A Multicomponent Time-Dependent Concentration Diffusion with Radioactive Decay Program
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

The views expressed in this report are not necessarily those of the US Nuclear Regulatory Commission.
DASH:
A Multicomponent Time-Dependent Concentration Diffusion with Radioactive Decay Program

C. E. Apperson, Jr.*
C. E. Lee**
L. M. Carruthers

*Department of Nuclear Engineering, University of Virginia, Charlottesville, VA 22903.
**Department of Nuclear Engineering, Texas A&M University, College Station, TX 77843.

Manuscript submitted: April 1979
Date published: April 1979

Prepared for
Division of Reactor Safety Research
US Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE Program R401
NRC FIN No. A 7014-9
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>I.</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II.</td>
<td>THEORY</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>A. Difference Equation Derivation</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>B. Analytic Operator Solution</td>
<td>8</td>
</tr>
<tr>
<td>III.</td>
<td>VALIDATION AND ACCURACY EVALUATION</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>A. One Material, One Specie Test Problems</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. Slab Problem</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>2. Cylinder Problems</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>a. Solid Cylinder</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>b. Hollow Cylinder</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3. Spherical Problems</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>a. Solid Sphere</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>b. Hollow Sphere</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>B. Two Material, Two Specie Test Problems</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>1. Critical Slab</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>2. Critical Cylinder</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>3. Critical Sphere</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>C. Inherent Differencing Error</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>D. Numerical Errors Associated with Matrix Inversion and Matrix Operator Solution</td>
<td>23</td>
</tr>
<tr>
<td>IV.</td>
<td>HOLDUP OF $^{90}$Sr BY GRAPHITE</td>
<td>23</td>
</tr>
<tr>
<td>V.</td>
<td>PROGRAM STRUCTURE</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>A. Role and Function of Subroutines</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>1. Primary Routines</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>a. INPA</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>b. INPB</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>c. GEOM</td>
<td>28</td>
</tr>
</tbody>
</table>
CONTENTS (cont)

d. TEMADJ .......................................................... 28
e. INPLT .......................................................... 28
f. DIJADJ .......................................................... 28
g. BCONL .......................................................... 28
h. MAKLAM .......................................................... 29
i. BIGEL .......................................................... 29
j. MAKEB .......................................................... 29
k. BCONR .......................................................... 29
l. SOLVER .......................................................... 29
m. MAKVOL .......................................................... 29
n. FSOLVE .......................................................... 29
o. CONCPLT .......................................................... 29

2. Secondary Routines .............................................. 29
3. Graphics .......................................................... 30
B. Program Flow ................................................... 31
C. DASH Input Instructions ........................................ 31
D. Machine Requirements .......................................... 32

VI. DASH TEST PROBLEM ........................................... 32

APPENDIX A. MATRIX OPERATOR EVALUATION .................. 38

APPENDIX B. DASH CODE LISTING ................................ 41

APPENDIX C. DASH TEST PROBLEM (with output) ............... 41

REFERENCES ........................................................ 42

TABLES

I. Geometric Variables ........................................... 3
II. Data for Validation Tests ..................................... 10

vi
TABLES (cont)

III. Two-Group Validation Test Data ................................................................. 16
IV. Spatial Differencing Error ............................................................................ 22
V. Data for Mass-90 Decay Chain .................................................................... 24
VI. Diffusion Coefficient Parameters ................................................................ 25
VII. Comparison of $^{90}$Sr Concentrations at One Year ........................................ 25
VIII. $^{90}$Sr Concentration in Fuel Matrix with Increasing Source ......................... 27
IX. DASH Input Instructions .............................................................................. 33
X. Special Read Format Options ...................................................................... 36
XI. Sample Problem Data .................................................................................. 37

FIGURES

Fig. 1. Discrete mesh function representation. ......................................................... 3
Fig. 2. Slab validation problem results. .................................................................. 11
Fig. 3. Solid cylinder validation problem results. ..................................................... 13
Fig. 4. Hollow cylinder validation problem results. ............................................... 13
Fig. 5. Solid sphere validation problem results. ...................................................... 14
Fig. 6. Hollow sphere validation problem results. ............................................... 15
Fig. 7. Critical slab analytic results. ..................................................................... 17
Fig. 8. Critical slab DASH results. ...................................................................... 17
Fig. 9. Slab flux ratio comparison. ...................................................................... 17
Fig. 10. Critical cylinder analytic results. .............................................................. 18
Fig. 11. Critical cylinder DASH results. ............................................................... 18
Fig. 12. Cylinder flux ratio comparison. ............................................................... 19
Fig. 13. Critical sphere analytic results. ............................................................... 20
Fig. 14. Critical sphere DASH results. ................................................................. 20
Fig. 15. Spherical flux ratio comparison. .............................................................. 20
Fig. 16. Relative inherent differencing error. ....................................................... 22
Fig. 17. Fuel-graphite-helium calculational model and beginning-of-life and six-year temperature profiles. .......................................................... 24
Fig. 18. $^{90}$Sr concentration profiles. ................................................................. 26

vii
Fig. 19. $^{90}\text{Y}$ concentration profiles. 27
Fig. 20. DASH flow diagram. 31
Fig. 21. Sample problem results for Diffusant A. 37
Fig. 22. Sample problem results for Diffusant B. 37
ABSTRACT

The multicomponent time-dependent diffusion with radioactive decay problem which arises in the study of high-temperature gas-cooled reactors fission product migration is solved in one-dimensional geometries. The spatial multicomponent diffusion operator is numerically represented by a conservative finite difference approximation. An analytic time-dependent solution is achieved using a matrix operator method. Comparisons of the analytic-numerical solution method with a variety of analytic solutions give excellent agreement. This solution technique has been incorporated into an algorithm for use in a computer code, DASH. The holdup of $^{90}$Sr by graphite is calculated.

I. INTRODUCTION

Multicomponent time-dependent concentration diffusion and radioactive decay of isotopic species$^1$ is an important aspect of fission product migration and release from fuel particles and fuel elements in High-Temperature Gas-Cooled Reactors (HTGRs). Analysis techniques for solving these types of problems are well known,$^2,3$ but are subject to time-step limitations to guarantee numerical accuracy and stability. These limitations are related to the magnitudes of the diffusion coefficients, decay constants, and spatial size of the system under consideration.

A one-dimensional analytic-numerical solution of this diffusion problem has been investigated. The diffusion operator is numerically approximated by a spatial finite-difference representation. The resulting time-dependent problem
is solved analytically using a matrix operator method. Comparisons to a number of known one-dimensional analytic solutions have been made. These comparison problems include the one specie and two species, two material slab, cylinder, and sphere.

In all instances considered, the agreement with analytic solutions is excellent, limited only by the accuracy limitations of the finite difference representation. The time-step limitation associated with other numerical solution methods has been eliminated.

This analytic-numerical technique has been utilized as the solution routine in a computer code, DASH, for solving the general problem of concentration diffusion with radioactive decay.

II. THEORY

The differential equation governing time-dependent multicomponent diffusion with radioactive decay is given by

$$\frac{d\bar{\mathbf{C}}}{dt} = \nabla \cdot D \nabla \bar{\mathbf{C}} - \lambda \bar{\mathbf{C}} + \bar{\mathbf{S}},$$

where $D$ is an $n \times n$ square positive definite diffusion matrix (cm$^2$/s), $\bar{\mathbf{C}}$ is an $n$-component column vector representing isotopic concentrations (atoms/cm$^3$), $\lambda$ is the decay matrix including branching rations (1/s), and $\bar{\mathbf{S}}$ is an $n$-component column source vector (atoms/cm$^3$s). Equation (1) is solved in one-dimensional geometries (slab, cylinder, or sphere) subject to the initial condition $\bar{\mathbf{C}}(r, t) = \bar{\mathbf{C}}(r, 0)$ and either homogeneous Newman ($D \nabla \bar{\mathbf{C}} = 0$) or inhomogeneous or homogeneous Dirichlet ($\bar{\mathbf{C}} = \bar{\sigma}$ or $\bar{\mathbf{C}} = 0$) boundary conditions.

A. Difference Equation Derivation

A finite-difference representation for the spatial diffusion operator is obtained by integrating Eq. (1) over a subvolume of a discrete mesh. Gauss' theorem, when applied to the integrated result, yields

$$V_k \frac{d}{dt} \bar{\mathbf{C}}_k = -A_k + 1/2 \bar{J}_k + 1/2 + A_k - 1/2 \bar{J}_k - 1/2 - \lambda_k V_k \bar{\mathbf{C}}_k + V_k \bar{\mathbf{S}}_k,$$  (2)

In Eq. (2), $\bar{\mathbf{C}}_k$ is the concentration vector averaged over the $k^{th}$ cell, $V_k$ is the volume of the $k^{th}$ cell (diagonal matrix), and $A_k + 1/2$ and $\bar{J}_k + 1/2$ are the area elements and current at the boundary between cell $k$ and
and $k \pm 1$. The cell-centered source vector averaged over the $k^{th}$ computational cell is denoted by $\vec{S}_k$. The decay matrix in cell $k$, $\lambda_k$, is cell dependent only if neutron processes are included in addition to $\beta$ decay.

The currents at the mesh boundaries, $\vec{J}_{k \pm 1/2}$, are evaluated in terms of the concentration vector

$$\vec{J}_{k \pm 1/2} = -D \nabla \vec{C}_{k \pm 1/2}. \quad (3)$$

The mesh spacing ($\Delta r_k$), area elements ($A_{k \pm 1/2}$), and volume elements ($V_k$) for cell $k$ as a function of geometry are given in Table I. The notation used throughout this discussion is illustrated in Fig. 1.

In order to develop difference equations that will be amenable to concentration-dependent diffusion coefficients and concentration discontinuities, the representation

---

**TABLE I**

**GEOMETRIC VARIABLES**

<table>
<thead>
<tr>
<th>Geometry</th>
<th>$\Delta r_k$</th>
<th>$A_{k \pm 1/2}$</th>
<th>$V_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slab</td>
<td>$r_k + \frac{1}{2} - r_k - \frac{1}{2}$</td>
<td>1</td>
<td>$\Delta r_k$</td>
</tr>
<tr>
<td>Cylinder</td>
<td>$r_k + \frac{1}{2} - r_k - \frac{1}{2}$</td>
<td>$2\pi r_k + \frac{1}{2}$</td>
<td>$\pi (r_k^2 + \frac{1}{2} - r_k^2 - \frac{1}{2})$</td>
</tr>
<tr>
<td>Sphere</td>
<td>$r_k + \frac{1}{2} - r_k - \frac{1}{2}$</td>
<td>$4\pi r_k^2 + \frac{1}{2}$</td>
<td>$\frac{4\pi}{3} (r_k^3 + \frac{1}{2} - r_k^3 - \frac{1}{2})$</td>
</tr>
</tbody>
</table>

---

Fig. 1. Discrete mesh function representation.
\[ \int_{r_k}^{r_{k+\frac{1}{2}}} \vec{J} \, dr = \int_{r_k}^{r_{k+1}} \ vec{J} \, dr = \int_{r_k}^{r_{k+\frac{1}{2}}} \vec{J} \, dr \]

is used or substituting \( \mathbf{D} \mathbf{v} \mathbf{C} \) for \( \mathbf{J} \),

\[ \int_{C_k}^{C_{k+\frac{1}{2}}} \mathbf{D} \, dC = \int_{C_k}^{C_{k+1}} \mathbf{D} \, dC \]

For the case of continuity of concentration and concentration independent diffusion coefficients \( \mathbf{D} \neq \mathbf{D}(\bar{C}) \), Eq. (5) yields

\[ \dot{C}_{k+\frac{1}{2}} = (\mathbf{D}_k + \mathbf{D}_{k+1})^{-1} (\mathbf{D}_k \dot{C}_k + \mathbf{D}_{k+1} \dot{C}_{k+1}) \]

where \( \mathbf{D}_k \) is the diffusion coefficient matrix in cell \( k \). These assumptions are valid for the problem being studied.

Using Eq. (6) the current at the boundary \( k + \frac{1}{2} \) may be evaluated as

\[ \dot{J}_{k+\frac{1}{2}} = 4 \mathbf{D}_k \left( \dot{C}_{k+\frac{1}{2}} - \dot{C}_k \right) / (\Delta r_k + \Delta r_{k+1}) \]

\[ = 4 \mathbf{D}_k (\mathbf{D}_k + \mathbf{D}_{k+1})^{-1} \mathbf{D}_{k+1} \left( \dot{C}_{k+1} - \dot{C}_k \right) / (\Delta r_k + \Delta r_{k+1}) \]  

Similarly, making the same argument for cells \( (k - 1, k) \),

\[ \dot{J}_{k-\frac{1}{2}} = 4 \mathbf{D}_k-1 \left( \mathbf{D}_k + \mathbf{D}_{k-1} \right)^{-1} \mathbf{D}_k \left( \dot{C}_k - \dot{C}_{k-1} \right) / (\Delta r_{k-1} + \Delta r_k) \]

Note that \( (\mathbf{D}_k + \mathbf{D}_{k+1})^{-1} \) represents a matrix inverse of a positive definite diffusion coefficient matrix. Substituting Eq. (7) and (8) into Eq. (2) results in
\[
\frac{d\hat{C}_k}{dt} = \bar{A}_k \hat{C}_k + 1 + \bar{B}_k \hat{C}_{k-1} + V_k \hat{S}_k,
\]

(9)

where the coefficient matrices are given by

\[
\bar{A}_k = 4 A_k + 1/2 D_k (D_k + D_k + 1)^{-1} D_k + \frac{1}{2}(\Delta r_k + \Delta r_k + 1),
\]

\[
\bar{B}_k = 4 A_k - 1/2 D_k - 1 (D_{k-1} + D_k)^{-1} D_k (\Delta r_{k-1} + \Delta r_k),
\]

and

\[
\bar{k}_k = - A_k - B_k \lambda_k V_k.
\]

(10)

Since \(\bar{A}_k = \bar{B}_k + 1\), a reciprocity relationship exists.\(^7\)

The spatial boundary conditions treated are reflection (\(J = 0\)), homogeneous Newman, and concentration specification, homogeneous and inhomogeneous Dirichlet.

For reflection at the left-hand side of the cell \(k = 1\), \(\bar{B}_k\) is set to zero for \(k = 1\). For reflection at the right-hand side of cell \(k = K\), \(\bar{A}_k\) is set to zero for \(k = K\) in Eq. (10). This procedure eliminates reference to either \(\hat{C}_0\) or \(\hat{C}_{K+1}\), and corresponds to a zero current boundary condition.

When the concentration is specified on the left-hand side of a slab or on the interior surface of a hollow cylinder or sphere, \(k\) is equal to \(k_0 - \frac{1}{2}\). A \(k_0\) value of 1 corresponds to the first calculational cell in a slab but it corresponds to the central cell in a hollow cylinder or hollow sphere. In a hollow cylinder or sphere the first calculational cell is \(k_0 = 2\).

The left-hand current for both cases is given by

\[
J_{k_0 - \frac{1}{2}} = 2 D_{k_0} \left( \hat{C}_{k_0} - \hat{C}_{k_0 - \frac{1}{2}} \right) / \Delta r_{k_0},
\]

(11)

where the concentration vector \(\hat{C}_{k_0 - \frac{1}{2}}\) is specified. The right-hand current is given by Eq. (7) with \(k = k_0\). From these results a modified set of coefficients for Eq. (9) can be evaluated

\[
\bar{A}_{k_0} = 4 A_{k_0} + 1/2 D_{k_0} (D_{k_0} + D_{k_0} + 1)^{-1} D_{k_0} + \frac{1}{2}(\Delta r_{k_0} + \Delta r_{k_0} + 1),
\]

(12)
\[ B_{k_0}^- = 2 A_{k_0} - 1/2 \frac{D_{k_0}}{\Delta r_{k_0}}, \]  
\[ K_{k_0}^- = -\bar{A}_{k_0} - \bar{B}_{k_0} - \lambda_{k_0} V_{k_0}. \]  

It should be noted that \( \bar{A}_{k} \) coefficients in Eqs. (10) and (12) are identical.

To account for the concentration diffusion of the material inside boundary 1, the source vector is modified.

\[ V_{k_0} \dot{\xi}_{k_0} = V_{k_0} \dot{\xi}_{k_0} + \bar{B}_{k_0} \xi_{k_0} - 1/2. \]  

Similarly, for concentration specified at the outside boundary of cell \( k = K \), a modified set of coefficients for Eq. (9) must also be developed. In this case the \( \bar{B}_k \) coefficients are identical between Eq. (10) and Eq. (14).

\[ \bar{A}_K^- = 2 A_K + 1/2 \frac{D_K}{\Delta r_K}, \]  
\[ \bar{B}_K = 4 A_K - 1/2 D_K - 1 (D_{K-1} + D_K)^{-1} D_K / (\Delta r_{K-1} + \Delta r_K), \]  
\[ \bar{K}_K^- = -\bar{A}_K^- - \bar{B}_K - \lambda_K V_K. \]  

In like manner, also the source vector must be modified to account for the concentration diffusion of the material specified on the outside boundary.

\[ V_K \dot{\xi}_K = V_K \dot{\xi}_K + \bar{A}_K^- \xi_K + 1/2. \]  

The equations represented by Eqs. (9-10) and (12-15) may be written in supermatrix, supervector form as

\[ \nu \frac{d}{dt} \dot{\mathbf{C}} = A \dot{\mathbf{C}} + \nu \dot{\mathbf{S}}, \]  

where
Equations (12) and (14) are included in $\tilde{S}$ for the first and/or last elements for the case of concentration specification at a boundary. The $V_k$ in Eq. (18) are diagonal matrices of the form

$$V_k = I \, V,$$

(20)
where $V$ is the scalar volume of the $k^{th}$ cell and $I$ is the $n$-dimensional identity matrix, where $n$ is the number of nuclides in the radioactive decay chain. Each of the elements in $A$ is an $n$-by-$n$ matrix. The elements $\bar{A}_k$, $\bar{B}_k$, and $\bar{R}_k$ are defined by Eq.s (10), (12), and (14).

