LNIF = LNIC + ML
      LIND = LNIF + ML
      LINN = LIND + ML*ML
      LINC = LINN + ML*ML
      LINF = LINC + ML*ML
      LYDN = LINF + ML*ML
      LYCN = LYDN + ML*ML
      LYNN = LYCN + ML*ML
      LYFN = LYNN + ML*ML
      LIDUM = LYFN + ML*ML
      LALAM = LIDUM + ML + 5
      LPHIB = LALAM + ML
      LAXS = LPHIB + IZM
      LFXS = LAXS + ML*IZM
      LTXS = LFXS + ML*IZM
      LMASSP = LTXS + ML*IZM
C     CROSS SECTIONS ALSO BEGIN AT THIS ADDRESS
      LN2 = LMASSP + ML*IZM
      LFO = LN2 + IM*IGM
      LF2 = LFO + IM
      LI3 = LF2 + IM
      LK6 = LI3 + M01
      LS2 = LK6 + IGM
      LCXS = LS2 + IM
      LVOL = LCXS + IP*IGM*2
      LMASS = LVOL + IZM
      LHA = LMASS + ML*IZM
      LPA = LHA + IM
      LEB = LPA + IM
      ITEMP = LEB + 8*IGP
      LAST = LN2 + ITL*IGM*MT
      IF (LAST .LT. ITEMP) LAST = ITEMP
      WRITE (NOUT, 320) LAST
320   FORMAT (' LAST = ', I8)
321   IF (LAST .GT. MEMORY) CALL ERRO2 ('   INP', 321,1)
```

```
      CALL CLEAR (.0, A(1), LAST)
      WRITE (NOUT,350)
350   FORMAT ()
C     READ FLUXES
      CALL FXINP (A(LNO))
C     READ ENERGY PER FISSION
      WRITE (NOUT, 360)
360   FORMAT (/' FISSION ENERGY (MEV/FISSION)')
      CALL REAG2 ('   FEF', A(LFEF), ML)
C     READ MESH POINTS
      WRITE (NOUT,380)
380   FORMAT (/' MESH POINTS')
      CALL REAG2 ('    R0',A(LR0),IP)
C     READ ZONE NUMBERS
      WRITE (NOUT,390)
390   FORMAT (/' ZONE NUMBERS BY MESH INTERVAL')
      CALL REAI2 ('    M0',A(LM0),IM)
C     READ MATERIAL NUMBERS
      WRITE (NOUT,400)
400   FORMAT (/' MATERIAL NUMBERS BY ZONE')
      CALL REAI2 ('    M2',A(LM2),IZM)
C     READ BUCKLING MODIFIERS
      IF (NEV .EQ. 5) GO TO 402
      IF (BUCK .EQ. .0) GO TO 408
402   WRITE (NOUT,404)
404   FORMAT (/' BUCKLING MODIFIERS BY ZONE')
      CALL REAG2 ('   GAM',A(LGAM),IZM)
C     READ FISSION FRACTIONS
408   WRITE (NOUT,410)
410   FORMAT (/' FISSION FRACTIONS')
      CALL REAG2 ('    K7',A(LK7),IGM)
C     READ VELOCITIES
      IF (NEV .EQ. 2 .OR. S02 .EQ. 2) GO TO 415
      GO TO 425
415   WRITE (NOUT,420)
420   FORMAT (/' NEUTRON VELOCITY')
      CALL REAG2 ('    V7',A(LV7),IGM)
C     READ MIXTURE SPECIFICATIONS
425   M01 = 0
      IF (NMIX .LE. 0) GO TO 440
      WRITE (NOUT,430)
430   FORMAT (/' MIXTURE SPECIFICATIONS FOR EACH MIX')
      MM01 = 0
      MM0 = LI0
      MM1 = LI1
      MM2 = LI2
      MT1 = MT - NMIX + 1
      DO 435 M=MT1,MT
      CALL REAIG2 ('   MIX', A(MM0), A(MM1), A(MM2),
     1             A(MM0+1), A(MM1+1), A(MM2+1),  M, NN)
      MM0 = MM0 + NN + 1
      MM1 = MM1 + NN + 1
      MM2 = MM2 + NN + 1
      N = M - ML
      IF (N .GT. NRCM) GO TO 435
      CALL SET (A(LNXS+N-1), NN+1)
      MM01 = MM01 + NN + 1
435   M01 = M01 + NN + 1
      NRCX = MM01 - RNCM
436   IF (M01 .GT. NMIX*(ML+1))  CALL ERR02 ('   INP',436,1)
437   IF (MM01 .GT. (NRCM+ML))   CALL ERR02 ('   INP',437,1)
C     READ ZONE MODIFIERS IF DELTA CALCULATION
440   IF (NEV .NE. 4) GO TO 500
      WRITE (NOUT,460)
460   FORMAT (/' ZONE MODIFIERS FOR DELTA OPTION')
      CALL REAG2 ('    R3',A(LR3),IZM)
C     READ ACTIVITATION MATERIALS AND CROSS SECTION POSITIONS
500   IF (NACT .LE. 0) GO TO 530
      WRITE (NOUT,520)
520   FORMAT (/' MATERIAL NUMBERS FOR ACTIVITIES')
      CALL REAI2 ('  MACT', A(LMACT), NACT)
      WRITE (NOUT,525)
525   FORMAT (/' CROSS SECTION POSITION FOR ACTIVITIES')
      CALL REAI2 ('  LACT', A(LLACT), NACT)
C     READ NUMBER OF GROUPS FOR EACH CRUNCHED GROUP
530   IF (NCG .LE. 0) GO TO 575
```

```
      WRITE (NOUT,550)
550   FORMAT (/' NUMBER OF GROUPS PER CRUNCHED GROUP')
      CALL REAI2 ('   NPN',A(LNPN),NCG)
C     READ NUMBERS OF MATERIALS TO BE CRUNCHED
      WRITE (NOUT,560)
560   FORMAT (/' NUMBERS OF MATERIALS TO BE CRUNCHED (NEG. FOR CELL)')
      CALL REAI2 ('   NFP',A(LNFP),NCM)
C     READ ZONE NUMBERS OF FLUXES USED FOR CRUNCH CALCULATION
      WRITE (NOUT,570)
570   FORMAT (/' ZONE NUMBERS OF FLUXES USED FOR CRUNCH CALCULATION')
      CALL REAI2 ('   NZN',A(LNZN),NCM)
575   IF (NRCM .LE. 0) GO TO 670
C     READ TEMPERATURES AND POT. SCAT. FACTOR FOR EACH TERM IN XS MIXES
      NNN = 1
      DO 625 N=1,NRCM
      CALL SET (NN, A(LNXS+N-1))
      WRITE (NOUT, 620) N
620   FORMAT (/' TEMP. AND POT. SCAT. FACTOR FOR EACH TERM IN MIX ', I6)
      CALL REAG2 ('  ATEM', A(LATEM+NNN), NN-1)
      CALL REAG2 ('   PSF', A(LPSF+NNN), NN-1)
625   NNN = NNN + NN
C     READ SEQUENCE NUMBERS ON TAPE FOR INPUT CROSS SECTIONS
      WRITE (NOUT,660)
660   FORMAT(/' SEQUENCE NUMBERS ON TAPE FOR XS DATA IN THE RUSSIAN',
     1       ' FORMAT')
      CALL REAI2 ('   NUT', A(LNUT), ML)
C     END OF INPUT DATA  (EXCEPT CROSS SECTION DATA)
670   CONTINUE
      RETURN
      END

      SUBROUTINE SET (NUM, N)
      NUM = N
      RETURN
      END

      SUBROUTINE CLEAR (X,Y,N)
      DIMENSION Y(1)
      DO 10 I=1,N
10    Y(I)=X
      RETURN
      END

      SUBROUTINE TIMEX (SEC)
      INTEGER*2 I1, I2, I3, I4
      CALL GETTIM (I1,I2,I3,I4)
      SEC = FLOAT (3600*I1 + 60*I2 + I3) + .01*FLOAT(I4)
      RETURN
      END

      SUBROUTINE CNNP (F2,K6,EB)
      DIMENSION F2(1), K6(1), EB(8,1)
$INCLUDE:'1DB.INC'
C     CONVERGENCE TESTS
25    E01=1.0-ALA
      E02=ABS(E01)
50    IF(EB(1,IGP))  55,130,55
55    IF (E02 - EPS)  60, 60, 70
60    CVT=1
70    CALL CLEAR (0.0, F2, IM)
      GO TO 105
80    EV=EV+POD*EQ*E01
      GO TO 170
C     FINAL PRINT
90    NGOTO=1
      IF (NEV .GT. 1) GO TO 80
      EV=0.0
      DO 100 I=1,IGM
100   EV=EV+K6(I)
      EV=SK7/EV
      GO TO 135
105   IF(CVT-1) 110, 90, 110
110   IF(NEV-1) 115, 120, 140
C     MONITOR PRINT
115   NGOTO=2
      GO TO 135
```

```
120   EV=0.
      DO 125 I=1,IGM
125   EV=EV+K6(I)
      EV=SK7/EV
      GO TO 115
130   CALL ERRO2('  CNNP',130,1)
135   GO TO 300
140   CONTINUE
C     CALCULATE NEW PARAMETERS FOR SEARCH CALCULATIONS
145   E03=ABS (ALA-LAR)
      IF (LAPP)  270, 150, 270
150   IF (LAP)  230, 155, 230
155   IF (EQ) 200, 160, 200
160   IF (E03-EPSA) 175, 175, 165
C     MONITOR PRINT.
165   NGOTO=2
      GO TO 300
C     FINAL PRINT EXIT.
170   NGOTO=1
      GO TO 300
175   LAP=ALA
      EVP=EV
      IF (E01) 185,185,180
180   EV=EV-EVM
      GO TO 190
185   EV=EV+EVM
190   IF (NEV .EQ. 2) GO TO 165
C     MIX X-SECS.
      NGOTO=3
      GO TO 300
200   IF (CVT) 170, 205,170
205   EV=EV+POD*EQ*E01
210   IF ((LAPP-1.0)/(LAP-1.0))  215, 190, 190
215   TEMP1=AMIN1(EVP,EVPP)
      IF (EV-TEMP1)  220, 225, 225
220   EV=(EVPP+EVP)/2.
      GO TO 190
225   TEMP1=AMAX1(EVP,EVPP)
      IF (EV-TEMP1)  190, 220, 220
230   IF (E03-EPSA)  235, 235, 165
235   EQ=(EVP-EV)/(LAP-ALA)
240   IF (CNT)  260, 245, 260
245   IF (E02-LAL)  265, 265, 250
250   IF (E02-LAH)  260, 260, 255
255   E01=SIGN (LAH,E01)
260   LAPP=LAP
      LAP=ALA
      EVPP=EVP
      EVP=EV
      GO TO 205
265   CNT=1
      LAP=0.0
      LAPP=0.0
      GO TO 205
270   IF (E03-EPSA) 275, 275, 165
C     CALCULATE QUADRATIC COEFFICIENTS.
275   TEMP1=EVP-EV
      TEMP2=EVPP-EV
      TEMP3=EVPP-EVP
      TEMP4=TEMP1*(EVP+EV)
      TEMP5=-TEMP2*(EV+EVPP)
      TEMP6=TEMP3*(EVPP+EVP)
      DENOM=TEMP3*TEMP2*TEMP1
      EQA=((LAPP-1.0)*TEMP1*EVP*EV-(LAP-1.0)*TEMP2
     1*EV*EVPP+(ALA-1.0)*TEMP3*EVPP*EVP)/DENOM
      EQB=-(LAPP*TEMP4+LAP*TEMP5+ALA*TEMP6)/DENOM
      EQC=(LAPP*TEMP1-LAP*TEMP2+ALA*TEMP3)/DENOM
      DISCR=EQB*EQB-4.0*EQA*EQC
      IF (DISCR)  235, 280, 280
280   IF (E02-LAL)  265, 265, 285
285   TEMP1=EQC+EQC
      TEMP=SQRT (DISCR)
      EQ=1.0/(EQB+EV*TEMP1)
      LAPP=LAP
      LAP=ALA
      EVPP=EVP
```

```
      EVP=EV
      EV1=(TEMP-EQB)/TEMP1
      EV2=-(TEMP+EQB)/TEMP1
      EVA=ABS (EV-EV1)
      EVB=ABS (EV-EV2)
      IF (EVA-EVB)  290, 290, 295
290   EV=EV1
      GO TO 210
295   EV=EV2
      GO TO 210
300   CONTINUE
      RETURN
      END

      SUBROUTINE CRUNCH (PHI,K8,PHJ,NPN,K7,C1,CO,NFP,NZN,HOLN,
     1                   ATW, NV, VOL, C2, ALPH, ALPS)
      DIMENSION PHI(IZM,1), K8(1), PHJ(IZM,1), NPN(1),
     1          K7(1), CO(ITL,1), NFP(1), NZN(1),
     2          ATW(1), NV(1), VOL(1), C2(IZM,1), ALPH(IZM,1),
     3          ALPS(IZM,1), HOLN(ML,1)
      DIMENSION C1(ITLP,NCG,1)
      REAL      K8
C     C1(ITLP,NCG,MT) COLLAPSED CROSS SECTION ARRAY--STARTS AT A(LN2)
C     K8(NCG) COLLAPSED FISSION SPECTRUM--STARTS AT A(LV7)
$INCLUDE:'1DB.INC'
C     CALCULATE ZONE AVERAGED FLUXES FOR EACH FINE GROUP
      T1 = .0
      DO 30  KZ=1,IZM
      T1 = T1 + VOL(KZ)
      PHI(KZ,IGP) = 0.0
      DO 30  IIG=1,IGM
      IF (VOL(KZ) .EQ. 0.) GO TO 30
      PHI(KZ,IIG) = PHI(KZ,IIG)*.001/VOL(KZ)
30    PHI(KZ,IGP) = PHI(KZ,IGP) + PHI(KZ,IIG)
C     CALC. FISS. SPECTRUM AND ZONE AV. FLUXES FOR EACH COLLAPSED GROUP
      CALL CLEAR (.0, K8(1), NCG)
      ITEMP = NCG*IZM
      CALL CLEAR (.0, PHJ(1,1), ITEMP)
      IIG = 0
      DO 100  JJG=1,NCG
      ITEMP = NPN(JJG)
      DO 100  K=1,ITEMP
      IIG = IIG + 1
      K8(JJG) = K8(JJG) + K7(IIG)
      NV(IIG) = JJG
      DO 100  KZ=1,IZM
100   PHJ(KZ,JJG) = PHJ(KZ,JJG) + PHI(KZ,IIG)
      WRITE (NOUT,110)  (KZ,PHI(KZ,IGP), VOL(KZ),KZ=1,IZM)
110   FORMAT (///' ZONE AVERAGED FLUXES'//
     1        ' ZONE          FLUX             VOLUME  '/
     2        '            (N/CM2-SEC)        (LITERS) '//(I4,1P2E18.5))
115   WRITE (NOUT,120)
120   FORMAT (///' COLLAPSED FISSION FRACTIONS AND ZONE AVERAGED',
     1           ' FLUXES BY GROUP'/)
      DO 180  KZ=1,IZM,5
      ITEMP = KZ + 4
      IF (ITEMP - IZM)  140,140,130
130   ITEMP = IZM
140   WRITE (NOUT,150)  ((K), K=KZ,ITEMP)
150   FORMAT (' GROUP  FISS. FCT.', 5(5X, 'ZONE', I3,:))
      WRITE (NOUT,160)
160   FORMAT ()
170   FORMAT (I6, 1P6E12.3)
      DO 180 JJG=1,NCG
180   WRITE (NOUT,170) JJG,K8(JJG),(PHJ(K,JJG), K=KZ,ITEMP)
C     NO GROUP COLLAPSING
      IF (IGM .NE. NCG) GO TO 225
      IF (NCCN .GT. 0) GO TO 225
      IF (NCCM .GT. 0) GO TO 225
      DO 185 M=1,NCM
185   IF (NFP(M) .LT. 0) GO TO 225
      DO 190 JJG = 1,IGM
      READ (NCR1) ((CO(I,J), I=1,ITL), J=1,MT)
      DO 190 M=1,NCM
      ITEMP1 = ABS(NFP(M))
      DO 190 JT = 1,ITLP
```