B. Analytic Operator Solution

Although Eq. (16) could be solved by a standard implicit time-differencing technique, such techniques are limited in time-step size by spectral considerations. Instead, an operator method is used.

By defining

$$\bar{X} = V \tilde{X}, \quad \tilde{g} = VS, \quad \text{and} \quad B = AV^{-1}$$

and assuming $A$ is constant over the interval $(0, t)$, Eq. (16) takes the form

$$\frac{d\bar{X}}{dt} = B \bar{X} + \tilde{g},$$

which has the solution

$$\bar{X}(t) = e^{Bt} \bar{X}(0) + tD(Bt) \tilde{g},$$

where

$$D(Bt) = (Bt)^{-1} (e^{Bt} - I),$$

and $X(0)$ is the vector of initial concentrations. Substituting Eq. (21) into Eq. (23), the solution to Eq. (16) is given by

$$\tilde{C}(t) = V^{-1} e^{Bt} \tilde{C}(0) + V^{-1} tD(Bt) VS,$$

where $B = AV^{-1}$ (1/s) and $V$ is a diagonal cell volume matrix. The details for evaluating the matrix operators $e^{Bt}$ and $D(Bt)$ for arbitrary $t$ are given in App. A.
III. VALIDATION AND ACCURACY EVALUATION

Although no experimental validation of DASH has been conducted, a substantial number of comparisons have been made to published analytic solutions. No attempt has been made to make all the possible comparisons, but a sufficient number of problems have been compared to establish confidence in the DASH methodology. For the problems considered, the observed errors are of the magnitude one would expect from a spatial finite-differencing technique. Some of these comparisons are discussed in detail. The test problems which are discussed were chosen because they point up unique features of the code.

A. One Material, One Specie Test Problems

The simplest problem type to utilize the full capabilities of the DASH code is the one material, one specie problem with concentration diffusion and radioactive decay. The one-dimensional geometry in the code permits the evaluation of problems involving an infinite slab, an infinite solid or hollow cylinder, and a solid or hollow sphere. The analytic solutions for comparison are taken from Crank \(^8\) and Carslaw and Jaeger. \(^3\) These published results are for concentration diffusion without radioactive decay or can be modified to fit this type of problem. A transformation developed by Danckwerts \(^9\) can be used to extend these results for time-dependent concentration diffusion to also handle radioactive decay. Danckwerts' transformation states that:

\[
 C = \lambda \int_0^t \dot{C} e^{-\lambda t} \, dt + C e^{\lambda t}, \tag{26}
\]

where \(\lambda\) is the radioactive decay constant (1/s), \(\dot{C}\) is the diffusion solution without radioactive decay (atoms/cm\(^3\)), \(t\) is the evaluation time (s), and \(C\) is the solution with both diffusion and decay (atoms/cm\(^3\)). This transformation is valid for an initial concentration of zero, and boundary conditions of either surface-saturation or surface-resistance.

1. Slab Problem

The analytic solution for time-dependent concentration diffusion in a slab \((0 \leq x \leq L)\) with a uniform initial distribution and different saturated surface concentrations is \(^8\)
\[ C^* = C_1 + (C_2 - C_1) \frac{x}{\ell} + \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{C_2 \cos (n\pi) - C_1}{n} \sin \frac{n\pi x}{\ell} \exp \left( -\frac{Dn^2\pi^2 t}{\ell^2} \right) \]

\[ + \frac{4C_0}{\pi} \sum_{m=0}^{\infty} \frac{1}{2m + 1} \sin \left( \frac{(2m + 1) \pi x}{\ell} \right) \exp \left( -D(2m + 1)^2\pi^2 t \right) \]  

(27)

where \( C_0 \) is the initial uniform concentration, \( C_1 \) is the surface concentration at \( x = 0 \), \( C_2 \) is the surface concentration at \( x = \ell \), \( D \) is the diffusion coefficient, and \( t \) is the evaluation time.

A simple one material, one species infinite slab problem has been defined which can be solved both by Eqs. (26) and (27) and by DASH. The data for this problem is tabulated in Table II. The test problem was solved analytically at 27 space points at 5 different times. The DASH solution was for the same 5 times using 25 mesh cells. The maximum error observed occurred during the first time step, 0.1 days, at the center of the slab and had a magnitude of 0.28%. The magnitude of the error is defined to be the absolute value of the difference in the analytic and DASH results divided by the analytic result. The results are compared in Fig. 2. The figure resolution is such that the analytic and analytic-numerical, DASH, results fall on top of each other.

2. Cylinder Problems

a. Solid Cylinder. The time-dependent concentration diffusion problem for an infinite solid cylinder \((0 \leq r \leq a)\) with a uniform initial distribution and a constant concentration at the outer radius is given analytically by 8

<table>
<thead>
<tr>
<th>TABLE II</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA FOR VALIDATION TESTS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GEOMETRY</th>
<th>DIFFUSION COEFFICIENT ((\text{cm}^2 \text{s}^{-1}))</th>
<th>DECAY CONSTANT ((\text{s}^{-1}))</th>
<th>UNIFORM INITIAL CONCENTRATION ((\text{atoms/cm}^3))</th>
<th>BOUNDARY CONDITIONS Left ((\text{atoms/cm}^3))</th>
<th>BOUNDARY CONDITIONS Right ((\text{atoms/cm}^3))</th>
<th>DIMENSIONS ((\text{cm}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slab</td>
<td>7.234 \times 10^{-6}</td>
<td>8.0225 \times 10^{-7}</td>
<td>0.0</td>
<td>1.0 \times 10^{10}</td>
<td>1.0 \times 10^{10}</td>
<td>1 cm thick</td>
</tr>
<tr>
<td>Solid Cylinder</td>
<td>7.234 \times 10^{-6}</td>
<td>8.0225 \times 10^{-7}</td>
<td>0.0</td>
<td>1.0 \times 10^{10}</td>
<td>1.0 \times 10^{10}</td>
<td>1 cm radius</td>
</tr>
<tr>
<td>Hollow Cylinder</td>
<td>7.234 \times 10^{-6}</td>
<td>8.0225 \times 10^{-7}</td>
<td>0.0</td>
<td>1.0 \times 10^{10}</td>
<td>1.0 \times 10^{10}</td>
<td>0.5 cm I.D., 2.0 cm o.d.</td>
</tr>
<tr>
<td>Solid Sphere</td>
<td>7.234 \times 10^{-6}</td>
<td>8.0225 \times 10^{-7}</td>
<td>0.0</td>
<td>1.0 \times 10^{10}</td>
<td>1.0 \times 10^{10}</td>
<td>1 cm radius</td>
</tr>
<tr>
<td>Hollow Sphere</td>
<td>7.234 \times 10^{-6}</td>
<td>8.0225 \times 10^{-7}</td>
<td>0.0</td>
<td>1.0 \times 10^{10}</td>
<td>1.0 \times 10^{10}</td>
<td>0.5 cm I.D., 2.0 cm o.d.</td>
</tr>
</tbody>
</table>
Fig. 2. Slab validation problem results.

\[ C = C_1 + (C_0 - C_1) \frac{2}{a} \sum_{n=1}^{\infty} \frac{\exp(-D\alpha_n^2 t) J_0(r\alpha_n)}{\alpha_n J_1(a \alpha_n)} , \]

(28)

where \( C_0 \) is the initial uniform concentration, \( C_1 \) is the boundary concentration, \( D \) is the diffusion coefficient, and \( t \) is the evaluation time. The \( \alpha_n \)'s are roots of the Bessel function of the first kind of order zero,

\[ J_0(a \alpha_n) = 0, \]

(29)

where \( a \) is the cylinder radius. The problem defined in Table II for a solid cylinder can be solved both by Eqs. (26) and (28) and by DASH. The analytic solution was evaluated at 27 space points at 5 different times. The same 5 time points were used when the problem was solved using DASH with 25 mesh cells. The maximum observed error of 1.5%, the largest error for the one material, one specie problems studied, occurred at the center of the cylinder on the first time step, 0.1 days. The results, Fig. 3, from the two calculations again fall on top of each other due to the resolution limits of the graphic scales.
b. Hollow Cylinder. The analytic solution to the problem of flow through a cylinder wall \((a \leq r \leq b)\) is

\[
C = C_1 \frac{\ln b - \ln \frac{b}{a}}{\ln (b/a)} + C_2 \frac{\ln \frac{b}{a}}{\ln (b/a)} + \pi C_0 \sum_{n=1}^{\infty} \frac{J_0(\alpha_n) U_0(\alpha_n) \exp(-D\alpha_n^2t)}{J_0(\alpha_n) + J_0(\beta_n)}
\]

\[
+ \pi \sum_{n=1}^{\infty} \frac{(C_2 J_0(\alpha_n) - C_1 J_0(\beta_n)) J_0(\alpha_n) U_0(\alpha_n) \exp(-D\alpha_n^2t)}{J_0^2(\alpha_n) - J_0^2(\beta_n)},
\]

(30)

where \(C_0\) is the initial uniform concentration, \(C_1\) is the inner boundary concentration \((r = a)\), and \(C_2\) is the outer boundary concentration \((r = b)\). The function \(U_0\) is given by

\[
U_0(\alpha_n) = J_0(\alpha_n) Y_0(\beta_n) - J_0(\beta_n) Y_0(\alpha_n).
\]

(31)

The values of \(\alpha_n\) are the positive roots of

\[
U_0(\alpha_n) = 0,
\]

(32)

where \(a\) is the inner radius and \(b\) is the outer radius of the hollow cylinder. The hollow cylinder problem solved both by Eqs. (26) and (30) and by DASH is stated in Table II.

Analytic solutions were evaluated at 26 space points at 5 different times. This problem was solved with DASH at the same 5 time points using 24 mesh cells. The maximum error observed was 0.24% and it was encountered at the first time step, 0.1 day. The error occurred at a point located a third of the way between the cylinder walls when measuring from the inside boundary. The results are illustrated in Fig. 4. It should be noted that the scaling of the ordinate is not the same as in the previous figures.
3. Spherical Problems

a. Solid Sphere. The problem of diffusion in a sphere \(0 < r < a\) has an analytic solution given by\(^8\)

\[
C = C_1 + (C_1 - C_0) \frac{2a}{\pi r} \sum_{n=1}^{\infty} \frac{(-1)^n}{n} \sin \frac{n\pi r}{a} \exp \left( -\frac{Dn^2\pi^2 t}{a^2} \right),
\] (33)

where \(C_0\) is the initial uniform concentration, \(C_1\) is the boundary concentration \((r = a)\), and \(D\) is the diffusion coefficient.

Using the solid sphere data of Table II, this problem can be solved analytically by Eqs. (26) and (33) and numerically by DASH.

Analytic solutions were obtained at 27 space points for 5 time intervals. DASH solutions were calculated for the same 5 time intervals in 25 mesh cells. The maximum error for this set of problems was 0.93\% and it occurred at the first time step, 0.1 days. This error was observed at a point \(a/4\) from the sphere center. The analytic and DASH results are given in Fig. 5.
**b. Hollow Sphere.** The analytic solution for flow through a spherical wall (a ≤ r ≤ b) is

\[
C = \frac{aC_1}{r} + \frac{(bC_2 - aC_1)(r - a)}{r(b - a)} + \sum_{n=1}^{\infty} \frac{b(C_2 - C_0)\cos(n\pi) - a(C_1 - C_0)}{n} \sin \left( \frac{n\pi(r - a)}{b - a} \right) \exp \left( -\frac{Dn^2\pi^2t}{(b - a)^2} \right),
\]

(34)

where \(C_0\) is the initial uniform concentration, \(C_1\) is the boundary concentration at \(r = a\), \(C_2\) is the boundary concentration at \(r = b\), and \(D\) is the diffusion coefficient.

The hollow sphere problem can be solved both by Eqs. (26) and (34) and by DASH.

The analytic results were evaluated at the 26 space points at 5 different times. The DASH solutions were for the same 5 time steps using 24 mesh cells.
A maximum error of 0.38% was observed at the first time step, 0.1 days, at a point located 20% of the way between the shell boundaries when measured from the inner wall. The calculated results are illustrated in Fig. 6.

B. Two Material, Two Specie Test Problems

Steady-state solutions can be readily obtained for the two-group neutron diffusion problem in reflected critical masses. One popular technique for solving these problems analytically is the critical determinant method. Using this approach, the critical radius of an infinite slab, infinite cylinder, or sphere can be evaluated. With this information the steady-state fast and thermal flux shapes in the fissile and reflector material can be determined.

The problem of neutron diffusion is extremely similar to the problems of concentration diffusion being studied. Because of this, the DASH code can be used to solve the two-group neutron diffusion problem with only minor modifications to the existing input routines. This is not to say that DASH can be used as a neutron diffusion code. DASH is optimized to solve Eq. (1) and lacks certain desirable characteristics for a production code for neutron diffusion.

![Hollow Sphere with Diffusion and Decay](image)

**Fig. 6.** Hollow sphere validation problem results.
The two-group neutron diffusion problem when set up in DASH produces full diffusion and decay matrices. This in conjunction with the two specie two-material nature of the problem provides an extensive test of the DASH code's ability to evaluate a steady-state solution. The test is further complicated by the need to reproduce the thermal flux peak. It is necessary to analytically determine the material interface for DASH, since it has no routines for evaluating the critical radius.

The basic data used in this series of problems is given in Table III. The $k_\infty$ is 1.388 9 and the reflector is always 25 cm thick.

1. Critical Slab

For the data given in Table III the half-thickness of a critical slab is 7.988 cm. Using the previously discussed analytic approach, the fast and thermal fluxes were calculated at 25 equally spaced points in material 1 and at 75 equally spaced points in material 2. More points were placed in material 2 to allow the thermal flux peak to be properly described. Numerical results were obtained with DASH using 12 mesh cells in material 1 and 38 mesh cells in material 2. These results are illustrated in Figs. 7 and 8.

The maximum error in the fast flux was 0.33% and the maximum thermal flux error was 0.60%. Both of these errors occurred in material 2 just after the material interface.

A further measure of the accuracy of the DASH results when compared to the analytic results is the fast-to-thermal flux ratio, Fig. 9. The ratio of the fast to thermal flux is plotted for both calculations. The maximum error observed in this ratio is 0.92% and it occurred in the same region as the other errors for this problem.

<table>
<thead>
<tr>
<th>Table III</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWO GROUP VALIDATION TEST DATA</td>
</tr>
<tr>
<td>Group 1</td>
</tr>
<tr>
<td>Material 1</td>
</tr>
<tr>
<td>Diffusion Coefficient (cm)</td>
</tr>
<tr>
<td>Absorption Cross-Section (cm⁻¹)</td>
</tr>
<tr>
<td>Fission Cross-Section (cm⁻¹)</td>
</tr>
</tbody>
</table>
Fig. 7. Critical slab analytic results.

Fig. 8. Critical slab DASH results.

Fig. 9. Slab flux ratio comparison.
2. Critical Cylinder

The critical cylinder problem when solved using the data of Table III has a critical radius of 15.368 cm. An analytic evaluation of the fast and thermal flux was done at 25 space points in material 1 and at 75 space points in material 2. DASH results were obtained for 19 material-1 mesh cells and 31 material-2 mesh cells. These results are illustrated individually in Figs. 10 and 11. The maximum error in the fast flux occurred 40 cm from the cylinder centerline and had a magnitude of 0.58%. The maximum thermal flux error was 0.80% and occurred 15 cm from the centerline. As in the slab problem the fast-to-thermal flux ratios were also compared, Fig. 12. The largest error observed was 1.21%. This error occurred at a point essentially at the material interface.

Fig. 10. Critical cylinder analytic results.
Fig. 11. Critical cylinder DASH results.
3. Critical Sphere

A critical radius of 21.91 cm is obtained when the Table III data is used to solve a spherical critical determinant problem. The analytically determined fluxes, Fig. 13, were evaluated at 25 space points in material 1 and at 75 space points in material 2. The DASH results, Fig. 14, were calculated based on 25 material-1 mesh cells and 25 material-2 mesh cells.

The maximum error for both flux groups occurred at the material interface. The largest fast flux error was 0.74% and the largest thermal flux error was 1.25%. The flux ratio comparison, Fig. 15, has its greatest error in material 2 near the material interface. The magnitude of this error is 0.97%.