```
      C1(JT,JJG,M) = CO(JT,ITEMP1)
  190 CONTINUE
      REWIND NCR1
      GO TO 955
C     CALCULATION OF COLLAPSED SIGF, SIGA, NUSIGF, AND SIGTR
  225 ITEMP = NCG*IZM
      CALL CLEAR (.0, ALPS(1,1), ITEMP)
      IIG = 0
      DO 235  JJG = 1, NCG
      ITEMP = NPN(JJG)
      DO 235  K=1,ITEMP
      IIG = IIG + 1
      DO 235  KZ=1,IZM
      ALPH(KZ,IIG) = PHI(KZ,IIG)/C2(KZ,IIG)
  235 ALPS(KZ,JJG) = ALPS(KZ,JJG) + ALPH(KZ,IIG)
      IIG = 0
      DO 290  JJG=1,NCG
      DO 240  JT=1,ITLP
      DO 240  M=1,NCM
  240 C1(JT,JJG,M) = 0.0
      ITEMP = NPN(JJG)
      DO 290  K=1,ITEMP
      IIG = IIG + 1
      READ (NCR1) ((CO(I,J),I=1,ITL),J=1,MT)
      DO 290  M=1,NCM
      ITEMP1 = ABS(NFP(M))
      KZ = NZN(M)
      T2 = PHJ(KZ,JJG)
      IF (NFP(M) .GT. 0) GO TO 275
      T2 = .0
      DO 250 N=1,IZM
  250 T2 = T2 + PHJ(N,JJG) * VOL(N)
      T2 = T2/T1
  275 DO 290 JT = 1,IHT
      IF ((JT .EQ. IHT) .AND. (NFP(M) .GT. 0)) GO TO 285
      C1(JT,JJG,M) = C1(JT,JJG,M) + CO(JT,ITEMP1)*PHI(KZ,IIG)/T2
      GO TO 290
  285 C1(IHT,JJG,M) = C1(IHT,JJG,M) + CO(IHT,ITEMP1)*
     1                ALPH(KZ,IIG)/ALPS(KZ,JJG)
  290 CONTINUE
  310 REWIND NCR1
C     CALCULATION OF SCATTERING MATRIX
      IF ((NXCMP+NXCNP) .LE. 0) GO TO 700
      IIG = 0
      DO 500  JJG=1,NCG
      ITEMP = NPN(JJG)
      DO 500  K=1,ITEMP
      READ (NCR1) ((CO(I,J),I=1,ITL),J=1,MT)
      IIG = IIG + 1
      DO 500  M=1,NCM
      ITEMP1 = ABS(NFP(M))
      KZ = NZN(M)
      L1 = -NXCN
      L2 = NXCM
      DO 500  LT=L1,L2,1
      IT = LT + IHS
      ING = IIG - LT
      JNG = NV(ING)
      JT = JJG + IHS - JNG
      T2 = PHJ(KZ,JNG)
      IF (NFP(M) .GT. 0) GO TO 400
      T2 = .0
      DO 350 N=1,IZM
  350 T2 = T2 + PHJ(N,JNG) * VOL(N)
      T2 = T2/T1
      IF (JT .EQ. IHS)  GO TO 500
      IF (ING .LE. 0)   GO TO 500
      IF (ING .GT. IGM) GO TO 500
  400 C1(JT,JJG,M) = C1(JT,JJG,M) + CO(IT,ITEMP1)*PHI(KZ,ING)/T2
  500 CONTINUE
      IF (NCCM .LT. 0) GO TO 600
      IF (NXCMP .LE. NCCM) GO TO 600
      ITEMP = NCCM + IHS
      ITEMP1 = ITEMP + 1
      ITEMP2 = ITLP
      ITLP = NCCM + IHS
```

```
      NXCMP = NCCM
      DO 570  M=1,NCM
      DO 570  JJG=1,NCG
      DO 570  JT=ITEMP1,ITEMP2
      KKG = JJG + JT + 1 - ITEMP1
      IF (KKG .GT. NCG) GO TO 570
      C1(ITEMP,JJG,M) = C1(ITEMP,JJG,M) + C1(JT,KKG,M)
570   CONTINUE
600   CONTINUE
      IF (NCCN .LT. 0) GO TO 700
      IF (NXCNP .LE. NCCN) GO TO 700
      ITEMP = IHS - NCCN
      ITEMP1 = ITEMP - 1
      ITEMP2 = IHS - NXCNP
      DO 670 M=1,NCM
      DO 670 JJG = 1,NCG
      DO 670 JT = ITEMP1, ITEMP2, -1
      KKG = JJG + JT -1 - ITEMP1
      IF (KKG .LE. 0) GO TO 670
      C1(ITEMP,JJG,M) = C1(ITEMP,JJG,M) + C1(JT,KKG,M)
670   CONTINUE
700   CONTINUE
      IF (NCCN .LT. 0)  GO TO 710
      IF (NXCNP .GT. NCCN) NXCNP = NCCN
710   ITEMP = NXCN - NXCNP
      IF (ITEMP .LE. 0) GO TO 900
      ITLP = ITLP - ITEMP
      DO 800 M=1,NCM
      DO 800 JJG = 1,NCG
      DO 800 JT = IHT+1,ITLP
800   C1(JT,JJG,M) = C1(JT+ITEMP,JJG,M)
900   CONTINUE
C     CALCULATION OF SIGG
      ITEMP = IHS - ITEMP
      DO 950  JJG=1,NCG
      DO 950  M=1,NCM
      TEMP = .0
      L1 = -NXCNP
      L2 = NXCMP
      DO 940 L=L1,L2,1
      IF (L .EQ. 0) GO TO 940
      JT = L + ITEMP
      J = JJG + L
      IF (J .LE. 0) GO TO 940
      IF (J .GT. NCG) GO TO 940
      TEMP = TEMP + C1(JT,J,M)
940   CONTINUE
950   C1(ITEMP,JJG,M) = C1(IHT,JJG,M) - C1(IHT-2,JJG,M) - TEMP
955   WRITE (NOUT,960)
960   FORMAT (//' COLLAPSED CROSS SECTIONS')
      DO 995 JJG=1,NCG
      WRITE (NOUT,970) JJG
970   FORMAT (/' GROUP', I3/
     1         ' MATERIAL  ZONE   N,2N     SIGF     SIGA   ',
     1         ' NUSIGF   SIGTR... GXG...'/)
      ITEMP = NST + 1
      DO 995  M=1,NCM
      N = ABS(NFP(M))
      KZ = NZN(M)
      IF (N .GT. ML) GO TO 980
      WRITE (NOUT, 975) (HOLN(N,L), L=1,2), KZ,
     1                  (C1(JT,JJG,M),JT=ITEMP,ITLP)
975   FORMAT (1X, 2A4, 1X, I3, 1P7E9.2, (/13X, 1P7E9.2))
      GO TO 995
980   WRITE (NOUT,990) N, KZ, (C1(JT,JJG,M),JT=ITEMP,ITLP)
990   FORMAT (' MAT# ', I3, 1X, I3, 1P7E9.2, (/13X, 1P7E9.2))
995   CONTINUE
      IF (NOXS .LE. 0) GO TO 1100
      WRITE (NOUT,1010)
1010  FORMAT (//' THE FOLLOWING MATERIALS WERE WRITTEN TO UNIT 13'/)
      DO 1070   M=1,NCM
      N = ABS(NFP(M))
      IF (N .LE. ML) GO TO 1028
      T1 = .0
1012  WRITE (NOUT, 1013)
1013  FORMAT (' MACROSCOPIC MIX')
```

```
      IF (NOXS .EQ. 2) GO TO 1020
      WRITE (13) (' MIX')
      WRITE (13) T1
      GO TO 1050
1020  WRITE (13, 1024) (' MIX')
1024  FORMAT (2A4)
1028  WRITE (NOUT,1030) N, (HOLN(N,L),L=1,2), ATW(N)
1030  FORMAT (I3, 4X, 2A4, F12.2)
      IF (NOXS .EQ. 2) GO TO 1040
      WRITE (13) (HOLN(N,L),L=1,2)
      WRITE (13) ATW(N)
      GO TO 1050
1040  WRITE (13, 1024) (HOLN(N,L), L=1,2)
1050  DO 1070 JJG=1,NCG
      IF (NOXS .EQ. 2) GO TO 1060
      WRITE (13) (C1(JT,JJG,M), JT=1,ITLP)
      GO TO 1070
1060  WRITE (13,1065) (C1(JT,JJG,M), JT=1,ITLP)
1065  FORMAT (6E12.4)
1070  CONTINUE
      REWIND 13
1100  REWIND NCR1
      RETURN
      END

      SUBROUTINE ERRO2 (HOL,JSUBR,I)
      COMMON      NCR1, NFF, NINP, NOUT, NSCRAT, NSCR1, NSCR2
      CHARACTER*6 HOL
      WRITE (NOUT,1)  HOL,JSUBR
1     FORMAT (//' * * *  ERROR IN ',A6,' AT ', I6, ' * * *'//)
      IF (I .EQ. 1) STOP
      RETURN
      END

      SUBROUTINE FINPR (F2,PA,CO,MO,M2,N2,R1,R4,VO,FO,EB,EXS,
     1                  CXS, PHI, C2, GAM)
      DIMENSION  F2(1), PA(1), CO(ITL,1), MO(1), M2(1), N2(IM,1),
     1           R1(1), R4(1), VO(1), FO(1), EB(8,1), EXS(IGM,1),
     2           CXS(IP,IGM,1), PHI(IZM,1), C2(IZM,1), GAM(1)
$INCLUDE:'1DB.INC'
      CHARACTER*4 EOFS
      DATA  EOFS/'eofs'/
C     FINAL PRINT
      CALL MONPR
C     COMPUTE ZONE FLUXES
      DO 10  IIG=1,IGM
      DO 5  KZ=1,IZM
5     PHI(KZ,IIG) = 0.0
      DO 10  I=1,IM
      KZ = MO(I)
10    PHI(KZ,IIG) = PHI(KZ,IIG) + N2(I,IIG)*VO(I)
      IF(NRED)  20,20,999
C     BALANCE TABLES
20    JEB = 8
      CALL NBAL (N2,CO,VO,CXS,MO,M2,EB,JEB)
      DO 21  I=1,IM
      F2(I) = .0
21    PA(I) = .0
      TEMP = 0.0
      IF (NEV .EQ. 5) GO TO 25
      IF (BUCK .EQ. .0) GO TO 26
      TEMP = BUCK
      GO TO 26
25    TEMP = EV
26    DO 38  IIG=1,IGM
      READ(NCR1)  ((CO(II,J), II=1,ITL), J=1,MT)
      IF (TEMP .EQ. .0) GO TO 36
      DO 35  MTZ = 1,MT
      DO 34  KZ = 1,IZM
      IF (MTZ - M2(KZ))  34,30,34
30    TEMP1 = (TEMP*GAM(KZ))/(3.*CO(IHT,MTZ))
      CO(IHT-2,MTZ) = CO(IHT-2,MTZ) - TEMP1
      CO(IHS,MTZ) = CO(IHS,MTZ) + TEMP1
      GO TO 35
34    CONTINUE
35    CONTINUE
```

```
36    WRITE(NSCRAT)  ((CO(I,J), I=1,ITL), J=1,MT)
      DO 37  KZ=1,IZM
      ITEMP = M2(KZ)
37    C2(KZ,IIG) = CO(IHT,ITEMP)
      DO 38  I=1,IM
      ITEMP = MO(I)
      ITEMP = M2(ITEMP)
C     POWER
      F2(I) = F2(I) + 1000. * EXS(IIG,ITEMP) * N2(I,IIG)
C     TOTAL FLUX
38    PA(I) = PA(I) + N2(I,IIG)
      IF(NCG)  42, 42, 39
39    IF(TEMP)  40, 42, 40
40    REWIND NCR1
      REWIND NSCRAT
      ITEMP1 = NSCRAT
      NSCRAT = NCR1
      NCR1 = ITEMP1
42    WRITE(NOUT,45)
45    FORMAT (/'   ZONE  MAT.    RADIUS    AVG. RADIUS   TOTAL FLUX',
     1        '   POWER     FISSION SOURCE'/
     3        '                  (CM)       (CM)        (N/CM2-SEC)',
     4        ' (MWT/LITER)'/)
      DO 50  I=1,IM
      ITEMP = MO(I)
      ITEMP1 = M2(ITEMP)
50    WRITE(NOUT,60)  I,ITEMP,ITEMP1,R1(I),R4(I),PA(I),F2(I),F0(I)
60    FORMAT (I4,I4,I4, 1P5E13.3)
      WRITE (NOUT,70) IP, R1(IP)
70    FORMAT (I4, 7X, 1PE14.3)
      IF (NPRT .EQ. 0 .OR. NPRT .EQ. 2) GO TO 160
75    WRITE(NOUT,80)
80    FORMAT(/' FLUX BY GROUP AND SPACE POINT'//)
      DO 150  IIG=1,IGM,6
      LL = IIG + 5
      IF(LL - IGM)  100,100,90
90    LL = IGM
100   WRITE(NOUT,110)  ((K), K=IIG,LL)
110   FORMAT(/'       ZONE       AVG RAD ', 6(7X,' GROUP',I4))
      WRITE(NOUT,125)
125   FORMAT()
      DO 130  I=1,IM
      ITEMP = MO(I)
130   WRITE(NOUT,140)  I, ITEMP, R4(I), (N2(I,K), K=IIG,LL)
140   FORMAT (I4,I6, 1PE16.5, 1P6E17.5)
      IF(LL .GE. IGM) GO TO 160
150   CONTINUE
160   IF (NPUN .NE. 1) GO TO 200
      DO 190  IIG=1,IGM
190   WRITE(16)  (N2(I,IIG), I=1,IM)
      WRITE(16)  EOFS
      REWIND 16
200   REWIND NCR1
999   RETURN
      END

      SUBROUTINE ACTIVE (F2,CO,MO,M2,N2,R4,MACT,LACT,HOLN,VO,VOL)
      DIMENSION F2(1), CO(ITL,1), MO(1), M2(1), N2(IM,1), R4(1),
     1          MACT(1), LACT(1), HOLN(ML,1), VO(1), VOL(1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE COMPUTES AND PRINTS ACTIVITY RATES
      DO 300 N=1,NACT
      M = MACT(N)
      L = LACT(N)
      IF (M .GT. 0) GO TO 40
      WRITE (NOUT, 20) L
20    FORMAT (/' MACROSCOPIC ACTIVITY FOR POSITION ', I2//)
      GO TO 70
40    WRITE (NOUT,50) M, L
50    FORMAT (/' ACTIVITY FOR MATERIAL ', I3,', POSITION ', I2//)
      IF (M .GT. ML) GO TO 70
      WRITE (NOUT, 60) (HOLN(M,LL),LL=1,2)
60    FORMAT (1X, 2A4, ' ACTIVATION DISTRIBUTION'//)
70    CALL CLEAR (.0, F2(1), IM)
      DO 100 IIG =1,IGM
      READ (NCR1) ((CO(II,J),II=1,ITL),J=1,MT)
```

```
      DO 100 I=1,IM
      IF (M .GT. 0) GO TO 90
      ITEMP = MO(I)
      ITEMP = M2(ITEMP)
      F2(I) = F2(I) + CO(L,ITEMP) * N2(I,IIG)
      GO TO 100
90    F2(I) = F2(I) + CO(L,M) * N2(I,IIG)
100   CONTINUE
      REWIND NCR1
      WRITE (NOUT,120)
120   FORMAT (' MESH   ZONE   AVG. RADIUS(CM)    ACTIVITY RATE'/)
      DO 200 I=1,IM
      ITEMP = MO(I)
      WRITE (NOUT,140) I, MO(I), R4(I), F2(I)
140   FORMAT (I6, 2X, I4, 4X, 1PE12.3, 4X, 1PE12.3)
200   CONTINUE
300   CONTINUE
      DO 600 N=1,NACT
      M = MACT(N)
      L = LACT(N)
      IF (M .GT. 0) GO TO 340
      WRITE (NOUT, 20) L
      GO TO 370
340   WRITE (NOUT,50) M, L
      IF (M .GT. ML) GO TO 370
      WRITE (NOUT, 360) (HOLN(M,LL),LL=1,2)
360   FORMAT (1X, 2A4, 'ACTIVATION TOTALS'//)
370   TEMP2 = .0
      TEMP3 = .0
      WRITE (NOUT, 380)
380   FORMAT (5X, ' ZONE', 8X, ' VOLUME ', 3X, 'AVG. ACTIVITY', 3X,
     1        'TOT. ACTIVITY'/
     2        5X, ' (#) ', 8X, '(LITERS)', 4X, ' (#/SEC-CC)  ', 2X,
     3        '   (#/SEC)   '/)
      DO 500 IZ=1,IZM
      TEMP = .0
      TEMP1 = .0
      VOL(IZ) = .0
      DO 400 IIG =1,IGM
      READ (NCR1) ((CO(II,J),II=1,ITL),J=1,MT)
      DO 400 I=1,IM
      ITEMP = MO(I)
      IF (ITEMP .NE. IZ) GO TO 400
      IF (IIG .EQ. 1) VOL(ITEMP) = VOL(ITEMP) + .001 * VO(I)
      IF (M .GT. 0) GO TO 390
      ITEMP = M2(ITEMP)
      TEMP = TEMP + CO(L,ITEMP) * N2(I,IIG) * VO(I)
      GO TO 400
390   TEMP = TEMP + CO(L,M) * N2(I,IIG) * VO(I)
400   CONTINUE
      IF (VOL(IZ) .LE. .0) GO TO 430
      TEMP1 = .001*TEMP/VOL(IZ)
430   WRITE (NOUT,440) IZ, VOL(IZ), TEMP1, TEMP
440   FORMAT (2X, I6, 4X, 1P3E14.3)
      TEMP2 = TEMP2 + TEMP
      TEMP3 = TEMP3 + VOL(IZ)
      REWIND NCR1
500   CONTINUE
      ITEMP = 0
      TEMP1 = .001 * TEMP2/TEMP3
      WRITE (NOUT, 560)
560   FORMAT ()
      WRITE (NOUT, 440) ITEMP, TEMP3, TEMP1, TEMP2
600   CONTINUE
      REWIND NCR1
      RETURN
      END

      SUBROUTINE FISCAL (N2, FO, VO, CO, K6, MO, M2, EB, EXS)
$INCLUDE:'1DB.INC'
      DIMENSION N2(IM,1), FO(1), VO(1), CO(ITL,1), K6(1), MO(1),
     1          M2(1), EB(8,1), EXS(IGM,1)
      LAR = ALA
C     FISSION SUMS
      DO 10 IIG = 1,IGM
      EB(8,IIG) = .0
```

```
      DO 10 I = 1,IM
      ITEMP = M0(I)
      ITEMP = M2(ITEMP)
10    EB(8,IIG) = EB(8,IIG) + EXS(IIG,ITEMP)*N2(I,IIG)*V0(I)
      IF (A02) 20, 40, 20
20    DO 30  IIG=1,IGM
      READ (NCR1) ((C0(I,J),I=1,ITL),J=1,MT)
      EB(1,IIG) = .0
      DO 30  I=1,IM
      ITEMP=M0(I)
      ITEMP=M2(ITEMP)
30    EB(1,IIG) = EB(1,IIG) + C0(IHT-1,ITEMP)*F0(I)*V0(I)
      REWIND NCR1
      GO TO 70
40    E01=0.
      DO 50  I=1,IM
50    E01=E01+V0(I)*F0(I)
      DO 60  IIG=1,IGM
60    EB(1,IIG) = K6(IIG)*E01
70    EB(1,IGP) = .0
      EB(8,IGP) = .0
      DO 80  IIG=1,IGM
      EB(1,IGP) = EB(1,IGP) + EB(1,IIG)
80    EB(8,IGP) = EB(8,IGP) + EB(8,IIG)
C     CALCULATE LAMBDA
      IF (B07 .GT. 0) GO TO 140
      ALA = EB(1,IGP)/T11
      TEMP=1.0/ALA
      IF (NEV-1) 140, 100, 140
100   DO 110  IIG=1,IGM
      EB(1,IIG) = EB(1,IIG)*TEMP
110   K6(IIG)=K6(IIG)*TEMP
      EB(1,IGP) = EB(1,IGP)*TEMP
      IF (A02) 120, 140, 120
120   DO 130  I=1,IM
130   F0(I)=F0(I)*TEMP
C     NORMALIZATION
140   B07=0
150   IF (S01)  160, 300, 170
160   E01 = ABS(S01)/EB(8,IGP)
      GO TO 180
170   E01 = S01/EB(1,IGP)
180   DO 190 IIG=1,IGP
190   EB(1,IIG) = E01*EB(1,IIG)
      DO 200 I=1,IM
200   F0(I)=E01*F0(I)
220   IF (NXCN) 300, 300, 240
240   DO 250 IIG=1,IGM
      DO 250 I=1,IM
250   N2(I,IIG) = E01*N2(I,IIG)
300   CONTINUE
      RETURN
      END

      SUBROUTINE FXINP (N0)
      DIMENSION  N0(IM,1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE READS FLUXES
      IF (NFX .GE. 1)  GO TO 50
      DO 40  IIG=1,IGM
      DO 40  I=1,IM
40    N0(I,IIG) = 1.0
      GO TO 90
50    DO 70  IIG=1,IGM
70    READ (14) (N0(I,IIG), I=1,IM)
      REWIND 14
90    RETURN
      END

      SUBROUTINE INIT (K6, K7, I0, I1, I2, M0, M2, N0, R0, R1, R3, R4,
     1                 R5, A0, F0, C0, V0, V7, FEF, EXS, GAM, HOLN)
$INCLUDE:'1DB.INC'
      DIMENSION  K6(1), K7(1), I0(1), I1(1), I2(1), R0(1), R1(1),
     1           R3(1), R4(1), R5(1), A0(1), C0(ITL,1),
     2           V0(1), M0(1), M2(1), N0(IM,1), F0(1), V7(1),
     3           FEF(1), EXS(IGM,1), GAM(1), HOLN(ML,1)
```

```
C     PERFORM ADJOINT REVERSALS ON V7 AND K7
      IF (P02 .EQ. 0)  WRITE (NOUT,2) DAY
2     FORMAT (//24X, 'T I M E  = ',1PE14.3, ' D A Y S'//)
      IF (A02 .EQ. 0) GO TO 40
      IF (P02 .NE. 0) GO TO 40
      IF (R02 .GT. 0) GO TO 40
      IIG = 1
      IGBAR=IGM
30    TEMP=K7(IIG)
      K7(IIG)=K7(IGBAR)
      K7(IGBAR)=TEMP
      TEMP=V7(IIG)
      V7(IIG)=V7(IGBAR)
      V7(IGBAR)=TEMP
      IIG=IIG+1
      IGBAR=IGBAR-1
      IF (IIG-IGBAR)  30, 40, 40
C     MIX CROSS SECTIONS
40    NO7=1
      IF (P02)  50, 60, 50
50    GO TO (430,430,120,430,320),  NEV
60    IF (M01)  90, 90, 70
70    IF (R02)  75,75,90
75    WRITE (NOUT, 77)
77    FORMAT (/' MIXTURE SPECIFICATIONS    '//
     1        3X, ' MIXTURE NUMBER    MIX COMMAND    ',
     2        ' MATERIAL DENSITY    NAME '//)
      DO 88 J=1,M01
      I = I1(J)
      IF (I .GE. 1 .AND. I .LE. ML) GO TO 84
      WRITE (NOUT, 80) J, I0(J), I1(J), I2(J)
80    FORMAT (I4,1X,I8,7X,I8,5X,1PE16.5,6X,2A4)
      GO TO 85
84    WRITE (NOUT,80) J, I0(J), I1(J), I2(J), (HOLN(I,LL), LL=1,2)
85    IF (J .EQ. M01 .OR. I0(J) .EQ. I0(J+1)) GO TO 88
      WRITE (NOUT, 86)
86    FORMAT (1X, 15('-----'))
88    CONTINUE
90    IF (NPRT .LE. 1)  GO TO 120
      IF (NRED .GT. 0)  GO TO 120
      KPAGE = 100
      WRITE (NOUT,110)
110   FORMAT (/' SHIELDED CROSS SECTIONS')
120   REWIND NCR1
      DO 128 IIG = 1,IGM
      READ (NCR1) ((C0(I,J), I=1,ITL), J=1,MT)
      DO 128 M=1,MT
      EXS(IIG,M) = .0
      IF (M .LE. ML) EXS(IIG,M) = C0(IHT-3,M)*FEF(M)*1.602E-19
      DO 126 MM=1,M01
      IF (M .NE. I0(MM)) GO TO 126
      MMM = I1(MM)
      IF (MMM .EQ. 0) GO TO 126
      IF (MMM .EQ. M) GO TO 124
      EXS(IIG,M) = EXS(IIG,M) + I2(MM) * EXS(IIG,MMM)
      GO TO 126
124   EXS(IIG,M) = EXS(IIG,M) + EV * EXS(IIG,M)
126   CONTINUE
128   CONTINUE
      REWIND NCR1
      DO 310  IIG=1,IGM
      READ (NCR1)    ((C0(I,J),I=1,ITL),J=1,MT)
      IF (M01) 130, 240, 130
130   DO 230  M=1,M01
      IF (I0(M)-MT) 150, 150, 140
140   CALL ERRO2('  INIT',140,1)
150   IF (I1(M)-MT) 160, 160, 140
160   N=I0(M)
      L=I1(M)
      E01=I2(M)
      IF (L) 200, 200, 170
170   IF (E01) 200, 180, 200
180   IF (N-L) 200, 190, 200
190   E01 = EV
      L = 0
200   DO 230  I=1,ITL
```

```
      IF (L) 210, 220, 210
210   CO(I,N)=CO(I,N)+CO(I,L)*E01
      GO TO 230
220   CO(I,N)=CO(I,N)*E01
230   CONTINUE
240   IF (P02) 300, 250, 300
250   IF (NPRT - 1) 300, 300, 260
260   IF (NRED)  265,265,300
265   WRITE (NOUT, 270)  IIG
270   FORMAT (/' GROUP',I3,'     N,2N        SIGF        SIGA',
     1        '       NUSIGF      SIGTR ...  GXG ...')
      ITEMP = NST + 1
      DO 280  N=1,MT
280   WRITE (NOUT, 290)  N,(CO(I,N),I=ITEMP,ITL)
290   FORMAT (' MAT',I3,1P6E12.2/(7X,1P6E12.2))
300   WRITE (NSCRAT)  ((CO(I,J),I=1,ITL),J=1,MT)
310   CONTINUE
      REWIND NCR1
      REWIND NSCRAT
C     SWITCH TAPE DESIGNATIONS
      ITEMP=NSCRAT
      NSCRAT=NCR1
      NCR1=ITEMP
320   IF (NEV .EQ. 5) GO TO 350
      IF (BUCK .EQ. .0) GO TO 430
      TEMP = BUCK
      GO TO 380
350   IF (P02 .GT. 0)  GO TO 370
      BUCK = 0.
370   TEMP = EV - BUCK
      BUCK = EV
380   DO 420   IIG=1,IGM
      READ(NCR1)  ((CO(I,J), I=1,ITL),J=1,MT)
      IF (TEMP .EQ. .0) GO TO 415
      DO 410  MTZ = 1,MT
      DO 400  KZ=1,IZM
      IF (M2(KZ) - MTZ)  400, 390, 400
390   TEMP1 = (TEMP*GAM(KZ))/(3.*CO(IHT,MTZ))
      CO(IHT-2,MTZ) = CO(IHT-2,MTZ) + TEMP1
      CO(IHS,MTZ) = CO(IHS,MTZ) - TEMP1
      GO TO 410
400   CONTINUE
410   CONTINUE
415   WRITE (NSCRAT)  ((CO(I,J), I=1,ITL),J=1,MT)
420   CONTINUE
      REWIND NCR1
      REWIND NSCRAT
C     SWITCH TAPE DESIGNATIONS
      ITEMP = NSCRAT
      NSCRAT = NCR1
      NCR1 = ITEMP
C
C     MODIFY GEOMETRY
430   IF (P02) 460, 440, 460
440   IF (R02)  445, 445, 620
445   DO 450  I=1,IP
450   R1(I)=R0(I)
460   IF (NEV-4)  490, 470, 490
470   DO 480 I=1,IM
      K=M0(I)
480   R1(I+1)=R1(I)+(R0(I+1)-R0(I))*(1.0+ EV*R3(K))
C
C     CALCULATE AREAS AND VOLUMES
490   PI2=6.28318
      IF (P02) 500, 510, 500
500   IF (NEV - 4)  620, 510, 620
510   DO 570 I=1,IM
      R4(I)=(R1(I+1)+R1(I))*0.5
      R5(I)=R1(I+1)-R1(I)
      IF ( R5(I) ) 520, 520, 530
520   CALL ERR02 ('  INIT',520,1)
530   GO TO (540,550,560),IGEP
540   A0(I)=1.0
      A0(IP)=1.0
      V0(I) = R5(I)
      GO TO 570
```

```
550   AO(I)=PI2*R1(I)
      AO(IP)=PI2*R1(IP)
      VO(I) = PI2*R5(I)*R4(I)
      GO TO 570
560   AO(I) = 2.*PI2*R1(I)*R1(I)
      AO(IP) = 2.*PI2*R1(IP)*R1(IP)
      VO(I) = (2.0*PI2*(R1(I+1)**3 - R1(I)**3))/3.0
570   CONTINUE
C
C     CALCULATE EFFECTIVE FISSION SPECTRUM
620   IF (P02) 680, 640, 680
640   SK7=0.
      DO 670 IIG=1,IGM
      IF (S02-1) 660, 650, 660
650   K6(IIG)=K7(IIG)/S03
      GO TO 670
660   K6(IIG)=K7(IIG)
670   SK7=SK7+K7(IIG)
680   CONTINUE
C
C     CALCULATE FISSION RATE
690   CALL CLEAR (0.0, F0, IM)
      DO 720 IIG=1,IGM
      READ (NCR1)  ((CO(I,J),I=1,ITL),J=1,MT)
      DO 720  I = 1, IM
      ITEMP=M0(I)
      ITEMP=M2(ITEMP)
      IF (A02) 700, 710, 700
700   F0(I)=F0(I)+K7(IIG)*N0(I,IIG)
      GO TO 720
710   F0(I)=F0(I)+C0(IHT-1,ITEMP)*N0(I,IIG)
720   CONTINUE
      REWIND NCR1
      RETURN
      END

      SUBROUTINE INNER (N0,N2,CXS,S2,HA,PA)
$INCLUDE:'1DB.INC'
      DIMENSION  N0(IM,1), N2(IM,1), CXS(IP,IGM,1), S2(1), HA(1), PA(1)
      IF (P02 - 1)  12, 12, 5
5     DO 10  I=1,IM
10    N0(I,IGV) = N2(I,IGV)
12    I=1
C     FLUX CALCULATION FOR VACUUM AND REFLECTIVE BOUNDARY CONDITIONS
      IF (B01-1)  15,15,40
15    HA(I) = CXS(I+1,IGV,1)/CXS(I,IGV,2)
      PA(I) = S2(I)/CXS(I,IGV,2)
      DO 20  I=2,IM
      HA(I) = CXS(I+1,IGV,1)/(CXS(I,IGV,2) - CXS(I,IGV,1)*HA(I-1))
20    PA(I) = (S2(I) + CXS(I,IGV,1)*PA(I-1))/(CXS(I,IGV,2) -
     1        CXS(I,IGV,1)*HA(I-1))
      N2(IM,IGV) = PA(IM)
      DO 30  KI=2,IM
      I = IM + 1 - KI
30    N2(I,IGV) = PA(I) + HA(I)*N2(I+1,IGV)
      GO TO 110
C     FLUX CALCULATION FOR PERIODIC BOUNDARY CONDITIONS
40    HA(I) = CXS(I+1,IGV,1)/CXS(I,IGV,2)
      N2(I,IGV) = CXS(1,IGV,1)/CXS(I,IGV,2)
      PA(I) = S2(I)/CXS(I,IGV,2)
      TEMP1 = N2(I,IGV)
      TEMP = HA(I)
      TEMP2 = PA(I)
      IKB = IM - 1
      DO 50  I=2,IKB
      HA(I) = CXS(I+1,IGV,1)/(CXS(I,IGV,2) - CXS(I,IGV,1)*HA(I-1))
      N2(I,IGV) = CXS(I,IGV,1)*N2(I-1,IGV)/(CXS(I,IGV,2) -
     1            CXS(I,IGV,1)*HA(I-1))
      PA(I) = (S2(I) + CXS(I,IGV,1)*PA(I-1))/(CXS(I,IGV,2) -
     1       CXS(I,IGV,1)*HA(I-1))
      TEMP1 = TEMP1 + TEMP*N2(I,IGV)
      TEMP2 = TEMP2 + TEMP*PA(I)
50    TEMP = TEMP*HA(I)
      I = IM
      TEMP1 = (TEMP1 + TEMP)*CXS(1,IGV,1) + CXS(I,IGV,1)*N2(I-1,IGV)
      N2(I,IGV) = (S2(I) + CXS(I,IGV,1)*PA(I-1) + CXS(1,IGV,1)*TEMP2)/
```

```
     1          (CXS(I,IGV,2) - CXS(I,IGV,1)*HA(I-1) - TEMP1)
      DO 60  KI=2,IM
      I = IM + 1 - KI
60    N2(I,IGV) = HA(I)*N2(I+1,IGV) + N2(I,IGV)*N2(IM,IGV) + PA(I)
110   RETURN
      END