C. Inherent Differencing Error

The DASH solution is obtained through the application of both analytic and numerical solution techniques. The procedure employed uses a matrix operator method to evaluate the time-dependent solution after the spatial variable has been differenced. The inherent error in the spatial differencing can be determined by expressing the difference equation with a Taylor's series.
Fig. 13. Critical sphere analytic results.

Fig. 14. Critical sphere DASH results.

Fig. 15. Spherical flux ratio comparison.
From the Taylor's series representation, the inherent error can be represented by an even power series of $h$, the mesh spacing. When $h$ is small, the principal error contribution comes from the $h^2$ term. Under these conditions, it is acceptable to assume that the inherent error due to spatially differencing Eq. (1) is proportional to $h^2$.

$$
\varepsilon = kh^2 ,
$$

(35)

where

\[ \varepsilon = \text{inherent error} \]
\[ k = \text{proportionality constant} \]
\[ h = \text{mesh spacing}. \]

By substituting $L/n$ for the mesh spacing in Eq. (35), where $L$ is the thickness of the sample and $n$ is the number of cells in $L$, a more general expression can be obtained.

$$
\varepsilon = (kL^2) \frac{1}{n^2} .
$$

(36)

For a given geometry $kL^2$ is constant. The analytic-numerical DASH solution accuracy, therefore, should vary inversely with the square of the number of cells if the code is properly constructed.

As a test of this property, the slab problem of paragraph III, A, l was evaluated at five different mesh sizes. The results of this exercise are given in Table IV and Fig. 16. The maximum observed error over five time steps was used in this study. One can see from Table IV that $\varepsilon n^2$ is approximately constant.
### TABLE IV

SPATIAL DIFFERENCING ERROR

<table>
<thead>
<tr>
<th>n</th>
<th>$\epsilon$</th>
<th>$n^2$</th>
<th>$\epsilon n^2$</th>
<th>Normalized $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.061320</td>
<td>25</td>
<td>1.53</td>
<td>1.000</td>
</tr>
<tr>
<td>10</td>
<td>0.016860</td>
<td>100</td>
<td>1.69</td>
<td>0.275</td>
</tr>
<tr>
<td>15</td>
<td>0.007778</td>
<td>225</td>
<td>1.75</td>
<td>0.127</td>
</tr>
<tr>
<td>20</td>
<td>0.004370</td>
<td>400</td>
<td>1.75</td>
<td>0.071</td>
</tr>
<tr>
<td>25</td>
<td>0.002820</td>
<td>625</td>
<td>1.76</td>
<td>0.046</td>
</tr>
</tbody>
</table>

Fig. 16. Relative inherent differencing error.
D. Numerical Errors Associated with Matrix Inversion and Matrix Operator Solution

The basic equation to be solved [Eq. (16)] involves a supermatrix $A$ given by Eq. (19) whose coefficients $\bar{A}$ and $\bar{B}$ depend on the inverse of the diffusion matrix [see Eq. (10)]. This inverse will be difficult to perform in some numerical situations. For submatrices with no off-diagonal terms this is not a problem, however.

The full set of equations involving the supermatrix is solved by a matrix operator method which involves summing the terms in the matrix as a first step. This sum is used to decide how many times the matrix should be divided by two to reduce the terms of the matrix to manageable size. If the matrix has a few very large terms, this method may cause the part of the solution which results from this operation to disappear. One type of problem which has this difficulty is one in which the cells are of very uneven sizes. The individual terms have $\Delta r$ in the denominator and this causes the elements of the supermatrix to be large if the cell they refer to is small.

IV. HOLDUP OF $^{90}$Sr BY GRAPHITE

A parameter study of the release and diffusion-decay of isotopes of strontium in a simplified one-dimensional slab model of an HTGR core block has been carried out. A typical element of the core block and the coolant hole was modeled as shown in Fig. 17; the dimensions of each region were taken from Ref. 11.

A decay chain used for the test problem is

$^{90}$Sr $\rightarrow^{90}$Y $\rightarrow^{90}$Zr

with yields and decay constants shown in Table V. The boundary conditions used are reflection at $x = 0$, zero concentration at $x = 1.05$.

The approach is to use data from the work of Appel and Roos$^{11}$ and calculate the distribution of the isotopes of this decay chain in the fuel matrix and structural graphite. The source term for $^{90}$Sr is taken to be $7.3 \times 10^9$ atoms/(cm$^3$.s) as given in Ref. 11. The source terms for the other isotopes in the chain are taken in proportion to the yields of Table V.
The temperature changes from the beginning to the end of the calculation (six years duration) are shown in Fig. 17. Temperatures at intermediate times are calculated by linear interpolation.

Data are given in Ref. 12 for the diagonal terms of the 3x3 diffusion matrix for the three species making up this problem. For the Arrhenius representation,

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>YIELD %</th>
<th>DECAY CONSTANT/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$^{90}\text{Sr}$</td>
<td>5.77</td>
<td>$7.844 \times 10^{-10}$</td>
</tr>
<tr>
<td>$^{90}\text{Y}$</td>
<td>5.77</td>
<td>$2.994 \times 10^{-6}$</td>
</tr>
<tr>
<td>$^{90}\text{Zr}$</td>
<td>0.0</td>
<td>$1.0 \times 10^{-20}$</td>
</tr>
</tbody>
</table>

Fig. 17. Fuel-graphite-helium calculational model and beginning-of-life and six-year temperature profiles.
-log_{10} D = A + 1000 B/T, \hspace{1cm} (37)

the coefficients A and B are given in Table VI. The data were taken from Ref. 12.

Appel and Roos\cite{11} assume that the concentration of $^{90}\text{Sr}$ drops by a factor of 300 at the fuel-graphite interface corresponding to the distribution coefficient between the two substances. This is handled in DASH by putting a small region (10^{-5} \text{ cm thick}) at the boundary and adjusting the diffusion coefficient of the region introduced until the ratio of $^{90}\text{Sr}$ concentrations is 300. Except for this boundary region, the mesh spacing is taken as 0.05 \text{ cm throughout.}

To compare with the work of Appel and Roos, the concentrations of $^{90}\text{Sr}$ were calculated at the end of one year using the diffusion coefficient data from Ref. 11 [A = -2.477 and B = 13.1 in Eq. (37)] and the data of Table VI for comparison. The comparisons are shown in Table VII.

**TABLE VI**

DIFFUSION COEFFICIENT PARAMETERS

<table>
<thead>
<tr>
<th>SPECIE</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$^{90}\text{Sr}$</td>
<td>0.34</td>
<td>6.5</td>
</tr>
<tr>
<td>$^{90}\text{Y}$</td>
<td>0.74</td>
<td>14.2</td>
</tr>
<tr>
<td>$^{90}\text{Zr}$</td>
<td>1.19</td>
<td>22.8</td>
</tr>
</tbody>
</table>

**TABLE VII**

COMPARISON OF $^{90}\text{Sr}$ CONCENTRATIONS AT ONE YEAR

<table>
<thead>
<tr>
<th>POSITION (cm)</th>
<th>MATERIAL</th>
<th>Results from Ref. 11 (atoms/cm³)</th>
<th>DASH Results</th>
<th>Ref. 11 Coefficients</th>
<th>Ref. 12 Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>Fuel</td>
<td>$1.7 \times 10^{18}$</td>
<td></td>
<td>$2.27 \times 10^{17}$</td>
<td>$1.80 \times 10^{17}$</td>
</tr>
<tr>
<td>0.30</td>
<td>Graphite</td>
<td>$3.0 \times 10^{15}$</td>
<td></td>
<td>$5.41 \times 10^{14}$</td>
<td>$5.68 \times 10^{14}$</td>
</tr>
<tr>
<td>0.50</td>
<td>Graphite</td>
<td>$6.0 \times 10^{14}$</td>
<td></td>
<td>$1.06 \times 10^{14}$</td>
<td>$4.30 \times 10^{14}$</td>
</tr>
<tr>
<td>0.75</td>
<td>Graphite</td>
<td>$1.0 \times 10^{13}$</td>
<td></td>
<td>$4.62 \times 10^{12}$</td>
<td>$2.23 \times 10^{14}$</td>
</tr>
</tbody>
</table>
It is apparent in looking at Table VII that the $^{90}\text{Sr}$ concentration in the fuel matrix as given by Appel and Roos is larger than that which a source of $7.3 \times 10^9$ atoms/(cm$^3$.s) would produce in one year with no diffusion. Further investigation leads us to believe that Appel and Roos used a source of $7.3 \times 10^{11}$ which probably explains the difference between DASH and the Appel and Roos results.

A more realistic treatment of the source$^{13}$ allows for an increased source strength in later years caused by an increase in fuel particle failure rates. We assumed that the initial source ($S_0 = 7.3 \times 10^9$ atoms/cm$^3$.s) increases with time such that $S_0$ is used for the first year, $2S_0$ for the second year, $3S_0$ for the third year, etc. Numerical results for $^{90}\text{Sr}$ concentration are listed in Table VIII and shown in Fig. 18. The diffusion coefficient data of Table VI was employed in this calculation. The $^{90}\text{Y}$ concentration profiles are shown in Fig. 19. Comparison of the amount of $^{90}\text{Sr}$ produced with amount retained is the fuel and structural graphite indicates that even at six years almost half of this species is held up by the presence of the graphite. On the other hand, the $^{90}\text{Y}$ does not diffuse significantly but decays into $^{90}\text{Zr}$.

![Fig. 18. $^{90}\text{Sr}$ concentration profiles.](image-url)
TABLE VIII

$^{90}$Sr CONCENTRATION IN FUEL MATRIX WITH INCREASING SOURCE

<table>
<thead>
<tr>
<th>Time (y)</th>
<th>Total Source Units</th>
<th>Fuel Concentration (atom/cm$^3$)</th>
<th>Fuel Concentration if no Diffusion (atom/cm$^3$)</th>
<th>Fraction Retained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$1.80 \times 10^{17}$</td>
<td>$2.30 \times 10^{17}$</td>
<td>0.78</td>
</tr>
<tr>
<td>2</td>
<td>$1 + 2 = 3$</td>
<td>$4.74 \times 10^{17}$</td>
<td>$6.91 \times 10^{17}$</td>
<td>0.69</td>
</tr>
<tr>
<td>3</td>
<td>$3 + 3 = 6$</td>
<td>$8.45 \times 10^{17}$</td>
<td>$1.38 \times 10^{18}$</td>
<td>0.61</td>
</tr>
<tr>
<td>4</td>
<td>$4 + 6 = 10$</td>
<td>$1.27 \times 10^{16}$</td>
<td>$2.30 \times 10^{18}$</td>
<td>0.55</td>
</tr>
<tr>
<td>5</td>
<td>$5 + 10 = 15$</td>
<td>$1.75 \times 10^{18}$</td>
<td>$3.46 \times 10^{18}$</td>
<td>0.51</td>
</tr>
<tr>
<td>6</td>
<td>$6 + 15 = 21$</td>
<td>$2.27 \times 10^{18}$</td>
<td>$4.84 \times 10^{18}$</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Fig. 19. $^{90}$Y concentration profiles.
V. PROGRAM STRUCTURE

A. Role and Function of Subroutines

The DASH program consists of a driver routine, DASH1, and 34 functions and subroutines. The functions and subroutines can be divided into three classifications: primary, secondary, and graphic. The primary routines are those that are called directly by the controlling routine, DASH1, and perform major tasks. The secondary routines are utility routines called by the primary subroutines that do vector and matrix operations and function evaluations. The graphic routines are available for the generation of plots on 35-mm film.

1. Primary Routines

The 15 primary routines are discussed in the order in which they are called by DASH1.

a. INPA. The subroutine INPA reads and prints the basic nuclear data used in constructing the radioactive decay chain matrix. The input is stored locally so that it is readily available for subsequently called routines.

b. INPB. The subroutine INPB reads and prints the problem-dependent data.

c. GEOM. The subroutine GEOM calculates the geometric information required by the solution routines. From data supplied in INPB, this routine evaluates the mesh cell dimensions, area, and volume.

d. TEMADJ. The routine TEMADJ takes the temperature data supplied in INPB and fits it to a spline. From the fit, the routine calculates effective mesh cell temperatures for all the cells in the problem.

e. INPLT. The routine INPLT displays the calculational cells graphically. The mesh cells are illustrated with boundary condition and dimensional data. The purpose of this routine is to facilitate the debugging of the geometric input.

f. DIJADJ. The routine DIJADJ use the Arrhenius relation to temperature correct the input diffusion coefficients on a cell-by-cell basis. The temperatures calculated in TEMADJ are used along with the activation energies and diffusion coefficients read by INPB.

g. BCONL. The routine BCONL is used to establish the left-hand spatial boundary condition. Based on input data a modified value of $B_k$, Eq. (12), is evaluated for Eq. (9). The modified source, Eq. (13), due to the left boundary is also determined in this routine.
The routine MAKLAM utilizes the nuclear data from INPA to construct the radioactive decay matrix, Eq. (2).

The routine BIGEL constructs all the matrices necessary for the matrix $A$, Eq. (19), except $R_k$. This determination is carried out on a cell-by-cell basis.

The routine MAKEB assembles the matrix $B$. It takes the matrices created by BIGEL, multiplies them by the inverse volume element matrix, and inserts them in the matrix $B$.

The routine BCONR is used to establish the right-hand spatial boundary condition. Based on input data a modified value of $A_k$, Eq. (14), is evaluated for Eq. (9). The modified source, Eq. (15), due to the right boundary is also determined in this routine.

The subroutine SOLVER operates on the matrix generated by MAKEB to calculate the two matrix operators, $D(Bt)$ and $e^{Bt}$. The recursion relations discussed in App. A are part of this routine.

The routine MAKVOL assembles the diagonal volume element matrix, Eq. (20), used in FSOLVE.

The subroutine FSOLVE uses the operators calculated in SOLVER, the initial concentration vector, and the diagonal volume matrix to evaluate the time-dependent spatial concentrations according to Eq. (25). This routine is evaluated for each time interval specified in INPB.

The routine CONCPLT prints the results from FSOLVE in a detailed manner as a function of time and space point in either terminal or line printer format.

2. Secondary Routines
There are 14 secondary routines in DASH. These routines do utility operations such as vector and matrix operations, curve fitting, and function evaluation.

a. The general mathematic routines are listed below.

- **SCALAR** - Multiplies a local matrix by a scalar.
- **SCAECS** - Multiplies an extended core storage matrix by a scalar.
- **IFACT** - Evaluates factorials.
- **GENID** - Generates an identity matrix.
- **MATMOV** - Equivalences two local matrices.
- **MOVECS** - Equivalences two extended core storage matrices.
• MATMPY - Multiplies combinations of local vectors and local matrices.
• MPYEC3 - Multiplies combinations of extended core storage vectors and matrices.
• MPYEC1 - Multiplies combinations of local and extended core storage vectors and matrices.

b. The specialized input and output routines are listed below.
• PRIM - Prints a local matrix.
• PRIMES - Prints an extended core storage matrix.
• PRIV - Prints a local vector.
• REAG - Reads floating point data.
• REAI - Reads integer data.

c. There is one special purpose secondary routine.
• WXSEC - Collapses multigroup cross sections by flux weighting.

3. Graphics

The graphic routines generate 35-mm-film output in the form of plots of the calculated results for each time step in the problem. The plots make use of the DISSPLA* system which should facilitate the transfer to other computer centers. The plotting is done entirely in subroutine DRAW. The plots can be deleted without affecting the remainder of the code.

• DRAW - Controls the plotting of time-dependent results.

The DISSPLA routines employed are
• G PLOT - Device-independent initialization routine.
• BGNPL - Begins a plot.
• HEIGHT - Sets the basic character height.
• TITLE - Draws axes and titles.
• GRAF - Scales axes.
• CURVE - Draws a curve.
• ENDPL - Ends a plot.
• DONEPL - Plot termination.

*DISSPLA is a proprietary software product developed by Integrated Software Systems Corporation, San Diego, CA. It is available at about 200 computer installations.
B. Program Flow

The flow of the DASH program is illustrated in Fig. 20. The name of the primary subroutine involved in a given step is enclosed by parenthesis.

C. DASH Input Instructions

The DASH input is contained in 17 cards which are divided into 4 sets. The first set consists of card 0, which establishes the print options. The second set consists of cards 1 and 2 and defines the nuclear decay chains.
Cards 3 and 4 compose the third card set, which contains the multigroup cross-section data. The fourth set, cards 5-16, defines the physical characteristics of the system being evaluated.

The specific data for the four sets are detailed in Table IX. The user should note that if words 3 and/or 4 of card 1 are negative, 1 or 2 branching ratio cards, card 2 must follow card 1 before the next card 1. It should also be noted that cards 2 and 3 and cards 4 and 5 are separated by a blank card.