      SUBROUTINE INNER1 (MO,M2,CXS,VO,CO,AO,R5,R4)
      DIMENSION  MO(1), M2(1), CXS(IP,IGM,2), VO(1), CO(ITL,1),
     1           AO(1), R5(1), R4(1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE CALCULATES COEFFICIENTS FOR THE FLUX EQUATION
      DO 30  I=1,IM
      ITEMP = MO(I)
      ITEMP = M2(ITEMP)
      CXS(I,IGV,2) = VO(I)*(CO(IHT,ITEMP) - CO(IHS,ITEMP))
      IF (I - 1)  30, 30, 5
5     ITEMP1 = MO(I-1)
      ITEMP1 = M2(ITEMP1)
      IF (ITEMP - ITEMP1)  20,10,20
10    CXS(I,IGV,1) = AO(I)/(3.*CO(IHT,ITEMP)*(R4(I) - R4(I-1)))
      GO TO 30
20    CXS(I,IGV,1) = AO(I)*(R5(I-1) + R5(I))/((R4(I) - R4(I-1))*
     1  (3.*(R5(I-1)*CO(IHT,ITEMP1) + R5(I)*CO(IHT,ITEMP))))
30    CONTINUE
      DO 200  I=1,IM
      ITEMP = MO(I)
      ITEMP = M2(ITEMP)
      IF (I-1)  40, 40, 80
40    IF (BO1 - 1)  50, 60, 70
50    CXS(I,IGV,1) = AO(I)/(3.*CO(IHT,ITEMP)*(.5*R5(I) + .71/
     1  CO(IHT,ITEMP)))
      GO TO 200
60    CXS(I,IGV,1) = .0
      GO TO 200
70    ITEMP3 = MO(IM)
      ITEMP3 = M2(ITEMP3)
      CXS(I,IGV,1) = 2.*AO(I)/(3.*(R5(IM)*CO(IHT,ITEMP3) +
     1   R5(I)*CO(IHT,ITEMP)))
      GO TO 200
80    IF (I - IM)  200, 90, 90
90    IF (BO2 - 1)  100, 110, 120
100   CXS(I+1,IGV,1) = AO(I+1)/(3.*CO(IHT,ITEMP)*(.5*R5(I) +
     1  .71/CO(IHT,ITEMP)))
      GO TO 200
110   CXS(I+1,IGV,1) = .0
      GO TO 200
120   CXS(I+1,IGV,1) = CXS(1,IGV,1)
200   CXS(I,IGV,2) = CXS(I,IGV,2) + CXS(I,IGV,1) + CXS(I+1,IGV,1)
      RETURN
      END

      SUBROUTINE MONPR
$INCLUDE:'1DB.INC'
C     MONITOR PRINT
      CALL TIMEX (SEC)
      TI = (SEC - GLH)/60.
      KPAGE = KPAGE + 1
      IF (KPAGE .LT. 40)  GO TO 40
      KPAGE = 0
      WRITE (NOUT,20 )  (ID(I), I=1,20)
20    FORMAT (//1X, 20A4/)
      WRITE (NOUT,25)
25    FORMAT ()
      WRITE (NOUT, 30 )
30    FORMAT(5X,'   TIME          OUTER    EIGENVALUE       EIGENVALUE',
     1        '          LAMBDA '/5X,
     2        ' (MINUTES)   ITERATIONS     SLOPE'/)
40    WRITE (NOUT, 50) TI, P02, EQ, EV, ALA
50    FORMAT (E12.3, I12, E18.8, 2E18.8)
60    P02=P02+1
      IF (TMAX .EQ. .0)     GO TO 67
      IF (TI .GT. 60.*TMAX) GO TO 70
67    IF (P02 .LE. NOIT)    GO TO 80
70    NGOTO = 1
      NRED = 0
```

```
      GO TO 90
80    NGOTO = 4
90    RETURN
      END

      SUBROUTINE NBAL (N2,CO,VO,CXS,MO,M2,EB,JEB)
      DIMENSION N2(1), CO(ITL,1), VO(1), CXS(1),
     1          MO(1), M2(1), EB(JEB,1)
$INCLUDE:'1DB.INC'
C     COMPUTE AND PRINT BALANCE TABLES
      DO 80  IIG = 1,IGM
      READ (NCR1) ((CO(I,J), I=1,ITL), J=1,MT)
      DO 10 K=3,7
10    EB(K,IIG) = .0
      DO 20 I=1,IM
      II = I + IM*(IIG-1)
      TEMP = VO(I) * N2(II)
      ITEMP1 = MO(I)
      ITEMP = M2(ITEMP1)
C     OUT-SCATTER
      EB(3,IIG) = EB(3,IIG) + (CO(IHT,ITEMP) - CO(IHS,ITEMP) -
     1             CO(IHT-2,ITEMP))*TEMP
C     ABSORPTION
20    EB(4,IIG) = EB(4,IIG) + CO(IHT-2, ITEMP) * TEMP
C     LEFT LEAKAGE
      IF (B01 .GT. 1) GO TO 40
      IF (B01 .EQ. 1) GO TO 50
      II = 1 + IM*(IIG-1)
      KK = 1 + IP*(IIG-1)
      EB(5,IIG) = N2(II) * CXS(KK)
      GO TO 50
40    II = 1 + IM*(IIG-1)
      KK = 1 + IP*(IIG-1)
      EB(5,IIG) = (N2(II) - N2(II+IM-1)) * CXS(KK)
50    IF (B02 .GT. 1) GO TO 70
      IF (B02 .EQ. 1) GO TO 80
      II= IM*IIG
      KK = 1 + IM + IP*(IIG-1)
      EB(6,IIG) = N2(II) * CXS(KK)
      GO TO 80
70    EB(6,IIG) = (-1.0) * EB(5,IIG)
80    EB(7,IIG) = EB(6,IIG) + EB(5,IIG)
      REWIND NCR1
      DO 110 K=2,7
110   EB(K,IGP) = .0
      DO 120 K=2,7
      DO 120 IIG=1,IGM
      EB(K,IGP) = EB(K,IGP) + EB(K,IIG)
120   CONTINUE
      WRITE (NOUT,130)
130   FORMAT (//' FINAL NEUTRON BALANCE TABLE'//
     1 ' GROUP  FISS. SOURCE  IN-SCATTER OUT-SCATTER ABSORPTION',
     2 '  LEFT LEAK.  RIGHT LEAK.'/)
      DO 140 IIG = 1, IGM
      WRITE (NOUT,150) IIG, (EB(L,IIG), L=1,6)
140   CONTINUE
150   FORMAT (I4, 3X, 1P6E12.4)
      WRITE (NOUT,155)
155   FORMAT ()
      WRITE (NOUT,150) IGP, (EB(K,IGP), K=1,6)
      WRITE (NOUT,155)
      RETURN
      END

      SUBROUTINE OUTER (AO,CO,FO,K6,MO,M2,NO,N2,S2,VO,V7,F2,
     1                  CXS, R5, R4, HA, PA, EB, EXS)
      DIMENSION AO(1), FO(1), K6(1), MO(1), M2(1), NO(IM,1), N2(IM,1),
     1          S2(1), VO(1), V7(1), F2(1), CO(ITL,1), HA(1), PA(1),
     2          CXS(IP,IGM,1),R5(1), R4(1), EB(8,1), EXS(IGM,1)

      INTEGER   GBAR, PBAR, SBAR
$INCLUDE:'1DB.INC'
      IGV=1
C     SOURCE CALCULATION
20    READ (NCR1) ((CO(I,M), I=1,ITL), M=1,MT)
      DO 30  I=1,IM
```

```
30    S2(I)=0.
40    IF (A02 .NE. 0) GO TO 70
      DO 60  I=1,IM
60    S2(I)=S2(I)+K6(IGV)*F0(I)
      GO TO 90
70    DO 80 I=1,IM
      ITEMP1=M0(I)
      ITEMP1=M2(ITEMP1)
80    S2(I)=S2(I)+C0(IHT-1,ITEMP1)*F0(I)
90    GBAR=IGV+IHS-ITL
      IF (GBAR-1) 100, 110, 110
100   GBAR=1
110   PBAR = IHS + IGV - 1
      IF (PBAR - ITL)  115, 115, 112
112   PBAR = ITL
115   IF (GBAR - IGV)  120, 140, 120
120   DO 130  I=1,IM
      ITEMP1=M0(I)
      ITEMP1=M2(ITEMP1)
      ITEMP=ITEMP1
      TEMP=C0(PBAR,ITEMP)
130   S2(I)=S2(I)+N2(I,GBAR)*TEMP
140   GBAR=GBAR+1
      PBAR=PBAR-1
      IF (GBAR - IGM) 145, 145, 150
145   IF (GBAR - IGV - NXCN)  115, 115, 150
150   IF (IGV - IGM)  170, 160, 170
160   REWIND NCR1
170   V11 = 0.
      DO 180  I=1,IM
      S2(I)=S2(I)*V0(I)
180   V11=V11+S2(I)
      EB(2,IGV) = V11 - EB(1,IGV)
C     SOURCE ALPHA
190   IF (NEV - 2)  200, 230, 200
200   IF (S02 - 2)  220, 210, 220
210   T7 = S03/V7(IGV)
      GO TO 240
220   T7 = 0.0
      GO TO 260
230   T7 = EV/V7(IGV)
240   DO 250  K = 1, IZM
      ITEMP1 = M2(K)
250   C0(IHS, ITEMP1) = C0(IHS,ITEMP1) - T7
260   CONTINUE
C     GROUP FLUX CALCULATION
270   IF (V11) 280, 300, 280
280   IF (P02 - 1)  284, 284, 282
282   IF (NEV-1)  284, 286, 284
284   CALL INNER1 (M0,M2,CXS,V0,C0,A0,R5,R4)
286   CALL INNER (N0,N2,CXS,S2,HA,PA)
      DO 290  K = 1,IZM
      ITEMP1 = M2(K)
290   C0(IHS,ITEMP1) = C0(IHS,ITEMP1) + T7
      GO TO 320
300   DO 310 I=1,IM
      N2(I,IGV) = .0
310   N0(I,IGV) = .0
C     CALCULATE FISSION RATE AND FISSION SOURCE RATE
320   IF (V11) 330, 380, 330
330   IF (A02) 340, 360, 340
340   EB(8,IGV) = .0
      DO 350  I=1,IM
      ITEMP1=M0(I)
      ITEMP1=M2(ITEMP1)
      EB(8,IGV) = EB(8,IGV) + EXS(IGV,ITEMP1)*N2(I,IGV)*V0(I)
350   F2(I)=F2(I)+K6(IGV)*N2(I,IGV)
      GO TO 380
360   EB(8,IGV) = .0
      DO 370  I=1,IM
      ITEMP1=M0(I)
      ITEMP1=M2(ITEMP1)
      EB(8,IGV) = EB(8,IGV) + EXS(IGV,ITEMP1)*N2(I,IGV)*V0(I)
370   F2(I)=F2(I)+C0(IHT-1,ITEMP1)*N2(I,IGV)
380   IGV=IGV+1
      IF (IGV-IGM) 20, 20, 390
```

```
390   T11 = EB(1,IGP)
      REWIND NCR1
C     OVER-RELAX FISSION SOURCE
      E01 = .0
      E02 = .0
      EB(1,IGP) = .0
      IF (A02 .EQ. 0) GO TO 580
C     FOR ADJOINT CALCULATION, S2(I) WILL STORE ORFED F2(I)
      DO 520  I=1,IM
520   S2(I) = F0(I) + ORF*(F2(I) - F0(I))
      DO 540  IIG=1,IGM
      READ (NCR1)  ((C0(I,J),I=1,ITL), J=1,MT)
      DO 540  I=1,IM
      ITEMP = M0(I)
      ITEMP = M2(ITEMP)
      E01 = E01 + C0(IHT-1,ITEMP)*F2(I)*V0(I)
540   E02 = E02 + C0(IHT-1,ITEMP)*S2(I)*V0(I)
      TEMP1 = E01/E02
      DO 550  I=1,IM
550   F0(I) = TEMP1*S2(I)
      REWIND NCR1
      GO TO 620
580   DO 590  I=1,IM
      E01 = E01 + V0(I)*F2(I)
      F2(I) = F0(I) + ORF*(F2(I) - F0(I))
590   E02 = E02 + V0(I)*F2(I)
      TEMP1 = E01/E02
      DO 600  I=1,IM
600   F0(I) = TEMP1*F2(I)
620   CONTINUE
      RETURN
      END

      SUBROUTINE RCCAL1 (NXS,I1,SR,PSF,I2,FSS,SIGO)
      DIMENSION   NXS(1), I1(1), SR(8,1), PSF(1), I2(1),
     1            FSS(5,NRCX,1), SIGO(NRCX,1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE CALCULATES SIGO
      DO 70  M=1,NRCX
      DO 70  IIG=1,IGM
70    SIGO(M,IIG) = 0.0
      ITEMP2 = 0
      DO 180  L=1,NRCM
      ITEMP1 = ITEMP2 + 2
      ITEMP2 = ITEMP1 + NXS(L) - 2
      DO 180  J=ITEMP1,ITEMP2
      M = I1(J)
      READ(NSR)  ((SR(K,IIG), K=1,8), IIG=1,IGM)
      DO 180 JJ=ITEMP1,ITEMP2
      MM = I1(JJ)
      IF (M .EQ. MM) GO TO 180
      DO 170  IIG=1,IGM
      IF (I2(JJ) .GT. 0.)  GO TO 140
      SIGO(MM,IIG) = 1.0E+12
      GO TO 170
140   IF (IRED .EQ. 2) GO TO 155
      TEMP = SR(2,IIG) + SR(4,IIG) + SR(5,IIG) + SR(6,IIG)
      SIGO(MM,IIG) = SIGO(MM,IIG) + PSF(J)*I2(J)*TEMP/I2(JJ)
      GO TO 170
155   TEMP = SR(2,IIG)*FSS(1,M,IIG) + SR(4,IIG)*FSS(2,M,IIG)
     1     + SR(5,IIG)                + SR(6,IIG)*FSS(4,M,IIG)
      SIGO(MM,IIG) = SIGO(MM,IIG) + PSF(J)*I2(J)*TEMP/I2(JJ)
170   CONTINUE
180   CONTINUE
      IRED = 2
      REWIND NSR
      RETURN
      END

      SUBROUTINE RCCAL2 (NXS,I1,NGB,NGE,NPFF,FF,SIGO,SF,FSS,HOLN)
      DIMENSION  NXS(1), I1(1), NGB(1), NGE(1), NPFF(5,1),
     1           FF(5,IGM,10,1), SIGO(NRCX,1), SF(5,ML,1),
     2           FSS(5,NRCX,1), HOLN(ML,1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE CALCULATES F FACTORS USING KIDMANS INTERPOLATIONS
      ITEMP2 = 0
```

```
      DO 400  L=1,NRCM
      ITEMP1 = ITEMP2 + 2
      ITEMP2 = ITEMP1 + NXS(L) - 2
      DO 400  J=ITEMP1,ITEMP2
      M = I1(J)
      ITEMP = I1(J)
      NB = NGB(ITEMP)
      NE = NGE(ITEMP)
      DO 300  K=1,5
      DO 10  IIG=1,IGM
10    FSS(K,M,IIG)= 1.0
      NF = NPFF(K,ITEMP)
      IF (NF .EQ. 0) GO TO 300
      READ (NFF)  ((FF(K,IIG,1,N), N=1,NF), IIG=NB,NE)
      DO 250 IIG=NB,NE
      DO 80 NN=2,NF
      N=NN
      IF (SIGO(M,IIG).GT.SF(K,ITEMP,N)) GO TO 100
80    CONTINUE
      IF (SIGO(M,IIG) .LE. .1*SF(K,ITEMP,NF))   SIGO(M,IIG) =
     1   .1*SF(K,ITEMP,NF)
100   IF (SIGO(M,IIG) .GT. 10.*SF(K,ITEMP,1))   SIGO(M,IIG) =
     1   10.*SF(K,ITEMP,1)
      AA=FF(K,IIG,1,N-1)
      BB=FF(K,IIG,1,N)
      CC=SF(K,ITEMP,N-1)
      DD=SF(K,ITEMP,N)
      FSS(K,M,IIG) = AA + (BB-AA) * (ALOG(SIGO(M,IIG)/CC)
     1             / ALOG(DD/CC))
      TEMP = AMIN1(0.01, FF(K,IIG,1,NF))
      IF (FSS(K,M,IIG) .LT. TEMP)  FSS(K,M,IIG) = TEMP
250   CONTINUE
300   CONTINUE
      IF (NSIGO .LT. 2) GO TO 400
      IF (NPRT .LE. 1) GO TO 400
      WRITE (NOUT,330) (HOLN(ITEMP,LL),LL=1,2), M, (IIG, SIGO(M,IIG),
     1                 (FSS(K,M,IIG), K=1,5), IIG=1,IGM)
330   FORMAT (//' SHIELDING FACTORS FOR ',2A4,', MATERIAL',I3//
     1        '  GROUP     SIGO          FF            FC       ',
     2        '  FTR           FE            FK  '/(I4, E12.2, 5F12.2))
400   CONTINUE
      REWIND NFF
      RETURN
      END