The diffusion coefficients are input as two matrices DIJO and AIJS. The full diffusion coefficients are:

\[
D = D_0 e^{-\frac{A}{RT}}
\]

\[
= DIJO*\text{EXP}(- AIJO/(R*T))
\]

(38)

For each element of DIJO and AIJO, values must be supplied for each isotope in each material.
**TABLE IX**

**DASH INPUT INSTRUCTIONS**

<table>
<thead>
<tr>
<th>CARD</th>
<th>WORD</th>
<th>SYMBOL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NPRINT</td>
<td>I4</td>
<td></td>
<td>PRINT OPTIONS</td>
</tr>
<tr>
<td>2</td>
<td>NPRINT</td>
<td>I4</td>
<td></td>
<td>0/1 LINE PRINTER/TERMINAL</td>
</tr>
<tr>
<td>2</td>
<td>NPRINT</td>
<td>I4</td>
<td></td>
<td>0/1 NO PLOT/ PLOT</td>
</tr>
</tbody>
</table>

| 1    | NANMAT(I,1)| A7     |        | NUCLIDE NAME                        |
| 2    | NANMAT(I,2)| I4     |        | ID NUMBER                           |
| 3    | NANMAT(I,3)| I4     |        | DECAY PARENT 1                      |
| 4    | NANMAT(I,4)| I4     |        | DECAY PARENT 2                      |
| 5    | NANMAT(I,5)| I4     |        | CAPTURE PARENT 1                    |
| 6    | NANMAT(I,6)| I4     |        | CAPTURE PARENT 2                    |
| 7    | NANMAT(I,7)| I4     |        | N-2N PARENT                         |
| 8    | NANMAT(I,8)| I4     |        | N-ALPHA PARENT                      |
| 9    | NANMAT(I,9)| I4     |        | N-P PARENT                          |
| 10   | ANMAT I,1  | E12.5  |        | DECAY CONSTANT (1/s)                |

| 2    | BRV(IBM)   | E12.5  |        | BRANCHING RATIO                     |

| 3    | NXSEC(I,1) | A6     |        | TITLE 1                             |
| 2    | NXSEC(I,2) | A6     |        | TITLE 2                             |
| 3    | NXSEC(I,3) | I4     |        | NUMBER OF GROUPS                    |
| 4    | NXSEC(I,4) | I4     |        | NUCLIDE ID                          |

<p>| 4    | NXSEC(I,3) | E12.5  |        | CROSS SECTION DATA                  |
| 1    | XSEC(I,KX,1)| E12.5  |        | SIGMA N-2N                          |
| 2    | XSEC(I,KX,2)| E12.5  |        | SIGMA N-ALPHA                       |
| 3    | XSEC(I,KX,3)| E12.5  |        | SIGMA N-P                           |
| 4    | XSEC(I,KX,4)| E12.5  |        | MICROSCOPIC CROSS SECTIONS (cm²)    |</p>
<table>
<thead>
<tr>
<th>CARD</th>
<th>WORD</th>
<th>SYMBOL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>NCELLS</td>
<td>I4</td>
<td>PROBLEM RELATED DATA</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TINT</td>
<td>E12.6</td>
<td>Number of cells in problem</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NGEOM</td>
<td>I4</td>
<td>1/2/3 Slab/Cylinder/Sphere</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NBCL</td>
<td>I4</td>
<td>Left boundary condition 1/2 reflected/concentration specified</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>NBCR</td>
<td>I4</td>
<td>Right boundary condition 1/2 reflected/concentration specified</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>NTEMPS</td>
<td>I4</td>
<td>Number of entries for specifying temperature field</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>IMATS</td>
<td>I4</td>
<td>Number of materials</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>IGP</td>
<td>I4</td>
<td>Number of neutron energy groups</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>TINC</td>
<td>E12.6</td>
<td>TIME STEP DATA</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TIMAX</td>
<td>E12.6</td>
<td>Initial time (days)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Number of time steps</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Time at end of problem (days)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DIST(1)</td>
<td>SPECIAL</td>
<td>DIMENSIONS</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>DIST(2)</td>
<td>SPECIAL</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DIST(3)</td>
<td>SPECIAL</td>
<td>First cell right boundary (cm)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DIST(NCELLS+1)</td>
<td>SPECIAL</td>
<td>Second cell right boundary (cm)</td>
<td></td>
</tr>
<tr>
<td>NCELLS+1</td>
<td>DIST(NCELLS+1)</td>
<td>SPECIAL</td>
<td>Last cell right boundary (cm)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>MATS(1)</td>
<td>SPECIAL</td>
<td>ASSIGN MATERIALS</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MATS(2)</td>
<td>SPECIAL</td>
<td>Material ID for cell 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPECIAL</td>
<td>Material ID for cell 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPECIAL</td>
<td>Material ID for cell NCELLS</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TEMPS(1)</td>
<td>SPECIAL</td>
<td>Dependent value for temperature field</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMPS(2)</td>
<td>SPECIAL</td>
<td>Temperature 1 (K)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMPS(NTEMPS)</td>
<td>SPECIAL</td>
<td>Temperature 2 (K)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMPS(NTEMPS)</td>
<td>SPECIAL</td>
<td>Temperature NTEMPS (K)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>TEMCOR(1)</td>
<td>SPECIAL</td>
<td>Independent value for temperature field</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMCOR(2)</td>
<td>SPECIAL</td>
<td>Coordinate of temperature 1 (cm)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMCOR(NTEMPS)</td>
<td>SPECIAL</td>
<td>Coordinate of temperature 2 (cm)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMCOR(NTEMPS)</td>
<td>SPECIAL</td>
<td>Coordinate of temperature NTEMPS (cm)</td>
<td></td>
</tr>
</tbody>
</table>
TABLE IX (cont)

<table>
<thead>
<tr>
<th>CARD</th>
<th>WORD</th>
<th>SYMBOL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>Diffusion Matrix ( \text{(cm}^2/\text{s}) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>DIJO(1,1,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>DIJO(2,1,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO</td>
<td>DIJO(ISO,1,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO + 1</td>
<td>DIJO(1,2,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO*ISO</td>
<td>DIJO(ISO,ISO,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO*ISO+1</td>
<td>DIJO(1,1,2)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td>Activation Energy Matrix ( \text{(cal/mole)} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>AIJS(1,1,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>AIJS(2,1,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO</td>
<td>AIJS(ISO,1,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO + 1</td>
<td>AIJS(1,2,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO*ISO</td>
<td>AIJS(ISO,ISO,1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO*ISO+1</td>
<td>AIJS(1,1,2)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>Fluxes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>PHI(1,N)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>PHI(2,N)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCELLS</td>
<td>PHI(NCELLS,N)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCELLS + 1</td>
<td>PHI(1,N+1)</td>
<td>SPECIAL</td>
</tr>
<tr>
<td>CARD</td>
<td>WORD</td>
<td>SYMBOL</td>
<td>FORMAT</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>14</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>CONBOU(1,1)</td>
<td>SPECIAL</td>
<td>Left boundary concentrations</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>CONBOU(2,1)</td>
<td>SPECIAL</td>
<td>Specie 1 left boundary concentration (atoms/cc)</td>
</tr>
<tr>
<td></td>
<td>ISO</td>
<td>CONBOU(ISO,1)</td>
<td>SPECIAL</td>
<td>Specie ISO left boundary concentration (atoms/cc)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>CONBOU(1,2)</td>
<td>SPECIAL</td>
<td>Right boundary concentrations</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>CONBOU(2,2)</td>
<td>SPECIAL</td>
<td>Specie 1 right boundary concentrations (atoms/cc)</td>
</tr>
<tr>
<td></td>
<td>ISO</td>
<td>CONBOU(ISO,2)</td>
<td>SPECIAL</td>
<td>Specie ISO right boundary concentrations (atoms/cc)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>CONINT(1)</td>
<td>SPECIAL</td>
<td>Initial concentration</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>CONINT(2)</td>
<td>SPECIAL</td>
<td>Initial concentration cell 1 specie 1 (atoms/cc)</td>
</tr>
<tr>
<td></td>
<td>ISO</td>
<td>CONINT(ISO)</td>
<td>SPECIAL</td>
<td>Initial concentration cell 1 specie ISO (atoms/cc)</td>
</tr>
<tr>
<td></td>
<td>ISO + 1</td>
<td>CONINT(ISO+1)</td>
<td>SPECIAL</td>
<td>Initial concentration cell 2 specie 1</td>
</tr>
<tr>
<td></td>
<td>ISO+NCELLS</td>
<td>CONINT(ISO+NCELLS)</td>
<td>SPECIAL</td>
<td>Initial concentration cell NCELLS specie ISO</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>SOURCE(1)</td>
<td>SPECIAL</td>
<td>Source cell 1 specie 1 (atoms/s)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>SOURCE(2)</td>
<td>SPECIAL</td>
<td>Source cell 1 specie 2 (atoms/s)</td>
</tr>
<tr>
<td></td>
<td>ISO</td>
<td>SOURCE(ISO)</td>
<td>SPECIAL</td>
<td>Source cell 1 specie ISO (atoms/s)</td>
</tr>
<tr>
<td></td>
<td>ISO+1</td>
<td>SOURCE(ISO+1)</td>
<td>SPECIAL</td>
<td>Source cell 2 specie 1 (atoms/s)</td>
</tr>
<tr>
<td></td>
<td>ISO+NCELLS</td>
<td>SOURCE(ISO+NCELLS)</td>
<td>SPECIAL</td>
<td>Source cell NCELLS specie ISO (atoms/s)</td>
</tr>
</tbody>
</table>
There are two special read formats. One is for integer data 6(I1, I2, I9), one for floating point data 6(I1, I2, E9.3). In each of these formats the first integer field, I1, designates the options listed in Table X. The second integer field, I2, controls the execution of the option, and the remainder of the field, I9 or E9.3, is for the input data. All data blocks read with these formats must be ended with a 3 in the I1 field after the last word of the block.

TABLE X
SPECIAL READ FORMAT OPTIONS

<table>
<thead>
<tr>
<th>Value of I1</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/blank</td>
<td>No action</td>
</tr>
<tr>
<td>1</td>
<td>Repeat data word in 9 field number of times indicated in I2 field.</td>
</tr>
<tr>
<td>2</td>
<td>Place number of linear interpolants indicated in I2 field between data word in 9 field and data word in next 9 field (not allowed for integers).</td>
</tr>
<tr>
<td>3</td>
<td>Terminate reading of the data block. A 3 must follow last data word of all blocks.</td>
</tr>
</tbody>
</table>

D. Machine Requirements
The DASH code requires both 35-mm-film hardware for graphics and the large core memory (LCM) capabilities of a CDC-7600. DASH was designed to operate on a CDC-7600 using the FTN compiler. The code is listed in App. B.

VI. DASH TEST PROBLEM
To demonstrate the application of the DASH code to solving a problem, a two-specie, three-material sample problem has been defined. The absorbent is a slab 5 cm thick consisting of three equal material regions. Initially, there is no diffusant in the absorbent. The material data for the two materials is summarized in Table XI. The test problem was run for 10 days with the results tabulated every 2 days. A detailed listing of the input and output is given in App. C. The graphic output is given here (Figs. 21 and 22). This problem requires approximately 5.5 CPU seconds of CDC-7600 time.
TABLE XI
SAMPLE PROBLEM DATA

<table>
<thead>
<tr>
<th>DIFFUSANT</th>
<th>DECAY CONSTANT ((s^{-1}))</th>
<th>DIFFUSION COEFFICIENT ((cm^2/s))</th>
<th>BOUNDARY CONCENTRATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>(8.0225 \times 10^{-7})</td>
<td>5.426 \times 10^{-6}</td>
<td>1.266 \times 10^{-5}</td>
</tr>
<tr>
<td>B</td>
<td>(1.6045 \times 10^{-6})</td>
<td>2.713 \times 10^{-6}</td>
<td>6.330 \times 10^{-6}</td>
</tr>
</tbody>
</table>

Fig. 21. Sample problem results for Diffusant A.

Fig. 22. Sample problem results for Diffusant B.

ACKNOWLEDGMENTS

The authors would like to thank Charles A. Anderson of LASL group Q-13 and James M. Hyman of LASL group T-7 for their helpful discussions.
REFERENCES


APPENDIX A
MATRIX OPERATOR EVALUATION

The time-dependent equation to be solved using the matrix operator method is

$$\frac{d\vec{x}}{dt} = \mathbf{B}\vec{x} + \vec{g}. \quad (A-1)$$

If the matrix \( \mathbf{B} \) is constant in the time interval \((0, t)\), we may construct the matricant \( \Omega^t_0 (\mathbf{B}) \), Eq. (A-2), using the Volterra method of the multiplicative integral.

$$\Omega^t_0 (\mathbf{B}) = \exp \left[ \int_0^t \mathbf{B}(s) \, ds \right] = \exp [\mathbf{B}t]. \quad (A-2)$$

The solution to Eq. (A-1) is given by

$$\vec{x}(t) = \Omega^t_0 (\mathbf{B}) \vec{x}(0) + \int_0^t dt' \mathbf{K}(t, t') \vec{g}(t'). \quad (A-3)$$

where

$$\mathbf{K}(t, t') = \Omega^t_0 (\mathbf{B}) \left[ \Omega^{t'}_0 (\mathbf{B}) \right]^{-1}. \quad (A-4)$$

Substituting Eq. (A-2) into Eqs. (A-3) and (A-4) gives

$$\vec{x}(t) = e^{\mathbf{B}t} \vec{x}(0) + e^{\mathbf{B}t} \int_0^t dt' e^{-\mathbf{B}t'} \vec{g}(t'). \quad (A-5)$$

Assuming that \( \vec{g}(t) = \vec{\dot{g}} \) is constant over the interval \((0, t)\), Eq. (A-5) becomes

$$\vec{x}(t) = e^{\mathbf{B}t} \vec{x}(0) + \vec{\dot{g}}^{-1} (e^{\mathbf{B}t} - \mathbf{I}) \vec{g}. \quad (A-6)$$

Defining the matrix operator \( \mathbf{D}(\mathbf{C}) \) by

$$\mathbf{D}(\mathbf{C}) = \mathbf{C}^{-1} (e^{\mathbf{C}} - \mathbf{I}) \quad (A-7)$$
or

$$t D(Bt) = B^{-1} (e^{Bt} - I),$$  \hspace{1cm} \text{(A-8)}

Eq. (A-6) becomes

$$\dot{X}(t) = \dot{X}(0) + t B D(Bt) \dot{X}(0) + t D(Bt) \dot{g}$$

$$= \dot{X}(0) + t D(Bt) \left[ B \dot{X}(0) + \dot{g} \right].$$  \hspace{1cm} \text{(A-9)}

Note that the matrix operator $D(C)$ defined by

$$D(C) = C^{-1} (e^C - I) = \sum_{n=0}^{\infty} \frac{C^n}{(n+1)!}$$  \hspace{1cm} \text{(A-10)}

exists even if $C = Bt$ is singular. Although the eigenvalues of $e^C$ are bound by unity, and the eigenvalues of $C$ are bound, but not necessarily by unity, the direct evaluation of $D(C)$ would prove difficult computationally if Eq. (A-10) is used. The matrix $C$ can be scaled so that the eigenvalues are bound by unity. Define

$$H = 2^{-p} C,$$  \hspace{1cm} \text{(A-11)}

where $p$ is determined by

$$||H|| < \frac{1}{2}$$  \hspace{1cm} \text{(A-12)}

or $^{4,15}$

$$p > \ln \left( \sum_{ij} |C_{ij}|^2 \right) / (2 \ln 2).$$  \hspace{1cm} \text{(A-13)}

We approximate the $D(H)$ matrix operator by a finite number of terms $M$ using Eq. (A-10).

$$D^M(H) \approx \sum_{n=0}^{M} \frac{H^n}{(n+1)!}$$  \hspace{1cm} \text{(A-14)}
The value of $M$ is chosen such that the excluded terms have an error less than $\varepsilon$, or

$$\left(\frac{H}{M+2}\right)^{M+1} < \frac{1}{2^M + 1} \frac{1}{(M+2)!} < \varepsilon. \quad (A-15)$$

Knowing $D(H)$ we may recur upwards by powers of 2 in $H$ to find $D(C)$ where $C = 2^PH$, using the recursion relation

$$D(2^P + 1H) = D(2^PH) \left[ I + \frac{1}{2} (2^PH)D(2^PH) \right]. \quad (A-16)$$

The recursion relation is readily proven by induction. Define

$$D(H) = H^{-1} (e^H - I) \quad (A-17)$$

and

$$C = 2^PH. \quad (A-18)$$

Clearly if $p = 0$, $D(C)$ is equal to $D(H)$. If $p = 1$, Eq. (A-16) yields

$$D(C) = D(2H) = (2H)^{-1} (e^{2H} - I)$$

$$= H^{-1} (e^H - I) \left( \frac{e^H + I}{2} \right) \quad (A-19)$$

$$= D(H) \left[ I + \frac{1}{2} HD(H) \right].$$

Induction based on Eq. (A-19) yields

$$D(2^PH) = D(2^P - 1H) \left[ I + \frac{1}{2} (2^P - 1H)D(2^P - 1H) \right]. \quad (A-20)$$

Assume Eq. (A-20), which is true for $p = 0$ and 1, is true for $p = n$. Evaluate $D(2^n + 1H)$ as
\[
D(2^n + 1^H) = (2^n + 1^H)^{-1} \left( e^{2^n + 1^H} - I \right)
\]
\[
= (2^n)^{-1} \left( e^{2^n} - I \right) \frac{1}{2^n} \left( e^{2^n} + I \right)
\]
\[
= D(2^n)^{-1} \left[ I + \frac{1}{2^n} D(2^n) \right].
\]

Since Eq. (A-20) is true for \( p = 0 \) and 1, and if it is assumed true for \( p = n \), it is true for \( p = n + 1 \); then by transfinite induction it is true for all \( p \).

APPENDIX B

DASH CODE LISTING
(LASL Code LP-1055)

PROGRAM DASH1 (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)

DASH - A MULTICOMPONENT TIME DEPENDENT CONCENTRATION DIFFUSION WITH RADIOACTIVE DECAY PROGRAM.

BY COURTNEY E. APPERSON, JR.
LUCY M. CARRUTHERS
JUDITH F. SHINN
ENERGY DIVISION
LOS ALAMOS SCIENTIFIC LABORATORY
AND
CLARENCE E. LEE
TEXAS A AND M UNIVERSITY
MARCH 1979

THE PROGRAM DASH CALCULATES THE TRANSIENT CONCENTRATION OF MULTIPLY DIFFUSING SPECIES WITH RADIOACTIVE DECAY USING FINITE DIFFERENCE AND EXPONENTIAL OPERATOR TECHNIQUES.

THIS IS THE FOURTH VERSION OF DASH. IT WAS CREATED ON 15 MARCH 1979.

RECOGNIZING THAT GRAPHICS HARDWARE AND SOFTWARE ARE USUALLY UNIQUE TO A PARTICULAR INSTALLATION THE GRAPHICS PACKAGE IN DASH CAN BE READILY DELETED WITHOUT EFFECTING THE REMAINDER OF THE CODE.

INPUT INSTRUCTIONS FOR THE CODE ARE--

CARD WORD SYMBOL FORMAT DESCRIPTION
0 PRINT OPTIONS
1 NPRINT I4 0/1 LINE PRINTER/TERMINAL
2 NPLOT I4 0/1 NO PLOT/ PLOT
1 --ONE CARD PER NUCLIDE--

   1  NANMAT(I,1)    A7  NUCLIDE NAME
   2  NANMAT(I,2)    I4  ID NUMBER
   3  NANMAT(I,3)    I4  DECAY PARENT 1
   4  NANMAT(I,4)    I4  DECAY PARENT 2
   5  NANMAT(I,5)    I4  CAPTURE PARENT 1
   6  NANMAT(I,6)    I4  CAPTURE PARENT 2
   7  NANMAT(I,7)    I4  N-2N PARENT
   8  NANMAT(I,8)    I4  N-ALPHA PARENT
   9  NANMAT(I,9)    I4  N-P PARENT
  10  ANMAT(I,1)   E12.5  DECAY CONSTANT (1/S)

2 --ONE CARD PER BRANCH FOR EACH NEGATIVE VALUE OF NANMAT(I,3-9)--

   1  BRV(IBR)   E12.5  BRANCHING RATIO

--BLANK CARD AFTER LAST SET OF CARDS 1 AND 2--

3 --NXSEC(II,3) CARDS--

   1  NXSEC(II,1)  A6  TITLE 1
   2  NXSEC(II,2)  A6  TITLE 2
   3  NXSEC(II,3)  I4  NUMBER OF GROUPS
   4  NXSEC(II,4)  I4  NUCLIDE ID

4 --NXSEC(II,3) CARDS--

   1  XSEC(II,KX,1)  E12.5  SIGMA N-GAMMA
   2  XSEC(II,KX,2)  E12.5  SIGMA N-2N
   3  XSEC(II,KX,3)  E12.5  SIGMA N-ALPHA
   4  XSEC(II,KX,4)  E12.5  SIGMA N-P

--BLANK CARD AFTER LAST SET OF CARDS 3 AND 4--

5 --PROBLEM RELATED DATA--

   1  NCELLS  I4  NUMBER OF CELLS IN PROBLEM
   2  NGEOM  I4  1/2/3 SLAB/CYLINDER/Sphere
   3  NBCL  I4  LEFT BOUNDARY CONDITION
   4  NBCR  I4  RIGHT BOUNDARY CONDITION
   5  NTEMPS  I4  NUMBER OF ENTRIES FOR SPECIFYING TEMPERATURE FIELD
   6  IMATS  I4  NUMBER OF MATERIALS
   7  IGF  I4  NUMBER OF NEUTRON ENERGY GROUPS

6 --TIME STEP DATA--

   1  TINT  E12.6  INITIAL TIME (DAYS)
   2  TINC  E12.6  NUMBER OF TIME STEPS
   3  TIMAX  E12.6  TIME AT END OF PROBLEM (DAYS)

7 --GEOMETRY (CM)--

   1  DIST(1)  SPECIAL  0.0  FIRST CELL RIGHT BOUNDARY
   2  DIST(2)  SPECIAL
   3  DIST(3)  SPECIAL
   4  DIST(NCELLS+1)  SPECIAL  LAST CELL RIGHT BOUNDARY

NCELLS+1 DIST(NCELLS+1) SPECIAL
ASSIGN MATERIALS

MATS(1) SPECIAL
MATS(2) SPECIAL
MATS(NCELLS) SPECIAL
NCELLS SPECIAL

TEMP(1) SPECIAL
TEMP(2) SPECIAL
NTEMP(S) SPECIAL

TEMCOR(1) SPECIAL
TEMCOR(2) SPECIAL
NTEMCOR(S) SPECIAL

DIJO(I,I,N) SPECIAL
DIJO(2,I,N) SPECIAL
DIJO(ISO,1,N) SPECIAL
DIJO(ISO,2,N) SPECIAL
DIJO(ISO,ISO,N) SPECIAL

AIJS(I,I,N) SPECIAL
AIJS(2,I,N) SPECIAL
AIJS(ISO,1,N) SPECIAL
AIJS(ISO,2,N) SPECIAL
AIJS(ISO,ISO,N) SPECIAL

PHI(1,N) SPECIAL
PHI(2,N) SPECIAL
PHI(ISO,N) SPECIAL
PHI(ISO,N+1) SPECIAL

CONBOU(1,1) SPECIAL
CONBOU(1,2) SPECIAL
CONBOU(ISO,1) SPECIAL

PHI(NCELLS,N) SPECIAL
PHI(NCELLS+1,N) SPECIAL

GROUP N FLUX IN CELL 1
GROUP N FLUX IN CELL 2
GROUP N FLUX IN CELL NCELLS
GROUP N+1 FLUX IN CELL 1

SPECIE 1 LEFT BOUNDARY CONC
SPECIE 2 LEFT BOUNDARY CONC
SPECIE ISO LEFT BOUNDARY CONC

NCELLS SPECIAL
NCELLS+1 SPECIAL

GROUP N FLUX IN CELL 1
GROUP N FLUX IN CELL NCELLS
GROUP N+1 FLUX IN CELL 1

SPECIE 1 LEFT BOUNDARY CONC
SPECIE 2 LEFT BOUNDARY CONC
SPECIE ISO LEFT BOUNDARY CONC

NCELLS SPECIAL
NCELLS+1 SPECIAL

GROUP N FLUX IN CELL 1
GROUP N FLUX IN CELL NCELLS
GROUP N+1 FLUX IN CELL 1

SPECIE 1 LEFT BOUNDARY CONC
SPECIE 2 LEFT BOUNDARY CONC
SPECIE ISO LEFT BOUNDARY CONC

NCELLS SPECIAL
NCELLS+1 SPECIAL
--SUPPLY ONLY IF NBCR = 2--

RIGHT BOUNDARY CONCENTRATIONS
(ATOMS/CC)

1 CONBOU(1,2) SPECIAL SPECIE 1 RIGHT BOUNDARY CONC
2 CONBOU(2,2) SPECIAL SPECIE 2 RIGHT BOUNDARY CONC
ISO CONBOU(ISO,2) SPECIAL SPECIE ISO RIGHT BOUNDARY CONC

INITIAL CONCENTRATION
(ATOMS/CC)

1 CONINT(1) SPECIAL INITIAL CONC CELL 1 SPECIE 1
2 CONINT(2) SPECIAL INITIAL CONC CELL 1 SPECIE 2
ISO CONINT(ISO) SPECIAL INITIAL CONC CELL 1 SPECIE ISO
ISO+1 CONINT(ISO+1) SPECIAL INITIAL CONC CELL 2 SPECIE 1
ISO*NCELLS CONINT(ISO*NCELLS) SPECIAL INITIAL CONC CELL NCELLS SPECIE ISO

SOURCE
(ATOMS/SEC)

1 SOURCE(1) SPECIAL SOURCE CELL 1 SPECIE 1
2 SOURCE(2) SPECIAL SOURCE CELL 1 SPECIE 2
ISO SOURCE(ISO) SPECIAL SOURCE CELL 1 SPECIE ISO
ISO+1 SOURCE(ISO+1) SPECIAL SOURCE CELL 2 SPECIE 1
ISO*NCELLS SOURCE(ISO*NCELLS) SPECIAL SOURCE CELL NCELLS SPECIE ISO

SPECIAL FORMATS

THERE ARE TWO SPECIAL READ FORMATS. ONE IS FOR INTEGER DATA
6(I1,I2,I9) AND ONE IS FOR FLOATING POINT DATA 6(I1,I2,E9-3).
IN EACH WORD OF BOTH THESE FORMATS, THE FIRST INTEGER FIELD, I1,
DESIGNATES THE OPTIONS LISTED BELOW. THE SECOND INTEGER FIELD,
I2, CONTROLS THE EXECUTION OF THE OPTION, AND THE REMAINDER OF
THE FIELD, I9 OR E9-3, IS FOR THE INPUT DATA. ALL DATA BLOCKS
READ WITH THESE FORMATS MUST BE ENDED WITH A 3 IN THE I1 FIELD
AFTER THE LAST WORD OF THE BLOCK.

--OPTIONS FOR SPECIAL READ FORMATS--

VALUE OF I1 NATURE OF OPTION
0 OR BLANK NO ACTION
1 REPEAT DATA WORD IN 9 FIELD NUMBER OF TIMES
   INDICATED IN I2 FIELD.
2 PLACE NUMBER OF LINEAR INTERPOLANTS INDICATED
   IN I2 FIELD BETWEEN DATA WORD IN 9 FIELD AND
   DATA WORD IN NEXT 9 FIELD. NOT ALLOWED FOR
   INTEGERS.
3 TERMINATE READING OF DATA BLOCK. A 3 MUST
   FOLLOW LAST DATA WORD OF ALL BLOCKS.

DIMENSION NANMAT(NISO1,9),ANMAT(NISO1,2),ERV(NBR),NXSEC(NXSP,4)
1,XSEC,NXP,NGP,4),NXP(CELL)
DIMENSION DIST(NCELL1),MATS(NCELL),TEMPS(NTEM),TEMCOR(NTEM),
1 DIJO(NISO,NIISO,NMAT),ALOS(NISO,NIISO,NMATS),DUM1(NTEM),DUM2(NTEM)
DIMENSION DUM1(NN),DUM22(NN),ALAN(NISO,NIISO),BB(NISO,NIISO)
WHERE NN IS THE GREATER OF NISO*NISO AND NCELL
DIMENSION DELR(NCELL),AREA(NCELL),VOL(NCELL),RBAR(NCELL),
1 PHI(NCELL,NGP),W(NTEM),DUM3(NTEM),DIFFK(NISO,NISO),
2 DIFFR(NISO,NISO),DAPLOT(NISO,NCELL,NTIME)
COMMON /SOLS/ BIGE(MM,MM),B(MM,MM),C(MM,MM),D(MM,MM),E(MM,MM),
1 F(MM,MM)
DIMENSION SOURCE(MM), DIFDUM1(NISO,NISO), DIFDUM2(NISO,NISO),
1 CONBOU(NISO,2), IPVT(NISO), AK(NISO,NISO), BK(NISO,NISO),
2 C(NISO,NISO), CONINT(MM), CONCEN(MM), DUM33(MM)
WHERE MM=MISO*NCELL
DIMENSION NANMAT(6,9), ANMAT(6,2), BRV(10), NXSEC(11,4), XSEC(10,4
1 4,1), NPP(25)
DIMENSION DIST(26), MATS(25), TEMPS(25), TEMCOR(25), DIJO(5,5,5),
1 AIJS(5,5,5), DUM1(25), DUM2(25)
DIMENSION DUM11(100), DUM22(100), ALAM(5,5), BB(5,5)
DIMENSION DELR(25), AREA(25), VOL(25), RBAR(25), PHI(25,4), W(25),
1 DUM3(25), DIFFK(5,5), DIFFK1(5,5), DAPLOT(5,25,11)
COMMON /SOL,/ BIGB(125,125), BI(125,125), C(125,125), D(125,125), E
1 (125,125), F(125,125)
DIMENSION SOURC(25), DIFDUM1(5,5), DIFDUM2(5,5), CONBOU(5,2),
1 IPVT(5), AK(5,5), BK(5,5), C(5,5), CONINT(125), CONCEN(125),
2 DUM33(125)
LEVEL 2, BIGB, B, C, D, E, F
COMMON /I0/, NINP, NOUT, IER, NPRINT
COMMON /NUCDAT/, ISO, IBR, IXS, IGP
COMMON /TIMES/, TINT, TINC, TIMAX

C TAP65 IS INPUT UNIT
NINP=5
C TAP66 IS OUTPUT UNIT
NOUT=6
READ (NINP,100) NPRINT,NPLOT
NISO=5
NISO=1
N=100
C NN IS THE GREATER OF NISO*NISO AND NGP*NCELL
NBR=10
NXS=10
NXS=NXS+1
NGP=4
NCELL=25
NCEIL=NCEIL+1
MM=MISO*NCELL
NTEM=25
C NTEM MUST BE GREATER THAN OR EQUAL TO NCELL
NMAT=5
NTIME=11
CALL INPA (NANMAT, ANMAT, BRV, NXSEC, NXSEC, NPP, NCELL, NISO, NISO1, NISO)
CALL INPB (NCELLS, NGEOM, NCBR, NCB, NTEMPS, IMATS, DIST, MATS, PHI, TEMPS
1 TEMOR, DIIJO, AIJS, NCELL1, NCELL, NTEM, NISO, NMATS, NPP, DUM11, DUM22, NN
2 NCELLS, NCELL, NTEM, NISO, NMATS, NPP, DUM11, DUM22, NN
N=NCELLS+1
CALL GEOM (NGEOM, DIST, DELR, AREA, VOL, RBAR, N, NCELLS)
CALL INPL (NCELL, NCELL1, NCELL, NCELL, NCBR, MATS, DIST, NGEOM, NCELLS)
CALL PRIV(DIST, N, NCELL, NCELL1, 10H DISTANCES, 10H)
CALL PRIV(DIST, N, NCELLS, NCELL1, 10H DELR, 10H)
CALL PRIV(DELRE2, NCELL, NCELL1, 10H AREA, 10H)
CALL PRIV(VELOCITY, NCELL, NCELL1, 10H VOLUME, 10H)
CALL PRIV(RBAR, NCELL, NCELL1, 10H RBAR, 10H)
CALL TEMADJ (TEMPS, TEMOR, NTEMPS, RBAR, NCELLS, W, DUM1, DUM2, DUM3, NTEM
1)
CALL PRIV(TEMPS, NCELLS, NTEM, 10H MESH TEMP, 10HERATURES)
IF(IXS.EQ.0) GO TO 5
CALL PHIM(PHI, NCELLS, IGP, NCELL, NPP, 10H FLUXES, 10H)
5 CONTINUE
DO 20 I=1, MM
DO 10 J=1, MM
BIGE(I,J)=0.0
10 CONTINUE
20 CONTINUE
II=1
C MAKE ADJUSTMENTS FOR HOLLOW CYLINDER AND SPHERE
C IF (NCELLS.EQ.2.AND. NGEOM.GT.1) II=2
MAT=MATS(II)
CALL DJADJ (DIJO, AIJS, TEMPS, DIFFK, NISO, NMATS, NCELL, ISO, II, MAT)
CALL PRINT (DIFFK, ISO, ISO, ISO, ISO, 10H DIFFK, 10H)
NIC=ISO*(NCELLS+1-II)
CALL BCOLN (BK,AREA,DIFFK,DELR,SOURCE,CONBOU,NCELLS,ISO,NBCL,NGEOM
1,NISO,NIC,DUM11,DUM33)
C CALL PRIM(BK,ISO,ISO,ISO,ISO,10H EK
1,10H =)
NM1=NCELLS-1
ICOL=1
DO 30 I=II,NM1
C CALL MALKAM (NANMAT,ANMAT,XSEC,NXSEC,PHI,BRV,ALAM,BB,NISO1,NXS
1,NXSP,NBR,NOP,ISO,IXS,IGP,NCELL,1)
C CALL PRIM(ALAM,ISO,ISO,ISO,ISO,10H LAMBDA
1,10H =)
K=I+1
MAT=MATS(K)
C CALL DIOJ (DIJO,AIJS,TEMPS,DIFFK1,NISO,NMATS,NCELL,ISO,K,MAT)
C CALL PRIM(DIFFK1,ISO,ISO,ISO,ISO,10H DIFFK1
1,10H =)
C CALL BIGEL (DIFFK1,DIFFK1,DIFDUM1,DIFDUM2,ALAM,AK,BK,CKK,ISO,AREA
1,10H ,DELR,NCELLS,IPVT,DUM11,I)
C CALL PRIM(CKK,ISO,ISO,ISO,ISO,10H KK
1,10H =)
C CALL PRIM(AK,ISO,ISO,ISO,ISO,10H AK
1,10H =)
C CALL MAKEM (BIGB,AK,BK,CKK,NIC,ISO,VERTICEL,NCELLS,ICOL,II)
30 CONTINUE
C CONTINUE
CALL MALKAM (NANMAT,ANMAT,XSEC,NXSEC,PHI,BRV,ALAM,BB,NISO1,NXS
1,NXSP,NBR,NOP,ISO,IXS,IGP,NCELL,NCELLS)
C CALL PRIM(ALAM,ISO,ISO,ISO,ISO,10H LAMBDA
1,10H =)
C CALL BCOLN (AK,AREA,DIFFK,DELR,SOURCE,CONBOU,NCELLS,ISO,NBCL,NISO
1,NIC,DUM11,DUM22,BK,CKK,ALAM,VOL,DIFDK1)
C CALL PRIM(AK,ISO,ISO,ISO,ISO,10H AK
1,10H =)
C CALL PRIM(AK,ISO,ISO,ISO,ISO,10H AK
1,10H =)
C CALL PRIMV(SOURCE,NIC,NIC,10H SOURCE
1,10H )
C CALL MAKEBL (BIGB,AK,BK,CKK,NIC,ISO,VOL,NCELLS,ICOL,II)
C CALL PRIMES(BIGB,NIC,NIC,NIC,10H BIGB
1,10H )
DTIME=(TIMAX/TINC)*24.*3600.
NSTEP=TINC
C CALL SOLVER (BIGB,B,C,D,E,F,DTIME,NIC,NIC)
C CALL PRIMES(E,NIC,NIC,NIC,10H E
1,10H =)
C CALL PRIMES(D,NIC,NIC,NIC,10H D
1,10H =)
NP=NCELLS-II+1
DO 50 J=1,NP
DO 40 I=1,ISO
I=ISO*(J-1)+1
DAPLOT(I,J,L)=CONINT(I1)
40 CONTINUE
C CONTINUE
CALL MAKVOL (B,NIC,VOL,NCELLS,II,ISO)
C CALL PRIMES(B,NIC,NIC,NIC,10H VOLMAT
1,10H )
DO 90 I=II,NSTEP
C CALL PSOLVE (B,C,D,E,DUM33,DTIME,NIC,CONINT,SOURCE,CONCEN)
DO 70 J=1,NP
DO 60 J1=1,ISO
I=ISO*(J-1)+J
DAPLOT(I,J,L)=CONCEN(I1)
60 CONTINUE
C CONTINUE
DO 80 J=1,NIC
CONINT(J)=CONCEN(J)
80 CONTINUE
C CONTINUE
NTIM=NSTEP+1
C CALL CONCIT (NANMAT,CONBOU,DAPLOT,ISO,NCELL,NISO1,NISO,NTIME,NP
1,NTIM,NPP,NBCL,NBCR,VOL)
IF (NPLLOT.NE.0) CALL DRAW (NANMAT,DELR,DAPLOT,NTIM,NP,CONCEN,DUM33
1,NISO,NCELLS,NCELL,NTIME,ISO,NISO1,CONBOU,DIST,NCELL1,NBCL,NBCL,NBCR)
STOP
C 100 FORMAT (214)
END
48
SUBROUTINE INPA (NANNAT,ANMAT,BRV,NXSEC,XSEC,NGP,NXS,NBR,NISO1,NXSP)