      SUBROUTINE RCCHK (M, SR, HOLN, SM, SE, NT, NPFF, FF, NGB, NGE)
      DIMENSION  SR(8,1), HOLN(ML,1), SM(IGM,1), SE(IGM,1), NT(1),
     1           NPFF(5,1), FF(5,IGM,10,1), NGB(1), NGE(1)
$INCLUDE:'1DB.INC'
      DIMENSION  HF(2,5)
      CHARACTER*4 HF
      DATA   HF(1,1), HF(2,1), HF(1,2), HF(2,2), HF(1,3), HF(2,3),
     1       HF(1,4), HF(2,4), HF(1,5), HF(2,5) /
     2       'FISS','ION ','CAPT','URE ','TRAN','SPRT ','ELAS','TIC ',
     3       'KERM','MA  '/
C     THIS SUBROUTINE CHECKS THE RUSSIAN DATA FOR CONSISTENCY
      EEPS=.02
      IF (ME .GT. MAXE) GO TO 900
      DO 300 IIG=1,IGM
      IF (SR(1,IIG) .GT. .0) GO TO 30
      ME = ME + 1
      IF (ME .GT. MAXE) GO TO 900
      WRITE (NOUT, 20) ME, (HOLN(M,L),L=1,2), IIG, SR(1,IIG)
20    FORMAT (I4, 2X, ' CHECK TRANSPORT IN ', 2A4, '  , GROUP', I4,
     1        ',  SIGTR = ', 1PE12.3)
30    TEMP = SR(2,IIG) + SR(4,IIG) + SR(5,IIG) + SR(6,IIG)
      IF (SR(1,IIG) .LE. TEMP) GO TO 70
40    ME = ME + 1
      IF (ME .GT. MAXE) GO TO 900
      TEMP = TEMP - SR(1,IIG)
      WRITE (NOUT, 50) ME, (HOLN(M,L), L=1,2), IIG, TEMP
50    FORMAT (I4, 2X, ' CHECK TRANSPORT IN ', 2A4,'  , GROUP', I4,
     1        ',  ERROR = ', 1PE12.3)
70    TEMP1 = .0
      TEMP2 = .0
      DO 100 K=1,IGM
```

```
      TEMP1 = TEMP1 + SM(IIG,K)
100   TEMP2 = TEMP2 + SE(IIG,K)
      TEMP1 = TEMP1 - SR(5,IIG)
      TEMP2 = TEMP2 - SR(6,IIG)
      IF (ABS(TEMP1) .LE. EEPS * SR(5,IIG)) GO TO 200
      ME = ME + 1
      IF (ME .GT. MAXE) GO TO 900
      WRITE (NOUT,150)  ME, (HOLN(M,L),L=1,2), IIG, TEMP1
150   FORMAT (I4, 2X, ' CHECK INELASTIC IN ', 2A4,'   , GROUP',I4,
     1        ',', 1X, ' ERROR = ', 1PE12.3)
200   IF (ABS(TEMP2) .LE. EEPS * SR(6,IIG)) GO TO 300
      ME = ME + 1
      IF (ME .GT. MAXE) GO TO 900
      WRITE (NOUT,250)  ME, (HOLN(M,L),L=1,2), IIG, TEMP2
250   FORMAT (I4, 2X, ' CHECK ELASTIC IN ', 2A4,'     , GROUP',I4,
     1        ',', 1X, ' ERROR = ', 1PE12.3)
300   CONTINUE
      NB = NGB(M)
      NE = NGE(M)
      DO 505  K=1,5
      NF = NPFF(K,M)
      IF (NF .LE. 0) GO TO 505
      DO 500  N=1,NF
      DO 500  J=NB,NE
      DO 500  L=1,NT(M)
      IF (N .LE. 1) GO TO 430
      TEMP2 = FF(K,J,L,N) - FF(K,J,L,N-1)
      IF (TEMP2 .LE. EEPS)  GO TO 430
      ME = ME + 1
      IF (ME .GT. MAXE) GO TO 900
      WRITE (NOUT,420) ME, (HF(JKL,K), JKL=1,2), (HOLN(M,LL),LL=1,2),
     1                 J, TEMP2
420   FORMAT (I4, 3X, A4, A3, ' F FACTORS VS. SIGO ', 2A4,
     1        ', GROUP ', I3, ', ERROR = ', 1PE12.3)
430   IF (FF(K,J,L,N) .LT. 1.2) GO TO 450
      IF (FF(K,J,L,N) .GT. 0.0)  GO TO 500
450   ME = ME + 1
      IF (ME .GT. MAXE) GO TO 900
      WRITE (NOUT,490) ME, (HF(JKL,K), JKL=1,2), (HOLN(M,LL),LL=1,2),
     1        J, FF(K,J,L,N)
490   FORMAT (I4, 3X, A4, A3, ' F FACTORS IN ', 2A4, ', GROUP ', I3,
     1        ', OUTSIDE LIMITS  (', F6.4, ')')
500   CONTINUE
505   CONTINUE
510   DO 560  K=1,5
      NF = NPFF(K,M)
      IF (NF .LE. 0) GO TO 560
      DO 540  N=1,NF
      DO 540  J=NB,NE
      DO 540  L=2,NT(M)
      TEMP2 = FF(K,J,L-1,N) - FF(K,J,L,N)
      IF (TEMP2 .LE. EEPS) GO TO 540
      ME = ME + 1
      IF (ME .GT. MAXE) GO TO 900
      WRITE (NOUT, 530) ME, (HF(JKL,K),JKL=1,2), (HOLN(M,LL),LL=1,2),
     1                  J,TEMP2
530   FORMAT (I4, 3X, A4, A3, ' F FACTORS VS. TEMP ' ,
     1        2A4, ', GROUP ', I3, ', ERROR = ',1PE12.3)
540   CONTINUE
560   CONTINUE
900   RETURN
      END

      SUBROUTINE RCCSS (NXS, I1, SR, SM, SE, C, FSS)
      DIMENSION  NXS(1), I1(1), SR(8,1), SM(IGM,1), SE(IGM,1),
     1           C(ITL,IGM,1), FSS(5,NRCX,1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE CALCULATES RESONANCE SHIELDED CROSS SECTIONS
      LAP = .0
      LAPP = .0
      LAR = .0
      ALA = .0
      P02 = 0
      CVT = 0
      CNT = 0
      IF (R02 .LE. 0) GO TO 20
```

```
      DO 10  IIG=1,IGM
10    READ (NCR1)  ((C(L,IIG,M), L=1,ITL), M=1,MT)
      REWIND NCR1
20    ITEMP2 = 0
      DO 400  L=1,NRCM
      ITEMP1 = ITEMP2 + 2
      ITEMP2 = ITEMP1 + NXS(L) - 2
      DO 400  J=ITEMP1,ITEMP2
      M = I1(J)
      READ (NSR)  ((SR(K,IIG), K=1,8), IIG=1,IGM)
      READ (NSM)  ((SM(IIG,K), K=1,IGM), IIG=1,IGM)
      READ (NSM)  ((SE(IIG,K), K=1,IGM), IIG=1,IGM)
C     PATCH FOR REDUCED SCATTERING MATRIX
      DO 25 I=1,IGM
      JJ = I + NXCM
      IF (JJ+1 .GT. IGM) GO TO 30
      DO 25 JJJ=JJ+1, IGM
      SM(I,JJ) = SM(I,JJ) + SM(I,JJJ)
25    SE(I,JJ) = SE(I,JJ) + SE(I,JJJ)
30    CONTINUE
      DO 35 I=IGM, 1, -1
      JJ = I - NXCN
      IF (JJ-1 .LT. 1) GO TO 40
      DO 35 JJJ=1, JJ-1
      SM(I,JJ) = SM(I,JJ) + SM(I,JJJ)
35    SE(I,JJ) = SE(I,JJ) + SE(I,JJJ)
40    CONTINUE
      DO 400  IIG=1,IGM
      C(IHT-5,IIG,M) = SR(8,IIG)*FSS(5,M,IIG)
      C(IHT-4,IIG,M) = SR(7,IIG)
      C(IHT-3,IIG,M) = FSS(1,M,IIG)*SR(2,IIG)
      C(IHT-2,IIG,M) = C(IHT-3,IIG,M) + FSS(2,M,IIG)*SR(4,IIG)
      C(IHT-1,IIG,M) = SR(3,IIG)*C(IHT-3,IIG,M)
      C(IHT,IIG,M) = FSS(3,M,IIG)*SR(1,IIG)
      K1 = IIG - NXCM
      IF (K1 .LT. 1) K1 = 1
      K2 = IIG + NXCN
      IF (K2 .GT. IGM) K2 = IGM
      DO 50 K=K1,K2
      KK = IHS + IIG - K
      IF (KK .LE. IHT) GO TO 50
      C(KK,IIG,M) = SM(K,IIG) + FSS(4,M,K) * SE(K,IIG)
50    CONTINUE
      C(IHS,IIG,M) = C(IHT,IIG,M) - C(IHT-2,IIG,M)
     1             - (SR(5,IIG) - SM(IIG,IIG))
     2             - (SR(6,IIG) - SE(IIG,IIG)) * FSS(4,M,IIG)
400   CONTINUE
      REWIND NSR
      REWIND NSM
      RETURN
      END

      SUBROUTINE RCINP1 (HOLN,ATW,MAT,ICHI,NT,NGB,NGE,NPFF,FF,SF,
     1                   SR, SM, SE, TEM, ATEM, I0, I1, PSF,
     2                   NUT, MPUP)
      DIMENSION  HOLN(ML,1), ATW(1), NT(1), NGB(1), NGE(1), NPFF(5,1),
     1           FF(5,IGM,10,1), SF(5,ML,1), SR(8,1), SM(IGM,1),
     2           SE(IGM,1), TEM(10,1), ATEM(1), I0(1), I1(1),
     3           PSF(1),NUT(1),MPUP(1),MAT(1),ICHI(1)
$INCLUDE:'1DB.INC'
C     READ CROSS SECTION DATA IN THE RUSSIAN FORMAT
      IF (NUT(1) .LT. 0) GO TO 500
      CALL RCPUP  (HOLN, ATW, MAT, ICHI, NT, NGB, NGE, NPFF, FF, SF,
     1            SR, SM, SE, TEM, NUT, MPUP)
      GO TO 600
500   CALL RCPUP1 (HOLN, ATW, MAT, ICHI, NT, NGB, NGE, NPFF, FF, SF,
     1            SR, SM, SE, TEM, NUT, MPUP)
600   WRITE (NOUT,650)
650   FORMAT (//' MIXTURE SPECIFICATIONS FOR CALCULATING SHIELDED CROSS',
     1 ' SECTIONS'//
     2 '    MIXTURE      MIX        TEMPERATURE   POT. SCAT. FACTOR  ',
     3 ' NAME'/
     4 '    NUMBER     COMMAND        (DEG K)'/)
      DO 655  J=1,MM01
      ITEMP = I1(J)
      IF (ITEMP.NE.0)  GO TO 654
```

```
      WRITE (NOUT,657)  J, I0(J), I1(J), ATEM(J), PSF(J)
      GO TO 655
654   WRITE (NOUT,657)  J, I0(J), I1(J), ATEM(J), PSF(J),
     1                   (HOLN(ITEMP,LL),LL=1,2)
655   CONTINUE
657   FORMAT (5X, I5, 5X, I5, I8, 5X, 1PE11.3, 2X, 1PE11.3, 5X, 2A4)
      REWIND NFF
      REWIND NSR
      REWIND NSM
      RETURN
      END

      SUBROUTINE RCPRT1 (HOLN, SR, SM, SE, NPFF, SF, FF, NT, M, TEM)
      DIMENSION  HOLN(ML,1), SR(8,1), SM(IGM,1), SE(IGM,1), NPFF(5,1),
     1           SF(5,ML,1), FF(5,IGM,10,1), NT(1), TEM(10,1)
$INCLUDE:'1DB.INC'
      DIMENSION  HF(2,5)
      CHARACTER*4 HF
      DATA   HF(1,1), HF(2,1), HF(1,2), HF(2,2), HF(1,3), HF(2,3),
     1       HF(1,4), HF(2,4), HF(1,5), HF(2,5) /
     2       'FISS','ION ','CAPT','URE ',
     3       'TRAN','SPRT ','ELAS','TIC ','KERM','A   '/
C     PRINT DATA IN THE RUSSIAN FORMAT
      WRITE (NOUT,40)  M, (HOLN(M,L),L=1,2)
40    FORMAT (//I5, 5X, ' CROSS SECTION DATA FOR ',2A4,6X,11A6//)
      WRITE (NOUT,50)
50    FORMAT (' GROUP   SIGTR    SIGF     NU       SIGC     SIGIN  ',
     1        '   SIGEL    N,2N    KERMA'/)
      DO 60  IIG=1,IGM
60    WRITE (NOUT,70)  IIG, (SR(K,IIG),K=1,8)
70    FORMAT (I5, 1P8E9.2)
      DO 100 IIG=1,IGM
      WRITE (NOUT,80)  (HOLN(M,L),L=1,2), IIG
80    FORMAT (//' INELASTIC SCATTERING MATRIX FOR ',2A4/
     1        ' GROUP ',I4, ' TO GROUP 1, 2, ... IGM '/)
      WRITE (NOUT,90)  (SM(IIG,K), K=1,IGM)
90    FORMAT (1X,8F8.3)
100   CONTINUE
      DO 120 IIG=1,IGM
      WRITE (NOUT,110)  (HOLN(M,L),L=1,2), IIG
110   FORMAT (//' ELASTIC SCATTERING MATRIX FOR ',2A4/
     1        ' GROUP ',I4, ' TO GROUP 1, 2, ... IGM '/)
      WRITE (NOUT,90)  (SE(IIG,K), K=1,IGM)
120   CONTINUE
      DO 300  K=1,5
      NF = NPFF(K,M)
      IF (NF .LE. 0) GO TO 300
220   WRITE (NOUT,230)  (HF(KI,K),KI=1,2), (HOLN(M,L),L=1,2)
230   FORMAT (//A4,A3,' SHIELDING FACTORS FOR ',2A4,
     1        ' FOR EACH SIGO, GROUP, AND TEMPERATURE'/)
      WRITE (NOUT,260)  (SF(K,M,N),N=1,NF)
260   FORMAT (' SIGO ', 5X, 10E7.1)
      DO 280  IIG=NB,NE
      DO 280 J=1,NT(M)
280   WRITE (NOUT,290) IIG, TEM(J,M), (FF(K,IIG,J,N),N=1,NF)
290   FORMAT (I4, F6.0, 9F7.2, F6.2)
300   CONTINUE
      RETURN
      END

      SUBROUTINE RCPUP (HOLN,ATW,MAT,ICHI,NT,NGB,NGE,NPFF,FF,SF,
     1                  SR, SM, SE, TEM, NUT, MPUP)
      DIMENSION  HOLN(ML,1), ATW(1), NT(1), NGB(1), NGE(1), NPFF(5,1),
     1           FF(5,IGM,10,1), SF(5,ML,1), SR(8,1), SM(IGM,1),
     2           TEM(10,1),NUT(1),MPUP(1),MAT(1),ICHI(1),SE(IGM,1)
      CHARACTER*4 EOFS,ITEMP5
$INCLUDE:'1DB.INC'
      DATA EOFS/'EOFS'/
C     THIS SUBROUTINE READS DATA IN THE RUSSIAN FORMAT FROM TAPE
      NLAST = 1
      DO 10  M=1,ML
      IF (NUT(M) .GT. NLAST)  NLAST = NUT(M)
10    CONTINUE
      REWIND 15
      DO 25 I=1,ML
      ITEMP = 1000
```

```
      DO 20 J=1,ML
      IF (NUT(J) .GE. ITEMP) GO TO 20
      ITEMP = NUT(J)
      JJ = J
20    CONTINUE
      MPUP(I) = JJ
25    NUT(JJ) = NUT(JJ) + 1000
      DO 30 I=1,ML
30    NUT(I) = NUT(I) - 1000
      MTEMP = 1
      DO 600  NN=1,NLAST
      M = MPUP(MTEMP)
      IF (NUT(M) .EQ. NN) GO TO 35
      GO TO 500
35    MTEMP = MTEMP + 1
      READ (15) (HOLN(M,L),L=1,2), MAT(M)
      READ (15) ATW(M),ICHI(M),NT(M),NGB(M),NGE(M),
     1          (NPFF(K,M),K=1,5)
      READ (15) (TEM(K,M),K=1,NT(M))
      NB = NGB(M)
      NE = NGE(M)
      DO 100  K=1,5
      NF = NPFF(K,M)
      IF (NF .LE. 0) GO TO 100
      READ (15) (SF(K,M,N),N=1,NF)
      READ (15)   (((FF(K,I,J,N), N=1,NF), J=1,NT(M)), I=NB,NE)
      WRITE (NFF) (((FF(K,I,J,N), N=1,NF), J=1,NT(M)), I=NB,NE)
100   CONTINUE
      READ (15)    ((SR(K,IIG), K=1,8), IIG=1,IGM)
      WRITE (NSR)  ((SR(K,IIG), K=1,8), IIG=1,IGM)
      READ (15)   ((SM(IIG,K), K=1,IGM), IIG=1,IGM)
      WRITE (NSM) ((SM(IIG,K), K=1,IGM), IIG=1,IGM)
      READ (15)   ((SE(IIG,K), K=1,IGM), IIG=1,IGM)
      WRITE (NSM) ((SE(IIG,K), K=1,IGM), IIG=1,IGM)
      IF (MTEMP .GT. ML) GO TO 350
200   IF (NUT(MPUP(MTEMP)) .NE. NUT(MPUP(MTEMP-1))) GO TO 350
      MM = M
      M = MPUP(MTEMP)
      MTEMP = MTEMP + 1
      DO 210 L=1,2
210   HOLN(M,L) = HOLN(MM,L)
      MAT(M) = MAT(MM)
      ATW(M) = ATW(MM)
      ICHI(M) = ICHI(MM)
      NT(M) = NT(MM)
      NGB(M) = NGB(MM)
      NGE(M) = NGE(MM)
      DO 220 K=1,5
220   NPFF(K,M) = NPFF(K,MM)
      DO 230 K=1,NT(M)
230   TEM(K,M) = TEM(K,MM)
      NB = NGB(M)
      NE = NGE(M)
      DO 300 K = 1,5
      NF = NPFF(K,M)
      IF (NF .LE. 0) GO TO 300
      DO 250 N=1,NF
250   SF(K,M,N) = SF(K,MM,N)
      WRITE (NFF) (((FF(K,I,J,N), N=1,NF), J=1,NT(M)), I=NB,NE)
300   CONTINUE
      WRITE (NSR)  ((SR(K,IIG), K=1,8), IIG=1,IGM)
      WRITE (NSM) ((SM(IIG,K), K=1,IGM), IIG=1,IGM)
      WRITE (NSM) ((SE(IIG,K), K=1,IGM), IIG=1,IGM)
      IF (MTEMP .GT. ML) GO TO 610
      GO TO 200
350   IF (NPRT .LE. 1) GO TO 380
      CALL RCPRT1 (HOLN, SR, SM, SE, NPFF, SF, FF, NT, M, TEM)
380   CALL RCCHK (M, SR, HOLN, SM, SE, NT, NPFF, FF, NGB, NGE)
500   READ (15) ITEMP5
      IF (ITEMP5 .NE. EOFS) GO TO 500
600   CONTINUE
610   CONTINUE
      REWIND 15
      IF (NPRT .GT. 1) GO TO 640
      WRITE (NOUT,620) (M, (HOLN(M,L), L=1,2), NUT(M), M=1,ML)
620   FORMAT (//' CROSS SECTIONS FOR THE FOLLOWING ISOTOPES WERE READ',
```