C INPUTA READS AND PRINTS THE NUCLEAR DATA

DIMENSION NANNAT(NISO1,9), ANMAT(NISO1,2), BRV(NBR), NXSEC(NXSP,4)

COMMON /NUCDAT/ ISO, IER, IXS, IGP
DATA NHJ /6H

I=1
IBR=1
II=1

READ NUCLEAR DATA
10 READ (NINP,130) (NANNAT(I,J),J=1,9),ANMAT(I,1)
IF (NANNAT(I,1).EQ.NHJ) GO TO 40

TEST FOR BRANCHING RATIOS
20 READ (NINP,140) BRV(IBR)
IF (BRV(IBR).GT.60) GO TO 40
IF (NANNAT(I,J)).GT.60) GO TO 60

READ CROSS SECTION DATA
50 READ (NINP,150) (NXSEC(II,J),J=1,4)
IF (NXSEC(II,1).EQ.NHJ) GO TO 70
IGP=NXSEC(II,3)
DO 60 J=1,IGP
READ (NINP,140) (XSEC(II,J,JJ),JJ=1,4)
60 CONTINUE
II=II+1
GO TO 50

PRINT DECAY DATA
70 PRINT (I,160) (NANNAT(J,JJ),JJ=1,9),ANMAT(J,1)
LCNT=LCNT+2
IF (LCNT.GE.60) WRITE (NOUT,160)

PRINT BRANCHING RATIOS
80 IF (IBR.EQ.0) GO TO 100
WRITE (NOUT,160)
WRITE (NOUT,180)
WRITE (NOUT,200)
WRITE (NOUT,220)
DO 90 J=1,IBR
WRITE (NOUT,230) HBRP(1,J),HBRP(2,J),HBRP(3,J)
90 CONTINUE

PRINT CROSS SECTIONS
100 IF (II.EQ.0) GO TO 120
WRITE (NOUT,160)
WRITE (NOUT,180)
WRITE (NOUT,220)
DO 110 J=1,II
WRITE (NOUT,250) (NXSEC(J,JN),JN=1,4)
110 CONTINUE

ISO=1
SUBROUTINE INPB (NCELL, NGEOM, NBCR, NBCL, NTEMPS, IMATS, DIST, MATS, PHI, TEMPS, TEMCOR, DIJO, AIJS, NCELL1, NCELL, NTEM, NISO, NMATS, NGP, DUM11, DUM22, NN, CONBOU, CONINT, SOURCE, MM)

DIMENSION DIST(NCELL1), MATS(NCELL), TEMPS(NTEM), TEMCOR(NTEM), DIJO(NISO, NISO, NMATS), AIJS(NISO, NISO, NMATS), DUM11(NN), DUM22(NN)

DIMENSION IGEOM(3)

COMMON /IO/ NINP, NOUT, IER, NPRINT
COMMON /NUCDAT/ ISO, IBR IXS, IGP
COMMON /TIME/ TINT, TINC, TMAX

DATA IGEOM(1), IGEOM(2), IGEOM(3) /9HSLAB.,9HCYLINDER.,9HSpher.

READ (NINP,130) NCELL, NGEOM, NBCR, NBCL, NTEMPS, IMATS, DIST, MATS, PHI, TEMPS, TEMCOR
PRINT (NOUT,140) IGEOM(NGEOM), NBCR

READ (NINP,150) TINT, TINC, TMAX

INDEX=ISO*ISO
DO 30 I=1,IMATS
CALL REAG (DUM11, INDEX, DIJO(I,J), AIJS(I,J), NCELL, NTEM, NISO, NMATS, NGP, DUM11)
DO 20 J=1,ISO
DO 10 JJ=1,ISO
IND=(J-1)*ISO+JJ
DIJO(JJ,J,J)=DUM11(IND)
AIJS(JJ,J,J)=DUM22(IND)
CONTINUE
CONTINUE
CONTINUE
IF (IXS.EQ.0) GO TO 60
INDEX=IGP*NCELL
CALL REAG (DUM11, INDEX, PHI(1:I), SOURCE(1:I), MM)
DO 80 I=1,2
DO 70 J=1,ISO
CONBOU(J,J)=0.0
CONTINUE
CONTINUE
IF (NBCL.NE.2) GO TO 100
CALL REAG (DUM11, ISO, 6HLEFT C,6HONCEN )
DO 90 I=1,ISO
CONBOU(I,1)=DUM11(I)
90 CONTINUE
100 IF (NBCR.NE.2) GO TO 120
   CALL REAG (DUM11,ISO,6HRIGHT,6HCONCEN)
   DO 110 I=1,ISO
   CONBOU(I,2)=DUM11(I)
110 CONTINUE
120 CONTINUE
   INDEX=ISO*NCELLS
   IF (NGEOM.GT.1.AND.NBCL.EQ.2) INDEX=INDEX-ISO
   CALL REAG (CONINT,INDEX,6HINITIA,6HCONC)
   INDEX=ISO*NCELLS
   CALL REAG (SOURCE,INDEX,6HSOURCE,6HINPUT)
   RETURN
C
130 FORMAT (1814)
140 FORMAT (///,1X,*THE GEOMETRY FOR THIS PROBLEM IS A *,A9,/ 2X,*THE
150 FORMAT (/1X,'LEFT BOUNDARY CONDITION IS *=,A9,/,2X,*THE RIGHT BOUNDARY CONDITION 
C
160 FORMAT (3E12.6,112)
C
C
C
170 FORMAT (1814)
180 FORMAT (///,1X,*THE GEOMETRY FOR THIS PROBLEM IS A *,A9,/ 2X,*THE
190 FORMAT (/1X,'LEFT BOUNDARY CONDITION IS *=,A9,/,2X,*THE RIGHT BOUNDARY CONDITION 
C
200 FORMAT (3E12.6,112)
C
C
C
210 FORMAT (1814)
220 FORMAT (///,1X,*THE GEOMETRY FOR THIS PROBLEM IS A *,A9,/ 2X,*THE
230 FORMAT (/1X,'LEFT BOUNDARY CONDITION IS *=,A9,/,2X,*THE RIGHT BOUNDARY CONDITION 
C
240 FORMAT (3E12.6,112)
C
C
C
250 FORMAT (1814)
260 FORMAT (///,1X,*THE GEOMETRY FOR THIS PROBLEM IS A *,A9,/ 2X,*THE
270 FORMAT (/1X,'LEFT BOUNDARY CONDITION IS *=,A9,/,2X,*THE RIGHT BOUNDARY CONDITION 
C
280 FORMAT (3E12.6,112)
C
C
C
290 FORMAT (1814)
300 FORMAT (///,1X,*THE GEOMETRY FOR THIS PROBLEM IS A *,A9,/ 2X,*THE
310 FORMAT (/1X,'LEFT BOUNDARY CONDITION IS *=,A9,/,2X,*THE RIGHT BOUNDARY CONDITION 
C
320 FORMAT (3E12.6,112)
GO TO 40
30 LOOP = (NCELLS/NBOX)
40 ICELL = NCELLS
ICOUNT = 0
DO 120 I = 1, LOOP
ICOUNT = ICOUNT + 1
IS =
IE = MINO(NBOX, ICELL)
IMM = IE - IS
IMM1 = IMM - I
IM2 = (IMM + 1) * 8 - 1
IM3 = IMM * 8 - 1
NDX = (1 * NBOX - NBOX) / IMM + 1
NDS = IMM + (1 * NBOX - NBOX)
NDSR = NDS + 1
NDR = NDSR + 1
ICELL = ICELL - NBOX
IF (LOOP.EQ.1) GO TO 100
IF (I.EQ.1) GO TO 70
WRITE (NOUT, 150) ISTR, (ISTR, J = 1, IM2)
WRITE (NOUT, 140) ISTR, (ISTR, J = 1, IMM)
WRITE (NOUT, 130) ISTR, (ISTR, J = 1, IMM1)
WRITE (NOUT, 120) ISTR, (ISTR, J = 1, IM2)
WRITE (NOUT, 110) ISTR, (ISTR, J = 1, IM3)
WRITE (NOUT, 100) ISTR, (ISTR, J = 1, IM2)
WRITE (NOUT, 90) ISTR, (ISTR, J = 1, IM3)
WRITE (NOUT, 80) ISTR, (ISTR, J = 1, IM2)
WRITE (NOUT, 70) ISTR, (ISTR, J = 1, IM3)
WRITE (NOUT, 60) ISTR, (ISTR, J = 1, IM2)
WRITE (NOUT, 50) ISTR, (ISTR, J = 1, IM3)
WRITE (NOUT, 40) ISTR, (ISTR, J = 1, IM2)
WRITE (NOUT, 30) ISTR, (ISTR, J = 1, IM3)
WRITE (NOUT, 20) ISTR, (ISTR, J = 1, IM2)
WRITE (NOUT, 10) ISTR, (ISTR, J = 1, IM3)
WRITE (NOUT, 0) ISTR, (ISTR, J = 1, IM2)
GO TO 120
SUBROUTINE CONCPLT (NANMAT, CONBOU, DAPLOT, ISO, NCELL, NISO1, NISO2, NTIM, NP, NPP, NBCR, VOL)

C DIMENSION NANMAT(NISO1,9), CONBOU(NISO2,2), DAPLOT(NISO, NCELL, NTIME 1), NPP(NECELL), VOL(NCELL)

DATA BNDRY /10HREFLECTED /

TIME=0.0

C TERMINAL OUTPUT
NBOX=4

C LINE PRINTER OUTPUT
IF (NPRINT.EQ.0) NBOX=6
NBOX1=NBOX-1
NP=NP-NBOX

IF (MOD(NP1 NBOX1)) 20,10,10
LOOP=NP1/NBOX1+2
GO TO 30

LOOP=NP1/NBOX1+1
DO 40 I=1,NP
40 NPP(I)=1
IF (NBCR.EQ.2) GO TO 60
DO 50 K=1,ISO
50 CONBOU(K,1)=BNDRY
DO 70 K=1,ISO
70 CONBOU(K,2)=ENDRY
80 DO 470 I=1,NTIM
470 INP=NP
ILINES=0
ICOUNT=0
TIME=TINT+((TIMAX-TINT)/TINC)*(I-1)
IF (NPRINT.EQ.1) GO TO 90
WRITE (NOUT,490) TIME
GO TO 100
90 WRITE (NOUT,500) TIME
100 DO 450 II=1,LOOP
450 I=I+1
IF (ICOUNT.EQ.LOOP) GO TO 240
IF (II.EQ.1) GO TO 120
IF (ICOUNT.EQ.II) GO TO 160
II=II+1
NDX=NDX+II-2
}
```
NDS = NDS + 1
IIE = IIP - 1
WRITE (NOUT, 550) IE, (NPP(J), J = NDX, NDS)
DO 110 III = 1, ISO
110 WRITE (NOUT, 560) NAMAT(III, 1), (DAPLOT(III, J, I), J = NDX, NDS)
GO TO 440
120 WRITE (NOUT, 570) IE, (NPP(J), J = NDX, NDS)
IF (NBCL.EQ.1) GO TO 140
DO 130 III = 1, ISO
130 WRITE (NOUT, 520) NAMAT(III, 1), CONBOU(III, 1), (DAPLOT(III, J, I), J = NDX, NDS)
GO TO 440
140 DO 150 III = 1, ISO
150 WRITE (NOUT, 630) NAMAT(III, 1), CONBOU(III, 1), (DAPLOT(III, J, I), J = NDX, NDS)
GO TO 440
160 NDX = NDX + II - 2
NDS = NDS + II - 2
IF (IE.EQ.0) GO TO 200
WRITE (NOUT, 570) IE, IE, (NPP(J), J = NDX, NDS)
IF (NBCR.EQ.1) GO TO 180
DO 170 III = 1, ISO
170 WRITE (NOUT, 580) NAMAT(III, 1), IE, (DAPLOT(III, J, I), J = NDX, NDS)
1, CONBOU(III, 2)
GO TO 440
180 DO 190 III = 1, ISO
190 WRITE (NOUT, 640) NAMAT(III, 1), IE, (DAPLOT(III, J, I), J = NDX, NDS)
1, CONBOU(III, 2)
GO TO 440
200 WRITE (NOUT, 610)
IF (NBCR.EQ.1) GO TO 220
DO 210 III = 1, ISO
210 WRITE (NOUT, 620) NAMAT(III, 1), CONBOU(III, 2)
GO TO 440
220 DO 230 III = 1, ISO
230 WRITE (NOUT, 650) NAMAT(III, 1), CONBOU(III, 2)
GO TO 440
240 IF (IE.EQ.0) GO TO 320
WRITE (NOUT, 590) IE, IE, (NPP(J), J = NDX, NDS)
IF (NBCL.EQ.1 .AND. NBCR.EQ.1) GO TO 300
IF (NBCR.EQ.1) GO TO 260
IF (NBCR.EQ.1) GO TO 260
DO 250 III = 1, ISO
250 WRITE (NOUT, 600) NAMAT(III, 1), CONBOU(III, 1), IE, (DAPLOT(III, J, I), J = NDX, NDS)
1, CONBOU(III, 2)
GO TO 440
260 DO 270 III = 1, ISO
270 WRITE (NOUT, 660) NAMAT(III, 1), CONBOU(III, 1), IE, (DAPLOT(III, J, I), J = NDX, NDS)
1, CONBOU(III, 2)
GO TO 440
280 DO 290 III = 1, ISO
290 WRITE (NOUT, 670) NAMAT(III, 1), CONBOU(III, 1), IE, (DAPLOT(III, J, I), J = NDX, NDS)
1, CONBOU(III, 2)
GO TO 440
300 DO 310 III = 1, ISO
310 WRITE (NOUT, 680) NAMAT(III, 1), CONBOU(III, 1), IE, (DAPLOT(III, J, I), J = NDX, NDS)
1, CONBOU(III, 2)
GO TO 440
320 WRITE (NOUT, 510) IE, (NPP(J), J = NDX, NDS)
IF (NBCL.EQ.1 .AND. NBCR.EQ.1) GO TO 410
IF (NBCR.EQ.1) GO TO 350
IF (NBCR.EQ.1) GO TO 350
DO 330 III = 1, ISO
330 WRITE (NOUT, 520) NAMAT(III, 1), CONBOU(III, 1), (DAPLOT(III, J, I), J = NDX, NDS)
1, CONBOU(III, 2)
GO TO 440
340 WRITE (NOUT, 530)
WRITE (NOUT, 610)
DO 340 III = 1, ISO
350 WRITE (NOUT, 620) NAMAT(III, 1), CONBOU(III, 2)
GO TO 440
```

360 WRITE (NOUT, 690) NANMAT(III, 1), CONBOU(III, 1), (DAPLOT(III, J, I), J
1 =NDX, NDS)
WRITE (NOUT, 530)
WRITE (NOUT, 610)
DO 370 III =1, ISO
370 WRITE (NOUT, 700) NANMAT(III, 1), CONBOU(III, 2)
GO TO 440
380 DO 390 III =1, ISO
390 WRITE (NOUT, 710) NANMAT(III, 1), CONBOU(III, 1), (DAPLOT(III, J, I), J
1 =NDX, NDS)
WRITE (NOUT, 530)
WRITE (NOUT, 610)
DO 400 III =1, ISO
400 WRITE (NOUT, 720) NANMAT(III, 1), CONBOU(III, 2)
GO TO 440
410 DO 420 III =1, ISO
420 WRITE (NOUT, 730) NANMAT(III, 1), CONBOU(III, 1), (DAPLOT(III, J, I), J
1 =NDX, NDS)
WRITE (NOUT, 530)
WRITE (NOUT, 610)
DO 430 III =1, ISO
430 WRITE (NOUT, 740) NANMAT(III, 1), CONBOU(III, 2)
440 WRITE (NOUT, 530)
IF (NPRINT.EQ.1 ) GO TO 450
ILL =ISO +3
ILINES =ILINES +ILL
IF ((ILINES +ILL).GT.55) WRITE (NOUT, 540)
450 CONTINUE
ATOMS =0.
DO 460 III =1, ISO
DO 460 J =1, NP
ATOMS =ATOMS +VOL(J)*DAPLOT(III, J, I)
CONTINUE
WRITE (NOUT, 480) ATOMS
460 CONTINUE
WRITE (NOUT, 480) ATOMS
470 CONTINUE
RETURN
C 480 FORMAT (15H NO. OF ATOMS =, 1PE22.15)
490 FORMAT (1H1, 137X, 23HCELL CONCENTRATIONS AT, 1PE12.5, 5H DAYS, //
1, 28H (CONCENTRATIONS IN ATOMS/CC))
500 FORMAT (1, 19X, 23HCELL CONCENTRATIONS AT, 1PE12.5, 5H DAYS, //
1, 28H (CONCENTRATIONS IN ATOMS/CC))
510 FORMAT (13X, 4HLEFT, /22H ISO TOPE BOUNDARY =, (3X, 5HCELL, I2, 3X))
520 FORMAT (1X, A7, 1PE13.5)
530 FORMAT (/)
540 FORMAT (1H1, 1/)
550 FORMAT (9H ISO TOPE =, (3X, 5HCELL, I2, 3X))
560 FORMAT (1X, A7, 9(1PE13.5))
570 FORMAT (9X, =, (13X), 4X, 5HRI GHT, /9H ISO TOPE =, (3X, 5HCELL, I2, 3X), 10H
1 BOUNDARY)
580 FORMAT (1X, A7, =, (1PE13.5), 1PE13.5)
590 FORMAT (13X, 4HLEFT, 4X, =, (13X), 4X, 5HRIGHT, /22H ISO TOPE BOUNDARY
1, =, (3X, 5HCELL, I2, 3X), 9H BOUNDARY)
600 FORMAT (1X, A7, 1PE13.5)
610 FORMAT (13X, 5HRIGHT, /9H ISO TOPE BOUNDARY)
620 FORMAT (1X, A7, 1PE13.5)
630 FORMAT (1X, A7, 3X, A10, 8(1PE13.5))
640 FORMAT (1X, A7, =, (1PE13.5), 3X, A10)
650 FORMAT (1X, A7, 1PE13.5)
660 FORMAT (1X, A7, 3X, A10, =, (1PE13.5), 1PE13.5)
670 FORMAT (1X, A7, 1PE13.5, =, (1PE13.5), 2X, A10)
680 FORMAT (1X, A7, 3X, A10, =, (1PE13.5), 1PE13.5)
690 FORMAT (1X, A7, 3X, A10, 8(1PE13.5))
700 FORMAT (1X, A7, 1PE13.5)
710 FORMAT (1X, A7, 1PE13.5, 8(1PE13.5))
720 FORMAT (1X, A7, 3X, A10)
730 FORMAT (1X, A7, 1PE13.5, 8(1PE13.5))
740 FORMAT (1X, A7, 3X, A10) END
SUBROUTINE TEMADJ (TEMP, TEMCOR, NTEMPS, RBAR, NCELS, W, A, B, C, NTEM)
EVALUATE TEMPERATURE FIELD FROM DATA SUPPLIED IN INPB
C
DIMENSION TEMPS(NTEM), TEMCOR(NTEM), RBAR(NCELLS), W(NTEM), A(NTEM)
1, B(NTEM), C(NTEM), IOP(2), TAB(3)

IOP(1)=5
IOP(2)=5
CALL SPL1D1 (NTEMPS,TEMCOR,TEMPS,W,IOP,IJ,A,B,C)
DO 10 J=1,NCELLS
DUM=REAR(J)
CALL SPL1D2 (NTEMPS,TEMCOR,TEMPS,W,IJ,DUM,TAB)
A(J)=TAB(1)
10 CONTINUE
DO 20 J=1,NCELLS
TEMPS(J)=A(J)
20 CONTINUE
END

SUBROUTINE MAKLAM (NANMAT,ANMAT,XSEC,NXSEC,PHI,BRV,ALAM,BB,NISO1,
1,NXS,NXSP,NBR,NGP,ISO,IXS,IGP,NCELL,K)
C MAKLAM CONSTRUCTS THE DECAY CHAIN MATRIX
DIMENSION NANMAT(NISO1,9), ANMAT(NISO1,2), XSEC(NXS,NGP,4), PHI
1(NCELL,NGP), BRV(NBR), ALAM(ISO,ISO), NXSEC(NXSP,4), BB(ISO,ISO)
DO 10 IK=1,ISO
DO 10 JK=1,ISO
ALAM(IK,JK)=0.0
10 IBR=1
DO 120 IK=1,ISO
DO 110 JK=1,ISO
DO 40 IDX=3,4
C IDENTIFY DECAY PARENTS AND STORE IN MATRIX ALAM
IF (IABS(NANMAT(IK,IDX)).NE.JK) GO TO 40
IF (NANMAT(IK,IDX)) 20,20,30
20 ALAM(IK,JK)=BRV(IBR)*ANMAT(JK,1)
IBR=IBR+1
GO TO 40
30 ALAM(IK,JK)=ANMAT(JK,1)
40 CONTINUE
DO 100 IDX=5,9
C IDENTIFY NEUTRON REACTION SOURCES
IF (IABS(NANMAT(IK,IDX)).NE.JK) GO TO 100
DO 50 J=1,IXS
IF (JK.NE.NXSEC(J,4)) GO TO 50
NM=J
GO TO 60
50 CONTINUE
PRINT 140, JK
CALL EXIT
60 CONTINUE
MM=1
IF (IDX.EQ.7) MM=2
IF (IDX.EQ.8) MM=3
IF (IDX.EQ.9) MM=4
C WEIGHT CROSS SECTIONS AND STORE IN MATRIX BB
CALL WXSEC (SIGPHI,PHI,XSEC,MM,MM,IGP,NGP,NXS,NCELL,K)
IF (NANMAT(IK,IDX)) 70,70,60
70 BB(IK,JK)=SIGPHI*BRV(IBR)
IBR=IBR+1
GO TO 90
80 BB(IK,JK)=SIGPHI
90 BB(JK,JK)=BB(JK,JK)-SIGPHI
100 CONTINUE
110 CONTINUE
ALAM(IK,IK)=-ANMAT(IK,1)
120 CONTINUE
DO 130 IK=1,ISO
DO 130 JK=1,ISO
130 ALAM(IK,JK)=-ALAM(IK,JK)-BB(IK,JK)
RETURN
C 140 FORMAT (1HO,4X,*CROSS SECTIONS CANNOT BE FOUND FOR NUCLIDE *,I4)
END
SUBROUTINE DIJADJ (DIJO, AIJS, TEMPS, DIFFK, NISO, NMATs, NCELL, ISO, K, KK)

USE ARRHENIUS RELATION TO ADJUST DIFFUSION COEFFICIENTS
DIMENSION DIJO(NISO, NISO, NMATs), AIJS(NISO, NISO, NMATs)
DIMENSION DIFFK(ISO, ISO), TEMPS(NCELL)

R = 1.987 CAL/K- MOLE

T DEGREES KELVIN

DIJO  CH**2/SEC

AIJS  CAL/ MOLE

R = 1.987

DO 20 J = 1, ISO

DO 10 JJ = 1, ISO

EXPON = AIJS(J, JJ, KK)/(R*TEMPS(K))

DIFFK(J, JJ) = DIJO(J, JJ, KK) * EXP(EXPON)

CONTINUE

END

SUBROUTINE BCONL (BK, AREA, DIFFK, DELR, SOURCE, CONBOU, NCELLs, ISO, NBCL

1, NGEOM, NISO, NIC, DUM11, DUM33)

SET LEFT BOUNDARY CONDITION

DIMENSION AREA(NCELLs), DIFFK(ISO, ISO), DELR(NCELLs), SOURCE(NIC),

1 CONBOU(NISO,2), BK(ISO, ISO), DUM11(ISO), DUM33(NIC)

IF (NBCL .EQ. 2) GO TO 30
IF (NBCL .LE. 1) GO TO 80

DO 20 I = 1, ISO

BK(I, J) = 0.0

CONTINUE

END

SUBROUTINE BCONR (AK, AREA, DIFFK, DELR, SOURCE, CONBOU, NCELLs, ISO, NBCR

1, NISO, NIC, DUM11, DUM22, BK, CKK, ALAM, VOL, DIPDM11)

SET RIGHT BOUNDARY CONDITION

DIMENSION AREA(NCELLs), DIFFK(ISO, ISO), DELR(NCELLs), SOURCE(NIC),

1 CONBOU(NISO,2), AK(ISO, ISO), DUM11(ISO), DUM22(ISO)

DIMENSION BK(ISO, ISO), CKK(ISO, ISO), ALAM(ISO, ISO), VOL(NCELLs),

1 DIPDM11(ISO, ISO)

IF (NBCHR .EQ. 2) GO TO 30
IF (NBCHR .LE. 1) GO TO 80

DO 20 I = 1, ISO

AK(I, J) = 0.0

CONTINUE

END
CALL MATMPY (ISO,ISO,AK,ISO,DUM11,NISO,DUM22,NIC)
J=NIC-ISO
DO 50 I=1,ISO
SOURCE(J+I)=SOURCE(J+I)+DUM22(I)
50 CONTINUE
60 CALL SCALAR (ALAM,VOL(NCELLS),DIFDUM1,ISO,ISO)
DO 80 I1=1,ISO
DO 70 I2=1,ISO
CKK(I1,I2)=-AK(I1,I2)-BK(I1,I2)-DIFDUM1(I1,I2)
70 CONTINUE
80 CONTINUE
RETURN
90 CONTINUE
PRINT 100
RETURN
C
FORMAT (5X,*RIGHT BOUNDARY CONDITION IMPROPERLY SPECIFIED*)
100 FORMAT
C
SUBROUTINE BIGEL (DIFFK,DIFFK1,DIFDUM1,DIFDUM2,ALAM,AK,BK,CKK,ISO
1,AREA,VOL,DELR,NCELLS,IPVT,Z)
1 EVALUATE ELEMENTS OF BIG MATRIX: A, B, AND K
2 DIMENSION DIFFK(ISO,ISO), DIFFK(ISO,ISO), DIFDUM1(ISO,ISO),
2 (ISO,ISO), IPVT(ISO), DET(2)
2 DIMENSION AREA(NCELLS), VOL(NCELLS), DELR(NCELLS)
DO 20 I1=1,ISO
DO 10 I2=1,ISO
DIFDUM1(I1,I2)=DIFFK(I1,I2)+DIFFK1(I1,I2)
10 CONTINUE
20 CONTINUE
CALL SGECO (DIFFDUM1,ISO,ISO,IPVT,RCOND,Z)
CALL SGEDI (DIFFDUM1,ISO,ISO,IPVT,DET,Z)
CALL MATMPY (ISO,ISO,ISO,DIFDUM1,ISO,DIFDUM2,ISO)
CALL MATMPY (ISO,ISO,ISO,DIFDUM1,ISO,DIFDUM2,ISO)
CONTINUE
CALL SCALAR (DIFDUM1,CON,AK,ISO,ISO)
CALL SCALAR ALAM,VOL,I,ISO,ISO)
DO 40 I1=1,ISO
DO 30 I2=1,ISO
CKK(I1,I2)=-AK(I1,I2)-BK(I1,I2)-DIFDUM1(I1,I2)
30 CONTINUE
40 CONTINUE
RETURN
C
END
SUBROUTINE MAKEB (BIGB,AK,BK,CKK,NIC,ISO,VOL,NCELLS,ICOL,II)
1 CONSTRUCT BIG MATRIX
2 DIMENSION BIGB(NIC,NIC), AK(ISO,ISO), BK(ISO,ISO), CKK(ISO,ISO),
2 VOL(NCELLS)
1 LEVEL 2, BIGB
IVOL=ICOL+II-1
DO 20 I=1,ISO
IR1=ICOL*ISO-ISO+I
IR2=ICOL*ISO+I
DO 10 J=1,ISO
IC1=ICOL*ISO-ISO+J
IC2=ICOL*ISO+J
BIGB(IR1,IC1)=CKK(I,J)/VOL(IVOL)
BIGB(IR1,IC2)=AK(I,J)/VOL(IVOL+1)
BIGB(IR2,IC1)=BK(I,J)/VOL(IVOL)
10 CONTINUE
20 CONTINUE
ICOL=ICOL+1
RETURN
ENTRY MAKEBL
IVOL=ICOL+II-1
DO 40 I=1,ISO
IR1=ICOL*ISO-ISO+I
DO 30 J=1,ISO
IC1=ICOL*ISO-ISO+J
30 CONTINUE
ICOL=ICOL+1
RETURN
ENTRY MAKEBL
IVOL=ICOL+II-1
DO 40 I=1,ISO
IR1=ICOL*ISO-ISO+I
DO 30 J=1,ISO
IC1=ICOL*ISO-ISO+J
30 CONTINUE
ICOL=ICOL+1
RETURN
ENTRY MAKEBL
IVOL=ICOL+II-1
DO 40 I=1,ISO
IR1=ICOL*ISO-ISO+I
DO 30 J=1,ISO
IC1=ICOL*ISO-ISO+J
30 CONTINUE
ICOL=ICOL+1
RETURN
$BIG(I,R1,I,C1)=CKK(I,J)/VOL(I,VOL)$

CONTINUE
RETURN
END

SUBROUTINE WXSEC (SIGPHI,PHI,XSEC,M,N,IGP,NGP,NXS,NCELL,K)

WEIGHT CROSS SECTIONS

DIMENSION PHI(NCELL,NGP), XSEC(NXS,NGP,4)

SIGPHI=0.0
DO 10 J=1,IGP

SIGPHI=SIGPHI+PHI(K,J)*XSEC(N,J,N)
RETURN
END

SUBROUTINE MAVOL (VOLMAT,NI,NC,VOL,NCELLS,II,ISO)

CONSTRUCT DIAGONAL VOLUME MATRIX

DIMENSION VOLMAT(NNIC,NIC), VOL(NCELLS)

LEVEL 2, VOLMAT
DO 20 I=1,NI
DO 10 J=1,NC
VOLMAT(I,J)=0.0
IF (I.NE.J) GO TO 10
IJ=(I-1)/ISO+II
VOLMAT(I,J)=VOL(IJ)
10 CONTINUE
20 CONTINUE
RETURN
END

SUBROUTINE SOLLFR (A,B,C,D,E,F,TINCD,I,NN)

SOLLFR EVALUATES D(A) AND I+A*D(A)

THESE VALUES ARE RETURNED IN D AND E.