```
     1       ' IN THE RUSSIAN FORMAT'//
     2       '  ISOTOPE             ISOTOPE              NUMBER  '/
     3       '  NUMBER                                   ON TAPE '//
     4        (I6, 17X, 2A4, 14X, I4, 6X))
640   RETURN
      END

      SUBROUTINE RCPUP1 (HOLN,ATW,MAT,ICHI,NT,NGB,NGE,NPFF,FF,SF,
     1                   SR, SM, SE, TEM, NUT, MPUP)
      DIMENSION  HOLN(ML,1), ATW(1), NT(1), NGB(1), NGE(1), NPFF(5,1),
     1           FF(5,IGM,10,1), SF(5,ML,1), SR(8,1), SM(IGM,1),
     2           TEM(10,1),NUT(1),MPUP(1),MAT(1),ICHI(1),SE(IGM,1)
      CHARACTER*4 EOFS,ITEMP5
      CHARACTER*10 FNAME
$INCLUDE:'1DB.INC'
      DATA EOFS/'EOFS'/
C     THIS SUBROUTINE READS DATA IN THE RUSSIAN FORMAT FROM TAPE
      M = 0
10    M = M + 1
      IF (N .GT. ML) GO TO 600
      MPUP(M) = M
      FNAME(1:10) = 'XS0000.DAT'
      NN = - NUT(M)
      WRITE (FNAME(3:6), 20) NN
20    FORMAT (I4.4)
      CLOSE (15)
      OPEN (15, FILE=FNAME, FORM='UNFORMATTED')
      READ (15) (HOLN(M,L),L=1,2), MAT(M)
      READ (15) ATW(M),ICHI(M),NT(M),NGB(M),NGE(M),
     1          (NPFF(K,M),K=1,5)
      READ (15) (TEM(K,M),K=1,NT(M))
      NB = NGB(M)
      NE = NGE(M)
      DO 100  K=1,5
      NF = NPFF(K,M)
      IF (NF .LE. 0) GO TO 100
      READ (15) (SF(K,M,N),N=1,NF)
      READ (15)   (((FF(K,I,J,N), N=1,NF), J=1,NT(M)), I=NB,NE)
      WRITE (NFF) (((FF(K,I,J,N), N=1,NF), J=1,NT(M)), I=NB,NE)
100   CONTINUE
      READ (15)    ((SR(K,IIG), K=1,8), IIG=1,IGM)
      WRITE (NSR)  ((SR(K,IIG), K=1,8), IIG=1,IGM)
      READ (15)   ((SM(IIG,K), K=1,IGM), IIG=1,IGM)
      WRITE (NSM) ((SM(IIG,K), K=1,IGM), IIG=1,IGM)
      READ (15)   ((SE(IIG,K), K=1,IGM), IIG=1,IGM)
      WRITE (NSM) ((SE(IIG,K), K=1,IGM), IIG=1,IGM)
350   IF (NPRT .LE. 1) GO TO 380
      CALL RCPRT1 (HOLN, SR, SM, SE, NPFF, SF, FF, NT, M, TEM)
380   CALL RCCHK (M, SR, HOLN, SM, SE, NT, NPFF, FF, NGB, NGE)
      GO TO 10
600   IF (NPRT .GT. 1) GO TO 640
      WRITE (NOUT,620) (M, (HOLN(M,L), L=1,2), NUT(M), M=1,ML)
620   FORMAT (//' CROSS SECTIONS FOR THE FOLLOWING ISOTOPES WERE READ',
     1       ' IN THE RUSSIAN FORMAT'/
     1       ' (NEG. TAPE NUMBER INDICATES FILE NAME XS----.DAT,',
     1       ' WHERE ---- = ABS(NUM))'//
     2       '  ISOTOPE             ISOTOPE              TAPE    '/
     3       '  NUMBER                                   NUMBER  '//
     4        (I6, 17X, 2A4, 14X, I4, 6X))
640   RETURN
      END

      SUBROUTINE RCSTUP (NPFF,NT,FF,TEM,ATEM,NGB,NGE,I1,SR,SM,SE,MPUP)
      DIMENSION  NPFF(5,1), NT(1), FF(5,IGM,10,1), TEM(10,1), ATEM(1),
     1           NGB(1),NGE(1),I1(1),SR(8,1),SM(IGM,1),SE(IGM,1),MPUP(1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE COMPUTES F-FACTORS FOR TEMP. DEPENDENT MATERIALS
C     AND PUTS THE CROSS SECTION DATA ON TAPE IN THE PROPER ORDER
      DO 200   J=1,MM01
      IF (I1(J) .LE. 0) GO TO 200
      MM = I1(J)
      DO 85  MK=1,ML
      M = MPUP(MK)
      NB = NGB(M)
      NE = NGE(M)
      DO 80  K=1,5
```

```
      NF = NPFF(K,M)
      IF (NF .LE. 0) GO TO 80
      READ (NFF) (((FF(K,JJ,L,N),N=1,NF),L=1,NT(M)),JJ=NB,NE)
80    CONTINUE
      READ (NSR)  ((SR(K,IIG), K=1,8), IIG=1,IGM)
      READ (NSM)  ((SM(IIG,K), K=1,IGM), IIG=1,IGM)
      READ (NSM)  ((SE(IIG,K), K=1,IGM), IIG=1,IGM)
      IF (M .EQ. MM) GO TO 90
85    CONTINUE
90    CONTINUE
      DO 94 L=2,NT(M)
      LL = L
      IF (ATEM(J) .LE. TEM(L,M)) GO TO 96
94    CONTINUE
96    CONTINUE
      DO 150  K=1,5
      NF = NPFF(K,M)
      IF (NF .LE. 0) GO TO 150
      DO 110 IIG=NB,NE
      DO 110 N=1,NF
      AA = FF(K,IIG,LL-1,N)
      BB = FF(K,IIG,LL,N)
      DD1=ALOG(TEM(LL,M)/TEM(LL-1,M))
      EE1=ALOG(ATEM(J)/TEM(LL-1,M))
      FF(K,IIG,1,N)= AA + (BB - AA)*EE1/DD1
110   CONTINUE
      WRITE (NSCRAT)  ((FF(K,IIG,1,N), N=1,NF), IIG=NB,NE)
150   CONTINUE
      WRITE (NSCR1) ((SR(K,IIG), K=1,8), IIG=1,IGM)
      WRITE (NSCR2) ((SM(IIG,K), K=1,IGM), IIG=1,IGM)
      WRITE (NSCR2) ((SE(IIG,K), K=1,IGM), IIG=1,IGM)
      REWIND NFF
      REWIND NSR
      REWIND NSM
200   CONTINUE
      REWIND NSCRAT
      REWIND NSCR1
      REWIND NSCR2
C     SWITCH TAPE DESIGNATIONS
      ITEMP = NFF
      ITEMP1 = NSR
      ITEMP2 = NSM
      NFF = NSCRAT
      NSR = NSCR1
      NSM = NSCR2
      NSCRAT = ITEMP
      NSCR1 = ITEMP1
      NSCR2 = ITEMP2
      RETURN
      END

      SUBROUTINE REAG2 (HOLL, ARRAY, NCOUNT)
      COMMON    NCR1, NFF, NINP, NOUT, NSCRAT, NSCR1, NSCR2
      DIMENSION ARRAY(1), HOL(80), HE(40), LE(40)
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      J=0
1     READ (NINP, 10) (HOL(I),I=1,80)
10    FORMAT (80A1)
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF (I.LE.80) GO TO 50
45    IF (LE(L) .EQ. 0) L=L-1
      GO TO 100
50    IF (HOL(I).EQ.' ') GO TO 60
      IF (HOL(I).EQ.'T') GO TO 55
      IF (HOL(I).EQ.'/') GO TO 45
      IF (HOL(I).EQ.'F') GO TO 55
      IF (HOL(I).EQ.'R') GO TO 55
      IF (HOL(I).EQ.'I') GO TO 55
      IF (HOL(I).EQ.'C') GO TO 55
```

```
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
 55   IF (LE(L).GT.0) L=L+1
      LE(L)=1
      HE(L)(1:1)= HOL(I)
      IF (HOL(I) .EQ. 'C') GO TO 30
      IF (HOL(I).EQ.'R' .OR. HOL(I).EQ. 'I') GO TO 30
      GO TO 100
 60   IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
 100  LL=L
      L=0
 110  L=L+1
      IF (L .GT. LL) GO TO 1
      IF (HE(L)(1:1).EQ. 'T') GO TO 150
      IF (HE(L)(1:1).EQ. 'F') GO TO 120
      IF (HE(L)(1:1).EQ. 'R') GO TO 130
      IF (HE(L)(1:1).EQ. 'I') GO TO 140
      IF (HE(L)(1:1).EQ. 'C') GO TO 115
      J=J+1
      READ (HE(L)(1:LE(L)),112) ARRAY(J)
 112  FORMAT (E20.2)
      GO TO 110
C     CYCLE
 115  READ (HE(L+1)(1:LE(L+1)),132) J1
      READ (HE(L+2)(1:LE(L+2)),132) J2
      J0=J
      DO 119 K1=1,J1
      DO 119 K2=1,J2
      J=J+1
 119  ARRAY(J) = ARRAY(J0-J2+K2)
      L=L+2
      GO TO 110
C     FILL
 120  DO 125 JJ=J+1,NCOUNT
 125  ARRAY(JJ)=ARRAY(J)
      J=NCOUNT
      GO TO 150
C     REPEAT
 130  READ (HE(L+1)(1:LE(L+1)),132) J1
 132  FORMAT (I20)
      READ (HE(L+2)(1:LE(L+2)),112) T1
      DO 135 JJ=J+1,J+J1
 135  ARRAY(JJ)=T1
      J=J+J1
      L=L+2
      GO TO 110
C     INTERPOLATE
 140  READ (HE(L+1)(1:LE(L+1)),132) J1
      READ (HE(L+2)(1:LE(L+2)),112) ARRAY(J+J1+1)
      T1= (ARRAY(J+J1+1) - ARRAY(J))/(J1+1)
      DO 145 JJ=J+1,J+J1
 145  ARRAY(JJ)= ARRAY(JJ-1) + T1
      J=J +J1+1
      L=L+2
      GO TO 110
 150  IF (HOLL .EQ. '   INP') GO TO 155
      IF (HOLL .EQ. '  INPB') GO TO 155
      IF (HOLL .EQ. '  RECS') GO TO 155
      WRITE (NOUT,160) HOLL,J,(ARRAY(I),I=1,J)
 155  IF (J-NCOUNT) 170,180,170
 160  FORMAT (6X,A6,I6/(6E12.5))
 170  CALL ERRO2 ( ' REAG2',170,1)
 180  RETURN
      END

      SUBROUTINE REAI2 (HOLL,IARRAY,NCOUNT)
      COMMON      NCR1, NFF, NINP, NOUT, NSCRAT, NSCR1, NSCR2
      DIMENSION IARRAY (1), HOL(80),HE(40),LE(40)
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      J=0
 1    READ (NINP, 10) (HOL(I),I=1,80)
 10   FORMAT (80A1)
```

```
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF (I.LE.80) GO TO 50
45    IF (LE(L) .EQ. 0) L=L-1
      GO TO 100
50    IF (HOL(I).EQ.' ') GO TO 60
      IF (HOL(I).EQ.'T') GO TO 55
      IF (HOL(I).EQ.'/') GO TO 45
      IF (HOL(I).EQ.'F') GO TO 55
      IF (HOL(I).EQ.'R') GO TO 55
      IF (HOL(I).EQ.'I') GO TO 55
      IF (HOL(I).EQ.'C') GO TO 55
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
55    IF (LE(L).GT.0) L=L+1
      LE(L)=1
      HE(L)(1:1)= HOL(I)
      IF (HOL(I) .EQ. 'C') GO TO 30
      IF (HOL(I).EQ.'R' .OR. HOL(I).EQ. 'I') GO TO 30
      GO TO 100
60    IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
100   LL=L
      L=0
110   L=L+1
      IF (L .GT. LL) GO TO 1
      IF (HE(L)(1:1).EQ.'T') GO TO 150
      IF (HE(L)(1:1).EQ.'C') GO TO 116
      IF (HE(L)(1:1).EQ. 'F') GO TO 120
      IF (HE(L)(1:1).EQ. 'R') GO TO 130
      IF (HE(L)(1:1).EQ.'I') GO TO 140
      J=J+1
      READ (HE(L)(1:LE(L)),115) IARRAY(J)
115   FORMAT (I20)
      GO TO 110
C     CYCLE
116   READ (HE(L+1)(1:LE(L+1)),115) J1
      READ (HE(L+2)(1:LE(L+2)),115) J2
      J0=J
      DO 119 K1=1,J1
      DO 119 K2=1,J2
      J=J+1
119   IARRAY(J) = IARRAY(J0-J2+K2)
      L=L+2
      GO TO 110
C     FILL
120   DO 125 JJ=J+1,NCOUNT
125   IARRAY(JJ)=IARRAY(J)
      J=NCOUNT
      GO TO 150
C     REPEAT
130   READ (HE(L+1)(1:LE(L+1)),132) J1
132   FORMAT (I20)
      READ (HE(L+2)(1:LE(L+2)),115) I1
      DO 135 JJ=J+1,J+J1
135   IARRAY(JJ)=I1
      J=J+J1
      L=L+2
      GO TO 110
C     INTERPOLATE
140   READ (HE(L+1)(1:LE(L+1)),132) J1
      READ (HE(L+2)(1:LE(L+2)),115) IARRAY(J+J1+1)
      I1= (IARRAY(J+J1+1) - IARRAY(J))/(J1+1)
      DO 145 JJ=J+1,J+J1
145   IARRAY(JJ)= IARRAY(JJ-1) + I1
      J=J +J1+1
      L=L+2
      GO TO 110
150   IF (HOLL.EQ.'   INP') GO TO 155
      IF (HOLL .EQ. '  INPB') GO TO 155
      WRITE (NOUT,160) HOLL,J,(IARRAY(I),I=1,J)
```

```
155   IF (J-NCOUNT) 170,180,170
160   FORMAT (6X,A6,I6/(10I6))
170   CALL ERRO2 (' REAI2',170,1)
180   RETURN
      END

      SUBROUTINE REAIG2 (HOLL, IIVS, IVS, AVS, IIV, IV, AV, NX, NCOUNT)
      COMMON  NCR1, NFF, NINP, NOUT
      DIMENSION  IIV(1), AV(1), IV(1), HOL(80), HE(40), LE(40)
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      IIVS = NX
      IVS = 0
      AVS = .0
      J=0
1     READ (NINP, 10) (HOL(I),I=1,80)
10    FORMAT (80A1)
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF(I.LE.80) GO TO 50
45    IF (LE(L) .EQ. 0) L=L-1
      GO TO 100
50    IF (HOL(I).EQ. ' ') GO TO 60
      IF (HOL(I).EQ. 'T') GO TO 55
      IF (HOL(I).EQ. '/') GO TO 45
      IF (HOL(I).EQ. 'C') GO TO 55
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
55    IF (LE(L).GT.0) L=L+1
      LE(L)=1
      HE(L)(1:1)= HOL(I)
      IF (HOL(I) .EQ. 'C') GO TO 30
      GO TO 100
60    IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
100   LL=L
      L=0
110   L=L+1
      IF (L .GT. LL) GO TO 1
      IF (HE(L)(1:1).EQ. 'T') GO TO 150
      IF (HE(L)(1:1).EQ. 'C') GO TO 116
      J=J+1
      J3 = J - 2*(J/2)
      IF (J3 .EQ. 0) READ (HE(L)(1:LE(L)),112) AV(J/2)
112   FORMAT (E20.6)
      IF (J3 .EQ. 1) READ (HE(L)(1:LE(L)),115) IV((J+1)/2)
115   FORMAT (I20)
      GO TO 110
C     CYCLE
116   READ (HE(L+1)(1:LE(L+1)),115) J1
      READ (HE(L+2)(1:LE(L+2)),115) J2
      J0=J
      DO 119 K1=1,J1
      DO 119 K2=1,J2
      J=J+1
      J3 = J - 2*(J/2)
      IF (J3 .EQ. 0) AV(J/2) = AV((J0-J2+K2)/2)
      IF (J3 .EQ. 1) IV((J+1)/2) = IV((J0-J2+K2+1)/2)
119   CONTINUE
      L=L+2
      GO TO 110
150   NCOUNT = J/2
      WRITE (NOUT, 160) HOLL, NX, NCOUNT, (IV(I), AV(I), I=1,NCOUNT)
160   FORMAT (6X, A6, ' (MAT. #, DENSITY) FOR MATERIAL ', I6,
     1        ', ', I6, ' VALUES'/(4(I6, E12.4)))
      IF ((J - 2*(J/2)) .EQ. 0) GO TO 180
170   CALL ERRO2 (' REAIG2',170,1)
180   DO 190 N=1,NCOUNT
190   IIV(N) = NX
      RETURN
```

```
      END

      SUBROUTINE REGC2 (A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12)
      COMMON NCR1, NFF, NINP, NOUT
      DIMENSION HE(40), LE(40), HOL(80)
      DIMENSION AV(12), AVC(12)
      CHARACTER*6 AVC
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      DATA AVC/'EV', 'EVM', 'S03', 'BUCK', 'LAL', 'LAH', 'EPS',
     1         'EPSA', 'POD', 'ORF', 'S01', 'TMAX'/
C     INPUT OF NAMED FLOATING PARAMETERS
      DO 2 I=1,12
2     AV(I) = .0
      AV(5) = .001
      AV(6) = .1
      AV(7) = .0001
      AV(8) = .001
      AV(9) = 1.0
      AV(10) = 1.3
      AV(11) = 1.0
      J=0
5     READ (NINP, 10) (HOL(I),I=1,80)
10    FORMAT (80A1)
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF(I.LE.80) GO TO 50
45    IF (LE(L) .EQ. 0) L=L-1
      GO TO 100
50    IF (HOL(I).EQ.' ') GO TO 60
      IF (HOL(I).EQ.'=') GO TO 60
      IF (HOL(I).EQ.',') GO TO 60
      IF (HOL(I).EQ.'/') GO TO 45
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
60    IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
100   LL=L
      L=0
110   L=L+1
      IF (L .GT. LL) GO TO 5
      J=J+1
      IF ((HE(L)(1:1).EQ. 'T') .AND. (LE(L) .EQ. 1)) GO TO 150
      J3 = J - 2*(J/2)
      IF (J3 .EQ. 1) GO TO 120
      READ (HE(L)(1:LE(L)),112) AV(KK)
112   FORMAT (E20.6)
      GO TO 110
120   DO 130 K=1,12
      KK = K
      IF (HE(L)(1:LE(L)) .EQ. AVC(K)(1:LE(L))) GO TO 110
130   CONTINUE
135   CALL ERRO2 (' REGC2', 135, 1)
150   IF ((J - 2*(J/2)) .EQ. 1) GO TO 180
170   CALL ERRO2 (' REGC2', 170, 1)
180   CONTINUE
      A1=AV(1)
      A2=AV(2)
      A3=AV(3)
      A4=AV(4)
      A5=AV(5)
      A6=AV(6)
      A7=AV(7)
      A8=AV(8)
      A9=AV(9)
      A10=AV(10)
      A11=AV(11)
      A12=AV(12)
      RETURN
      END
```

```
      SUBROUTINE REIC2 ( I1, I2, I3, I4, I5, I6, I7, I8, I9,I10,
     1                   I11,I12,I13,I14,I15,I16,I17,I18,I19,I20,
     2                   I21,I22,I23,I24,I25,I26,I27)
      COMMON NCR1, NFF, NINP, NOUT
      DIMENSION HE(40), LE(40), HOL(80)
      DIMENSION IV(27), IVC(27)
      CHARACTER*6 IVC
      CHARACTER*6 HOLL
      CHARACTER*1 HOL
      CHARACTER*20 HE
      DATA IVC/'A02', 'NEV', 'S02',  'IGM',  'NXCM', 'NXCN', 'NST',
     1         'ML',  'NFX', 'NPRT', 'NPUN', 'NOIT', 'NRCM', 'NMIX',
     2         'IGE', 'IM',  'IZM',  'B01',  'B02',  'NACT',
     3         'NCG', 'NCCM','NCCN', 'NCM',  'NOXS',  'NIXS', 'MAXE'/
C     INPUT OF NAMED INTEGER PARAMETERS
      DO 2 I=1,27
2     IV(I) = 0
      IV(2) = 1
      J=0
5     READ (NINP, 10) (HOL(I),I=1,80)
10    FORMAT (80A1)
      DO 20 L=1,40
20    LE(L)=0
      I=0
      L=0
30    L=L+1
40    I=I+1
      IF(I.LE.80) GO TO 50
45    IF (LE(L) .EQ. 0) L=L-1
      GO TO 100
50    IF (HOL(I).EQ.' ') GO TO 60
      IF (HOL(I).EQ.'=') GO TO 60
      IF (HOL(I).EQ.',') GO TO 60
      IF (HOL(I).EQ.'/') GO TO 45
      LE(L)=LE(L)+1
      HE(L)(LE(L):LE(L))=HOL(I)
      GO TO 40
60    IF (LE(L) .EQ. 0) GO TO 40
      GO TO 30
100   LL=L
      L=0
110   L=L+1
      IF (L .GT. LL) GO TO 5
      J=J+1
      IF ((HE(L)(1:1).EQ. 'T') .AND. (LE(L) .EQ. 1)) GO TO 150
      J3 = J - 2*(J/2)
      IF (J3 .EQ. 1) GO TO 120
      READ (HE(L)(1:LE(L)),112) IV(KK)
112   FORMAT (I20)
      GO TO 110
120   DO 130 K=1,27
      KK = K
      IF (HE(L)(1:LE(L)) .EQ. IVC(K)(1:LE(L))) GO TO 110
130   CONTINUE
135   CALL ERRO2 (' REIC2', 135, 1)
150   IF ((J - 2*(J/2)) .EQ. 1) GO TO 180
170   CALL ERRO2 (' REIC2', 170, 1)
180   CONTINUE
      I1=IV(1)
      I2=IV(2)
      I3=IV(3)
      I4=IV(4)
      I5=IV(5)
      I6=IV(6)
      I7=IV(7)
      I8=IV(8)
      I9=IV(9)
      I10=IV(10)
      I11=IV(11)
      I12=IV(12)
      I13=IV(13)
      I14=IV(14)
      I15=IV(15)
      I16=IV(16)
      I17=IV(17)
      I18=IV(18)
```

```
      I19=IV(19)
      I20=IV(20)
      I21=IV(21)
      I22=IV(22)
      I23=IV(23)
      I24=IV(24)
      I25=IV(25)
      I26=IV(26)
      I27=IV(27)
      RETURN
      END

      SUBROUTINE RECS (C, ATW, HOLN)
$INCLUDE:'1DB.INC'
      DIMENSION  ATW(1), HOLN(ML,1)
      DIMENSION  C(ITL,IGM,1)
C     C(ITL,IGM,MT) CROSS SECTION ARRAY--STARTS AT A(LN2)
C     READ AND CHECK CROSS SECTIONS, PERFORM ADJOINT REVERSALS IF
C     REQUIRED, AND WRITE CROSS SECTION TAPE
      IF (NRCM .GT. 0) GO TO 160
      WRITE (NOUT,5)  (ID(I), I=1,20)
5     FORMAT (//1X,20A4/)
      WRITE (NOUT,10)
10    FORMAT (' THE FOLLOWING NUCLIDES ARE IN THE DTF FORMAT'/)
      DO 50  I=1,ML
      IF (NIXS .EQ. 0) GO TO 30
      IF (NIXS .EQ. 2) GO TO 40
      READ (15) (HOLN(I,L),L=1,2)
      READ (15) ATW(I)
      DO 20 IIG=1,IGM
20    READ (15) (C(L,IIG,I), L=1,ITL)
      GO TO 50
30    READ (NINP, 32)  (HOLN(I,L),L=1,2)
32    FORMAT (2A4)
      CALL REAG2 ('  RECS', ATW(I), 1)
      DO 35  IIG=1,IGM
35    CALL REAG2 ('  RECS', C(1,IIG,I), ITL)
      GO TO 50
40    READ (15, 32) (HOLN(I,L),L=1,2)
      READ (15, 42) ATW(I)
42    FORMAT (6E12.3)
      DO 45 IIG=1,IGM
45    READ (15,42) (C(L,IIG,I), L=1,ITL)
50    WRITE (NOUT, 60) I, (HOLN(I,L),L=1,2)
60    FORMAT (I3, 6X, 2A4)
      IF (NIXS .EQ. 0) GO TO 80
      REWIND 15
C     CHECK ON CROSS SECTION CONSISTENCY AND ORDER
80    ITEMP = 0
      DO 140  J=1,ML
      DO 140  I=1,IGM
      G = C(IHT-2,I,J)
      K1 = - NXCN
      K2 = NXCM
      DO 110  K = K1,K2,1
      KK = I + K
      M = IHS + K
      IF (KK .LE. 0)    GO TO 110
      IF (KK .GT. IGM)  GO TO 110
      G = G + C(M,KK,J)
110   CONTINUE
      IF (C(IHT,I,J) .LE. .0)  GO TO 140
      G = 100. * ABS ((G - C(IHT,I,J))/C(IHT,I,J))
      IF (G .LT. 1.0) GO TO 140
      ITEMP = ITEMP + 1
      WRITE (NOUT, 135) ITEMP, J, I, G
135   FORMAT (' ',I3, 3X, ' CHECK MATERIAL ',I3, 3X, ' GROUP ', I3,
     1        5X, '  ERR%= ', G10.4)
      IF (ITEMP .GE. MAXE) GO TO 160
140   CONTINUE
C     A02=0/1=FLUX CALCULATION/ADJOINT CALCULATION
160   IF (A02 .EQ. 0) GO TO 280
      DO 190  IIG=1,IGM
      IGBAR=IGM-IIG+1
      DO 180  M=1,MT
      DO 180  L = 1,IHT
```

```
      TEMP=C(L,IIG,M)
      C(L,IIG,M)=C(L,IGBAR,M)
180   C(L,IGBAR,M)=TEMP
      IF (IGBAR .LE. IIG+1) GO TO 200
190   CONTINUE
200   K1 = IHT - IHS + 1
      K2 = ITL - IHS
      DO 240  M = 1,MT
      DO 240  IIG = 1,IGM
      IGBAR = IGM - IIG + 1
      DO 240  L = K1,K2,1
      IF (L .GE. IIG) GO TO 240
      I = L + IHS
      ITEMP = IGBAR + L
      IF (ITEMP .LE. 0)   GO TO 240
      IF (IIG .GT. ITEMP) GO TO 240
      TEMP = C(I, IIG, M)
      C(I,IIG,M) = C(I,ITEMP,M)
      C(I,ITEMP,M) = TEMP
240   CONTINUE
C     WRITE CROSS SECTION TAPE
280   DO 300  IIG=1,IGM
300   WRITE (NCR1) ((C(L,IIG,M), L=1,ITL), M=1,MT)
      REWIND NCR1
      RETURN
      END

      SUBROUTINE GRAM (MASS, VOL, ATW, HOLN, M0, M2, V0,
     1                 I0, I1, I2, I3)
$INCLUDE:'1DB.INC'
      DIMENSION MASS(ML,1), VOL(1), ATW(1), HOLN(ML,1), M0(1),
     1          M2(1), V0(1), I0(1), I1(1), I2(1), I3(1)
C     CALCULATE MATERIAL INVENTORIES
      WRITE (NOUT,10) (ID(I), I=1,20)
10    FORMAT (//1X,20A4/)
      WRITE (NOUT,20)
20    FORMAT (/' MATERIAL INVENTORY (KILOGRAMS) FOR EACH ZONE '/)
      CALL CLEAR (0.0,VOL,IZM)
      ITEMP = ML*IZM
      CALL CLEAR (0.0,MASS,ITEMP)
      DO 30  I = 1, IM
      K = M0(I)
30    VOL(K) = VOL(K) + V0(I)*.001
      DO 39  M=1,M01
      I3(M) = I2(M)
      IF (I0(M) - I1(M))  39,35,39
35    IF (I2(M))  39,36,39
36    DO 38  MM=1,M
      IF (I0(M) - I0(MM))  38,37,38
37    I3(MM) = I2(MM)*EV
38    CONTINUE
39    CONTINUE
      DO 190 N =1, IZM
      NN = M2(N)
      DO 190 M = 1,M01
      IF (I0(M) - NN)  190, 40, 190
40    L = I1(M)
48    IF (L - ML)  170, 170, 50
50    NNAA = L
      IF (L - I0(M))  130,190, 130
130   DO 160 MAA = 1, M01
      IF (I0(MAA) - NNAA)  160, 140, 160
140   L = I1(MAA)
148   IF (L)  160, 160, 150
150   E01 = I3(MAA)*I3(M)
      MASS(L,N) = ((E01*ATW(L)*VOL(N))/.6023) + MASS(L,N)
160   CONTINUE
      GO TO 190
170   IF (L)  190, 190, 180
180   E01 = I3(M)
      MASS(L,N) = ((E01*ATW(L)*VOL(N)) .6023) + MASS(L,N)
190   CONTINUE
      DO 270 L = 1, IZM, 5
      LL = L + 4
      IF (LL - IZM) 210, 210, 200
200   LL = IZM
```

```
210   WRITE (NOUT,220) ((K), K=L,LL)
220   FORMAT (//' MATERIAL   ATOMIC WT.',5(4X,'ZONE',I3,:))
      WRITE (NOUT,230)
230   FORMAT ()
      DO 250 K = 1, ML
250   WRITE (NOUT,260)K,(HOLN(K,I),I=1,2),ATW(K),(MASS(K,I),I=L,LL)
260   FORMAT (1X,I3,1X,2A4,F8.2,1X,5E11.3)
      IF (LL - IZM)  270, 280, 280
270   CONTINUE
C     COMPUTE TOTAL MASSES
280   WRITE (NOUT, 275)
275   FORMAT (//' MATERIAL   ATOMIC WT.    TOTAL '/)
      DO 310 K=1,ML
      TEMP = .0
      DO 300 L=1,IZM
300   TEMP = TEMP + MASS(K,L)
310   WRITE (NOUT, 260) K, (HOLN(K,N), N=1,2), ATW(K), TEMP
      WRITE (NOUT,350)
350   FORMAT (//' ZONE NUMBER    VOLUME (LITERS)'/)
      DO 400 L=1,IZM
      WRITE (NOUT,360) L, VOL(L)
360   FORMAT (6X,I4,6X,1PE12.3)
400   CONTINUE
      RETURN
      END

      SUBROUTINE AVERAG (PHIB,AXS,FXS,TXS,MATN,MASS,ATW,VOL,CO,N2,MO,VO,
     1                   HOLN, NBR)
      DIMENSION PHIB(1), AXS(ML,1), FXS(ML,1), MATN(1), MASS(ML,1),
     1          ATW(1), VOL(1), CO(ITL,1), N2(IM,1), MO(1), VO(1),
     2          HOLN(ML,1), NBR(1), TXS(ML,1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE CALCULATES ZONE AVERAGED FLUXES, FISSION CROSS
C     SECTIONS, AND ABSORPTION CROSS SECTIONS.
      RL = 0.0
      RC = 0.0
      DO 10  KZ=1,IZM
      PHIB(KZ) = 0.0
      DO 10  KN =1,NCON
      AXS(KN,KZ) = 0.0
      FXS(KN,KZ) = 0.0
      TXS(KN,KZ) = .0
      LN = MATN(KN)
      IF (MASS(LN,KZ) .EQ. 0) GO TO 10
      MASS(LN,KZ) = (MASS(LN,KZ)*.6023)/(ATW(LN)*VOL(KZ))
10    CONTINUE
      DO 100  IIG=1,IGM
      READ (NCR1)  ((CO(II,J), II=1,ITL),J=1,MT)
      DO 100  I=1,IM
      KZ = MO(I)
      PHIB(KZ) = PHIB(KZ) + N2(I,IIG)*VO(I)
      DO 100  KN=1,NCON
      LN = MATN(KN)
      TXS(KN,KZ) = TXS(KN,KZ) + CO(IHT-4,LN)*N2(I,IIG)*VO(I)
      AXS(KN,KZ) = AXS(KN,KZ) + CO(IHT-2,LN)*N2(I,IIG)*VO(I)
100   FXS(KN,KZ) = FXS(KN,KZ) + CO(IHT-3,LN)*N2(I,IIG)*VO(I)
      DO 200  KZ=1,IZM
      TEMP3 = PHIB(KZ)
      IF (VOL(KZ) .EQ. .0) GO TO 105
      PHIB(KZ) = PHIB(KZ)/(VOL(KZ)*1000.)
105   CONTINUE
      WRITE (NOUT,110)  KZ, PHIB(KZ), VOL(KZ)
110   FORMAT (//'    Z O N E ', I3, 5X, ' FLUX =', 1PE10.4,
     1        5X, ' VOLUME =', 1PE10.4, ' LITERS'/)
      WRITE (NOUT,120)
120   FORMAT (1X,'BURNABLE MAT.    NAME       ATOM  ',
     1        3X, 'FISSION    ABSORPTION   SIGMA     SIGMA '/
     2        1X, 'ISOTOPE  NO. ', 15X,'DENSITY',4X,'RATE',8X,'RATE',
     3        4X, 'FISSION',2X,'ABSORPTION'//)
      DO 200  KN=1,NCON
      LN = MATN(KN)
      TEMP1 = AXS(KN,KZ)*MASS(LN,KZ)
      TEMP2 = FXS(KN,KZ)*MASS(LN,KZ)
      IF (TEMP3 .EQ. .0) GO TO 125
      AXS(KN,KZ) = AXS(KN,KZ)/TEMP3
      FXS(KN,KZ) = FXS(KN,KZ)/TEMP3
```

```
      TXS(KN,KZ) = TXS(KN,KZ)/TEMP3
125   CONTINUE
      WRITE (NOUT,130)  KN, LN, (HOLN(LN,K),K=1,2), MASS(LN,KZ), TEMP2,
     1                  TEMP1, FXS(KN,KZ), AXS(KN,KZ)
130   FORMAT (1X, I3, 5X, I3, 4X, 2A4, 1P5E11.2)
      ITEMP = NBR(KN)
      IF (ITEMP - 1)  200, 140, 160
140   RC = RC + TEMP1 - TEMP2
      GO TO 200
160   RL = RL + TEMP1
200   CONTINUE
      IF (RL .EQ. .0) GO TO 360
      TEMP = RC/RL
      WRITE (NOUT,350)  TEMP
350   FORMAT (//' BREEDING RATIO        =', F7.4)
360   REWIND NCR1
      RETURN
      END

      SUBROUTINE INPB (MATN, NBR, NID, NIN, NIC, NIF,
     1                 IND, INN, INC, INF,
     2                 YDN, YNN, YCN, YFN, ALAM, HOLN)
$INCLUDE:'1DB.INC'
      DIMENSION  MATN(1), NBR(1), ALAM(1), HOLN(ML,1),
     2           YDN(1), YNN(1), YCN(1), YFN(1),
     3           NID(1), NIN(1), NIC(1), NIF(1),
     4           IND(1), INN(1), INC(1), INF(1)
C     THIS SUBROUTINE READS AND PRINTS THE BURNUP DATA
10    CALL REAG2 ('  INPB',TEMP,1)
      DELT = .0
      IF (TEMP .LT. .0) DELT = -TEMP
      DAY = DAY + DELT
      CVT = 0
      CNT = 0
      P02 = 0
      ALA = 0.0
      LAP = 0.0
      LAPP = 0.0
      LAR = 0.0
      KPAGE = 100
      IF (TEMP)  300, 15, 20
15    NCON = INT (TEMP + .001)
      GO TO 300
20    NCON = INT (TEMP + .001)
      IF (NCON .GT. ML) CALL ERRO2 ('  INPB', 20, 1)
      WRITE (NOUT,24)
24    FORMAT (//' BURNUP DATA'//)
      WRITE (NOUT,25)
25    FORMAT (/' MATERIAL SEQUENCE NUMBERS FOR BURNABLE ISOTOPES')
      CALL REAI2 ('  MATN', MATN(1), NCON)
      WRITE (NOUT,30)
30    FORMAT (/' DECAY CONSTANT (DAYS-1) FOR BURNABLE ISOTOPES')
      CALL REAG2 ('  ALAM',ALAM(1), NCON)
      WRITE (NOUT,35)
35    FORMAT (/' BURNABLE ISOTOPE TYPE (0/1/2=NONE/FERTILE/FISSILE)')
      CALL REAI2 ('   NBR', NBR(1), NCON)
40    WRITE (NOUT, 45)
45    FORMAT (/' NUMBER OF DECAY SOURCES FOR EACH BURNABLE ISOTOPE')
      CALL REAI2 ('   NID', NID(1), NCON)
      K1=0
      DO 50 I=1,NCON
50    K1 = K1 + NID(I)
      IF (K1 .LE. 0) GO TO 70
      WRITE (NOUT, 55)
55    FORMAT (/' BURNABLE ISOTOPE NUMBER FOR EACH DECAY SOURCE')
      CALL REAI2 ('   IND', IND(1), K1)
      WRITE (NOUT, 60)
60    FORMAT (/' FRACTIONAL YIELD FOR EACH DECAY SOURCE')
      CALL REAG2 ('   YDN', YDN(1), K1)
70    WRITE (NOUT, 75)
75    FORMAT (/' NUMBER OF N,2N SOURCES FOR EACH BURNABLE ISOTOPE')
      CALL REAI2 ('   NIN', NIN(1), NCON)
      K2=0
      DO 80 I=1,NCON
80    K2 = K2 + NIN(I)
      IF (K2 .LE. 0) GO TO 100
```

```
      WRITE (NOUT, 85)
85    FORMAT (/' BURNABLE ISOTOPE NUMBER FOR EACH N,2N SOURCE')
      CALL REAI2 ('   INN', INN(1), K2)
      WRITE (NOUT, 90)
90    FORMAT (/' FRACTIONAL YIELD FOR EACH N,2N SOURCE')
      CALL REAG2 ('   YNN', YNN(1), K2)
100   WRITE (NOUT, 105)
105   FORMAT (/' NUMBER OF CAPTURE SOURCES FOR EACH BURNABLE ISOTOPE')
      CALL REAI2 ('   NIC', NIC(1), NCON)
      K3=0
      DO 110 I=1,NCON
110   K3 = K3 + NIC(I)
      IF (K3 .LE. 0) GO TO 130
      WRITE (NOUT, 115)
115   FORMAT (/' BURNABLE ISOTOPE NUMBER FOR EACH CAPTURE SOURCE')
      CALL REAI2 ('   INC', INC(1), K3)
      WRITE (NOUT, 120)
120   FORMAT (/' FRACTIONAL YIELD FOR EACH CAPTURE SOURCE')
      CALL REAG2 ('   YCN', YCN(1), K3)
130   WRITE (NOUT, 135)
135   FORMAT (/' NUMBER OF FISSION SOURCES FOR EACH BURNABLE ISOTOPE')
      CALL REAI2 ('   NIF', NIF(1), NCON)
      K4=0
      DO 140 I=1,NCON
140   K4 = K4 + NIF(I)
      IF (K4 .LE. 0) GO TO 160
      WRITE (NOUT, 145)
145   FORMAT (/' BURNABLE ISOTOPE NUMBER FOR EACH FISSION SOURCE')
      CALL REAI2 ('   INF', INF(1), K4)
      WRITE (NOUT, 150)
150   FORMAT (/' FRACTIONAL YIELD FOR EACH FISSION SOURCE')
      CALL REAG2 ('   YFN', YFN(1), K4)
160   IF (K1 .LE. 0) GO TO 190
      WRITE (NOUT, 165)
165   FORMAT (//30X, 'DECAY SOURCES'//)
      WRITE (NOUT, 166)
166   FORMAT (12X, 'TO', 17X, 'FROM', 15X, 'YIELD'/
     2        8X, '(#    NAME)', 8X, '(#    NAME)', 10X, '(FRACTION)'/)
      I = 0
      DO 180 II=1,NCON
      IF (NID(II) .LE. 0) GO TO 180
      DO 175 III=1,NID(II)
      I = I + 1
      LL = IND(I)
      LL = MATN(LL)
      LLL = MATN(II)
      WRITE (NOUT,170) I, II, (HOLN(LLL,L),L=1,2), IND(I),
     1                        (HOLN(LL,L), L=1,2), YDN(I)
170   FORMAT (I4, 2X, I4, 4X, 2A4, 3X, I4, 4X, 2A4, 4X, E12.3)
175   CONTINUE
180   CONTINUE
190   IF (K2 .LE. 0) GO TO 220
      WRITE (NOUT, 195)
195   FORMAT (//30X, 'N,2N SOURCES'//)
      WRITE (NOUT, 166)
      I = 0
      DO 210 II=1,NCON
      IF (NIN(II) .LE. 0) GO TO 210
      DO 205 III=1,NIN(II)
      I = I + 1
      LL = INN(I)
      LL = MATN(LL)
      LLL = MATN(II)
      WRITE (NOUT,170) I, II, (HOLN(LLL,L),L=1,2), INN(I),
     1                        (HOLN(LL,L), L=1,2), YNN(I)
205   CONTINUE
210   CONTINUE
220   IF (K3 .LE. 0) GO TO 250
      WRITE (NOUT, 225)
225   FORMAT (//30X, 'CAPTURE SOURCES'//)
      WRITE (NOUT, 166)
      I = 0
      DO 240 II=1,NCON
      IF (NIC(II) .LE. 0) GO TO 240
      DO 235 III=1,NIC(II)
      I = I + 1
```

```
      LL = INC(I)
      LL = MATN(LL)
      LLL = MATN(II)
      WRITE (NOUT,170) I, II, (HOLN(LLL,L),L=1,2), INC(I),
     1                         (HOLN(LL,L), L=1,2), YCN(I)
235   CONTINUE
240   CONTINUE
250   IF (K4 .LE. 0) GO TO 280
      WRITE (NOUT, 255)
255   FORMAT (//30X, 'FISSION SOURCES'//)
      WRITE (NOUT, 166)
      I = 0
      DO 270 II=1,NCON
      IF (NIF(II) .LE. 0) GO TO 270
      DO 265 III=1,NIF(II)
      I = I + 1
      LL = INF(I)
      LL = MATN(LL)
      LLL = MATN(II)
      WRITE (NOUT,170) I, II, (HOLN(LLL,L),L=1,2), INF(I),
     1                         (HOLN(LL,L), L=1,2), YFN(I)
265   CONTINUE
270   CONTINUE

280   DO 290  N=1, NCON
290   ALAM(N) = ALAM(N)/(3600.*24.)
      GO TO 10
300   CONTINUE
      RETURN
      END

      SUBROUTINE MARCH (PHIB,MATN,FXS,AXS,TXS,VOL,MASS,MASSP,ALAM,
     1                  NID, NIN, NIC, NIF, IND, INN, INC, INF,
     1                  YDN, YNN, YCN, YFN, IO, I1, I2, M2)
      DIMENSION  PHIB(1), MATN(1), FXS(ML,1), AXS(ML,1), VOL(1),
     1           MASS(ML,1), MASSP(ML,1), ALAM(1),
     2           IO(1), I1(1), I2(1), M2(1), TXS(ML,1),
     3           YDN(1), YNN(1), YCN(1), YFN(1),
     4           NID(1), NIN(1), NIC(1), NIF(1),
     5           IND(1), INN(1), INC(1), INF(1)
$INCLUDE:'1DB.INC'
C     THIS SUBROUTINE COMPUTES THE TIME DEPENDENT ISOTOPIC CONCENTRATION
      NSTEP = 10
      NITER = 3
      TEMP = DELT * 24. * 3600. / FLOAT(NSTEP)
      TEMP1 = .0
      DO 5 KZ = 1,IZM
      PHIB(KZ) = PHIB(KZ) * 10.**(-24)
      DO 5 KN = 1,NCON
      LN = MATN(KN)
5     TEMP1 = TEMP1 + FXS(KN,KZ)*PHIB(KZ)*MASS(LN,KZ)*VOL(KZ)
      DO 200 KT = 1,NSTEP
      TEMP3 = .0
      DO 20 KZ = 1,IZM
      DO 20 KN = 1,NCON
      LN = MATN(KN)
20    MASSP(LN,KZ) = MASS(LN,KZ)
      DO 100 KZ = 1,IZM
      DO 60 KKK = 1,NITER
      K1 = 0
      K2 = 0
      K3 = 0
      K4 = 0
      DO 60 KN = 1,NCON
      LN = MATN(KN)
      TEMP2=-(MASS(LN,KZ)+MASSP(LN,KZ))*(ALAM(KN)+AXS(KN,KZ)*PHIB(KZ))
      IF (NID(KN) .LE. 0) GO TO 30
      DO 25 KK=1,NID(KN)
      K1 = K1 + 1
      K = IND(K1)
      KL = MATN(K)
25    TEMP2 = TEMP2 + ALAM(K)*(MASS(KL,KZ) + MASSP(KL,KZ)) * YDN(K1)
30    IF (NIN(KN) .LE. 0) GO TO 40
      DO 35 KK=1,NIN(KN)
      K2 = K2 + 1
      K = INN(K2)
```

```
      KL = MATN(K)
35    TEMP2=TEMP2+TXS(K,KZ)*PHIB(KZ)*(MASS(KL,KZ)+MASSP(KL,KZ))*YNN(K2)
40    IF (NIC(KN) .LE. 0) GO TO 50
      DO 45 KK=1,NIC(KN)
      K3 = K3 + 1
      K = INC(K3)
      KL = MATN(K)
45    TEMP2 = TEMP2 + (AXS(K,KZ) - FXS(K,KZ))*PHIB(KZ) * YCN(K3) *
     1                (MASS(KL,KZ) + MASSP(KL,KZ))
50    IF (NIF(KN) .LE. 0) GO TO 60
      DO 55 KK=1,NIF(KN)
      K4 = K4 + 1
      K = INF(K4)
      KL = MATN(K)
55    TEMP2=TEMP2+YFN(K4)*FXS(K,KZ)*PHIB(KZ)*(MASS(KL,KZ)+MASSP(KL,KZ))
60    MASS(LN,KZ) = MASSP(LN,KZ) + .5*TEMP*TEMP2
      DO 100 KN = 1,NCON
      LN = MATN(KN)
100   TEMP3 = TEMP3 + FXS(KN,KZ)*PHIB(KZ)*MASS(LN,KZ)*VOL(KZ)
      IF (TEMP3)  200,200,110
110   DO 120 KZ = 1,IZM
120   PHIB(KZ) = PHIB(KZ) * TEMP1/TEMP3
200   CONTINUE
      DO 500 KZ = 1,IZM
500   PHIB(KZ) = PHIB(KZ)*10.**(24)
      DO 540  KZ=1,IZM
      DO 540  M=1,M01
      IF (I0(M) - M2(KZ))  540,520,540
520   DO 530  KN=1,NCON
      LN = MATN(KN)
      IF (LN - I1(M))  530,525,530
525   I2(M) = MASS(LN,KZ)
530   CONTINUE
540   CONTINUE
      RETURN
      END
```

This page intentionally left blank

DISTRIBUTION

Number of copies

ONSITE

2	U.S. Department of Energy-Richland Operations Office	
	O. A. Farabee	A6-55
	DOE-RL Public Reading Room	A1-65
35	Westinghouse Hanford Company	
	C. L. Bennett	N2-32
	R. A. Bennett	N2-32
	L. R. Campbell	N2-32
	L. L. Carter	H0-35
	J. W. Daughtry	H0-35
	K. D. Dobbin	N2-32
	S. R. Gedeon	H0-35
	R. A. Harris	H0-35
	S. F. Kessler	N2-32
	B. J. Knutson	L0-14
	J. S. Lan	H0-40
	F. M. Mann	H0-36
	J. V. Nelson	N2-32
	R. P. Omberg	H0-40
	J. A. Rawlins	H0-36
	R. F. Richard	N2-32
	F. A. Schmittroth	H0-36
	R. A. Schwarz	H0-35
	R. L. Simons	H0-35
	E. L. Tang	N2-32
	H. Toffer	H0-38
	J. C. Vankeuren	N1-19
	A. E. Waltar	H0-32
	R. H. Webb	L5-02
	D. W. Wootan	H0-35
	Central Files	L8-04
	Documentation and Record Services (4)	H4-17
	Documentation Processing and Distribution (2)	L8-15
	Information Release Administration (3)	A2-24
10	Pacific Northwest Laboratory	
	L. R. Dodd	K6-48
	S. W. Heaberlin	K6-42
	W. W. Little (2)	K6-42
	G. H. Meriwether	K6-42
	Technical Files (5)	K1-11

This page intentionally left blank.

END

DATE FILMED

01/23/92