THE FOLLOWING ARE REQUIRED ROUTINES

IFACT - CALCULATES FACTORIALS
SCALAR - MULTIPLIES A SCALAR TIMES A MATRIX
GENID - CREATES AN IDENTITY MATRIX
MPYEC2 - MULTIPLIES TWO LCM MATRICES - CALLS SDOT
SDOT - CALCULATES THE DOT PRODUCT OF TWO VECTORS

DIMENSION A(NN,NN), B(NN,NN), C(NN,NN), D(NN,NN), E(NN,NN), F(NN
1,NN)

LEVEL 2, A, B, C, D, E, F
EPS=1.0E-15
Y=-ALOG(EPS)
TLOG=ALOG(2.0)
DO 10 M=1,20
FACT=IFACT(M+2)
X=(M+1)*TLOG+ALOG(FACT)
IF (X.GE.Y) GO TO .20
CONTINUE
10 CONTINUE
20 CONTINUE
SUM=0.0
DO 40 JJ=1,I
SUM=SUM+A(J,JJ)*A(J,JJ)
40 CONTINUE
30 CONTINUE
40 CONTINUE

THIS USES SCHUR'S THEOREM FOR THE BOUND OF THE MAXIMUM EIGENVALUE

P=(0.5*ALOG(SUM)+ALOG(TINCD))/TLOG
IF (P) 50,50,60

NP=1
GO TO 70
60 NP=P+.1.0
70 CONTINUE
S=S+2.
80 CONTINUE

THIS LOOP IS USED IN PLACE OF 2**NP AS THAT WAS SET TO ZERO
FOR NP GREATER THAN 48 (CDC-6600)
T=TINCD/S
CALL SCAECS (A,T,C,I,NN)
CALL GENID (D,1,NN)
C CALCULATE D(H)
DO 100 J=1,M
FM=1.0/(M+2.0-J)
CALL SCAECS(D,FM,F,I,NN)
CALL MPYEC3(I,I,I,C,NN,F,NN,D,NN)
DO 90 JJ=1,1
D(JJ,JJ)=D(JJ,JJ)+1.0
90 CONTINUE
100 CONTINUE
CALL MPYEC3(I,I,I,C,NN,D,NN,E,NN)
DO 110 JJ=1,1
E(JJ,JJ)=E(JJ,JJ)+1.0
110 CONTINUE
C D AND E CONTAIN THE SCALED DOWN VALUES
CALL GENID(C,I,NN)
SI=1.0/S
DO 130 J=1,NP
CALL MOVECS(I,I,E,NN,F,NN)
CALL MPYEC3(I,I,I,E,NN,F,NN,E,NN)
CALL MOVECS(I,I,E,NN,E,NN)
DO 120 JJ=1,1
F(JJ,JJ)=F(JJ,JJ)+1.0
120 CONTINUE
CALL MPYEC3(I,I,I,C,NN,F,NN,B,NN)
CALL MOVECS(I,I,E,NN,C,NN)
130 CONTINUE
C C CONTAINS THE ENTIRE PRODUCT
CALL SCAECS(D,SI,F,I,NN)
CALL MPYEC3(I,I,F,NN,C,NN,D,NN)
C I + A * D(A) IS IN E
C TEST E MATRIX FOR ALL ZEROS
SUM=0.
DO 150 JJ=1,1
DO 140 J=1,1
SUM=SUM+E(J,JJ)
140 CONTINUE
150 CONTINUE
IF (SUM.NE.0.0) RETURN
PRINT 160
STOP
160 FORMAT (32H ALARM SOUNDED, E MATRIX ZERO. )
END
SUBROUTINE PSOLVE (VOLMAT,C,D,E,DUM33,DTIME,NIC,CONINT,SOURCE
1,CONCEN)
C CONSTRUCT SOLUTION FROM RESULTS OF SOLVER
DIMENSION VOLMAT(NIC,NIC), C(NIC,NIC), D(NIC,NIC), E(NIC,NIC),
1 DUM33(NIC), CONINT(NIC), SOURCE(NIC), CONCEN(NIC)
LEVEL 2, VOLMAT, C, D, E
CALL PRIMES (VOLMAT,NIC,NIC,NIC,NIC,10H VOLMAT, 10H )
CALL PRIV(CONINT,NIC,NIC,10H CONINT, 10H )
CALL MPYEC1(NIC,NIC,1,VOLMAT,NIC,CONINT,NIC,CONCEN,NIC)
CALL PRIV(CONCEN,NIC,NIC,10H VOLMAT*CO,10HNINT)
CALL MPYEC1(NIC,NIC,1,E,NIC,CONCEN,NIC,CONINT,NIC)
CALL PRIV(CONCEN,NIC,NIC,10H VOLMAT*CO,10HNINT*E)
DO 10 I=1,NIC
VOLMAT(I,I)=1./VOLMAT(I,I)
10 CONTINUE
CALL MPYEC1(NIC,NIC,1,VOLMAT,NIC,CONINT,NIC,CONCEN,NIC)
CALL PRIV(CONCEN,NIC,NIC,10H FIRST PAR,10H )
CALL MPYEC1(NIC,NIC,1,D,NIC,SOURCE,NIC,CONINT,NIC)
CALL PRIV(CONINT,NIC,NIC,10H D*SOURCE, 10H )
CALL SCAECS (VOLMAT,DTIME,C,NIC,NIC)
CALL PRIMES(C,NIC,NIC,NIC,NIC,10H DTIME*VOL,10HMAT)
CALL MPYEC1(NIC,NIC,1,C,NIC,CONINT,NIC,DUM33,NIC)
CALL PRIV(DUM33,NIC,NIC,10H SECOND PA,10HNT)
DO 20 I=1,NIC
VOLMAT(I,I)=1./VOLMAT(I,I)
CONCEN(I)=CONCEN(I)+DUM33(I)
20 CONTINUE
CALL PRIV(CONCEN,NIC,NIC,10H ANSWER, 10H )
RETURN
END
SUBROUTINE SCALAR (A, S, B, N, Ndim)
C SCALAR MULTIPLIES A SCALAR TIMES A MATRIX
DIMENSION A(NDIM,NDIM), B(NDIM,NDIM)
DO 10 J=1,N
DO 10 I=1,N
B(I,J)=S*A(I,J)
10 CONTINUE
RETURN
END

SUBROUTINE SCAEC (A, S, B, N, Ndim)
C SCAEC MULTIPLIES A SCALAR TIMES A MATRIX
LEVEL 2, A, B
DIMENSION A(NDIM,NDIM), B(NDIM,NDIM)
DO 10 J=1,N
DO 10 I=1,N
B(I,J)=S*A(I,J)
10 CONTINUE
RETURN
END

FUNCTION IFACT (N)
C EVALUATE N FACTORIAL
IFACT=1
IF (N.LE.1) RETURN
DO 10 I=1,N
IFACT=IFACT*I
10 CONTINUE
RETURN
END

SUBROUTINE GENID (A, N, IA)
C GENERATE IDENTITY MATRIX
DIMENSION A(IA,N)
LEVEL 2, A
DO 20 J=1,N
DO 20 I=1,N
20 A(I,J)=0.0
RETURN
END

SUBROUTINE MATMOV (N, M, A, IA, B, IB)
C EQUIVALENCE TWO MATRICES
DIMENSION A(IA,M), B(IB,M)
DO 10 J=1,M
DO 10 I=1,N
10 B(I,J)=A(I,J)
RETURN
END

SUBROUTINE MOVECS (N, M, A, IA, B, IB)
C EQUIVALENCE TWO LCM MATRICES
DIMENSION A(IA,M), B(IB,M)
LEVEL 2, A, B
DO 10 J=1,M
DO 10 I=1,N
10 B(I,J)=A(I,J)
RETURN
END

SUBROUTINE MATMPY (N, M, L, A, IA, B, IB, C, IC)
C MULTIPLY TWO MATRICES
DIMENSION A(IA,M), B(IB,L), C(IC,L)
DO 10 J=1,L
DO 10 I=1,N
10 C(I,J)=SUMT(M,A(I,1),IA,B(1,J),1)
RETURN
END

SUBROUTINE NPYEC3 (N, M, L, A, IA, B, IB, C, IC)
C MULTIPLY TWO LCM MATRICES
DIMENSION A(IA,M), B(IB,L), C(IC,L)
LEVEL 2, A, B, C
DO 20 J=1,L
DO 20 I=1,N
AM=0.0
DO 20 K=1,M
C SUBROUTINE MPYEC1 (N,M,L,A,IA,E,IB,C,IC)
C MULTIPLY A SCM AND LCM MATRIX
C DIMENSION A(IA,M), B(IB,L), C(IC,L)
C LEVEL 2, A
C DO 20 K=1,M
C AM=AM+A(I,K)*B(K,J)
C END
C SUBROUTINE PRIM (A,N1,N2,N1D,N2D,TITLE1,TITLE2)
C PRINT A MATRIX
C DIMENSION A(NID,N2D)
C PRINT 20, TITLE1,TITLE2
C DO 10 J=1,N1
C PRINT 30, (A(J,JJ),JJ=1,N2)
C RETURN
C 20 FORMAT (/1X,2A10/) 30 FORMAT (6E13.5)
C END
C SUBROUTINE PRIMES (A,N1,N2,N1D,N2D,TITLE1,TITLE2)
C PRINT A LCM MATRIX
C DIMENSION A(N1D,N2D)
C PRINT 20, TITLE1,TITLE2
C DO 10 J=1,N1
C PRINT 30, (A(J,JJ),JJ=1,N2)
C RETURN
C 20 FORMAT (/1X,2A10/) 30 FORMAT (6E13.5)
C END
C SUBROUTINE PRIV (A,N1,N1D,TITLE1,TITLE2)
C PRINT A VECTOR
C DIMENSION A(NID)
C PRINT 20, (A(I),I=1,N1)
C RETURN
C 10 FORMAT (/1X,2A10,/)
C 20 FORMAT (6E13.5)
C END
C SUBROUTINE REAG (ARRAY,NCOUNT,I1O1,H1O2)
C READS FLOATING POINT DATA
C DIMENSION ARRAY(NCOUNT), V(12), K(12), IN(12)
C COMMON /10/ NINP, NOUT, IER, NPRINT
C JFLAG=0
C J=1
C IF (JFLAG.EQ.0) GO TO 30
C 10 DO 20 JJ=1,6
C K(JJ)=K(JJ+6)
C IN(JJ)=IN(JJ+6)
C 20 V(JJ)=V(JJ+6)
C JFLAG=0
C GO TO 40
C 30 READ (NINP,200) (K(I),IN(I),V(I),I=1,6)
C 40 DO 160 I=1,6
C L=K(I)+1
C GO TO (50,60,80,170,120), L
C DO MODIFICATION
C 50 ARRAY(J)=V(I)
C J=J+1
C GO TO 160
C
60   REPEAT
   L=IN(I)
   DO 70 M=1,L
   ARRAY(J)=V(I)
   J=J+1
   GO TO 160
C
70   INTERPOLATE
80   IF (I-6).GT.100,90,90
90   READ (NINP,200) (K(JJ),IN(JJ),V(JJ),JJ=7,12)
   JFLAG=1
100  L=IN(I)+1
   DEL=(V(I+1)-V(I))/FLOAT(L)
   DO 110 M=1,L
   ARRAY(J)=V(I)+DEL*FLOAT(M-1)
   J=J+1
   GO TO 160
C
110  INTERPOLATE WITH CONSTANT RATIO
120  IF (I.LT.6) GO TO 130
130  L=MAXO(2,IN(I)+1)
   T1=0.
   T2=1.
   DO 140 JJ=1,L
   T1=T1+T2
   T2=T2*V(I)
140  T2=(V(I+1)-ARRAY(J-1))/T1
   L=MAXO(IN(I),IN(I)+1)
   DO 150 JJ=1,L
   ARRAY(J)=ARRAY(J-1)+T2
   T2=T2*V(I)
   J=J+1
150  CONTINUE
160  CONTINUE
   GO TO 10
C
170  TERMINATE
180  WRITE (NOUT,210) HOL1,HOL2,J,(ARRAY(I),I=1,NCOUNT)
   IER=1
190  RETURN
C
200  FORMAT (6(I1,I2,E9.3))
210  FORMAT (/1X,2A6,E16.16/6E13.5))
220  FORMAT (/33H INCORRECT NUMBER OF INPUT ITEMS ,2A6)
END
SUBROUTINE REAL (IARRAY,NCOUNT,IOL1,HOL2)
C
READS INTEGER DATA
DIMENSION IARRAY(NCOUNT), IV(6), K(6), IN(6)
COMMON /10/ NINP, NOUT, IER, NPRINT
J=1
10   READ (NINP,100) (K(I),IN(I),IV(I),I=1,6)
   DO 50 I=1,6
   L=K(I)+1
   GO TO (20,30,50,70), L
C
50   NO MODIFICATION
20   IARRAY(J)=IV(I)
   J=J+1
   GO TO 60
C
30   REPEAT
   L=IN(I)
   DO 40 M=1,L
   IARRAY(J)=IV(I)
   J=J+1
   GO TO 60
C
INTERPOLATE
50   WRITE (NOUT,120) HOL1,HOL2
   IER=1
   RETURN
60 CONTINUE
go to 10

terminate

c 70 j=j-1
write (nout,110) hol1,hol2,j,(iarray(i),i=1,ncount)
if (j(ncount)) 80,90,80
80 write (nout,130) hol1,hol2
ier=1
90 return

c 100 format (6(i1,i2,i9))
110 format (/x,2a6,i6/(6i12))
120 format (44hratting to interpolate between integers ,2a6)
130 format (33hoincorrect number of input items ,2a6)

end

draw plots of results

dimension nanmat(niso1,2), rbar(ncells), concen(np), dum33(np),
1 daplot(niso,ncell,ntime), dist(ncell1), conbou(dum33(np),
1 dimension x(2), y(2), labelx(3), labely(4)

call gplot (1h0,10dash plots,10)
call bgmpl (0)
call height (0.25)
y(1)=0.0
x(1)=dist(1)
x(2)=dist(ncells+1)
xscale=aint(alog10(x(2)))
xscldiv=10.0**xscale
if (np.eq.ncells) go to 40
if (xscale.ge.2.) go to 20
10 dum33(i)=rbar(i+1)
x=13
encode (nx,160,labelx)
go to 80
20 x(2)=x(2)/xscldiv
ixs=xscale
nx=23
encode (nx,170,labelx) ixs
d0 30 i=1,np
30 dum33(i)=rbar(i+1)/xscldiv
go to 80
40 if (xscale.ge.2.) go to 60
do 50 i=1,np
50 dum33(i)=rbar(i)
x=13
encode (nx,160,labelx)
go to 80
60 do 70 i=1,np
70 dum33(i)=rbar(i)/xscldiv
x(2)=x(2)/xscldiv
ixs=xscale
nx=23
encode (nx,170,labelx) ixs
do 80 i=1,iso
80 z=0.0
do 100 nt=1,ntim
d0 90 j=1,np
z=amax1(z,daplot(i,j,nt))
90 continue
100 continue
y(2)=amax1(conbou(i,1),conbou(i,2),z)
if (ncbl.eq.1) y(2)=amax1(conbou(i,2),z)
if (nbcrl.eq.1) y(2)=amax1(conbou(i,1),z)
if (nbcrl.eq.1 .and. ncl.eq.1) y(2)=z
yscale=aint(alog10(y(2)))
yscldiv=1.0
if (yscale.lt.2.) go to 110
yscldiv=10.0**yscale
y(2)=y(2)/yscldiv
ys=yscale
ny=33
ENCODE (NY, 180, LABELY) IYS
GO TO 120
110 NY = 23
ENCODE (NY, 190, LABELY)
120 CALL TITLE (NAMMAT(1, 1), 7, LABELX, NAX, LABELX, NY, 5, 5, 8.)
CALL GRAP (X(1), 10HSCALE , X(2), Y(1), 10HSCALE , Y(2))
ND 140 NT = 1, NTIM
DO 130 J = 1, NP
CONC(J) = DAPLOT(I, J, NT)
130 CONC(J) = CONC(J) / YSCLDIV
CALL CURVE (DUM33, CONC, NP, 0)
CONTINUE
CALL ENDPL (1)
CONTINUE
CALL DONEPL
RETURN

C
160 FORMAT (13HDISTANCE (CM))
170 FORMAT (21HDISTANCE (CM) X 10**-, I2)
180 FORMAT (31HCONEENTRATION (ATOM/CC) X 10**-, I2)
190 FORMAT (23HCONEENTRATION (ATOM/CC))
END
SUBROUTINE SPL1D1 (N, X, F, W, IOP, IJ, A, B, C)
WHERE N = NUMBER OF POINTS IN THE INTERPOLATION
X = ORIGIN OF TABLE OF INDEPENDENT VARIABLE
F = ORIGIN OF TABLE OF DEPENDENT VARIABLE
W = AN ARRAY OF DIMENSION N WHICH CONTAINS THE CALCULATED
SECOND DERIVATIVES UPON RETURN
IOP = AN ARRAY OF DIMENSION 2 WHICH CONTAINS COMBINATIONS OF
THE INTEGERS 1 THRU 5 USED TO SPECIFY THE BOUNDARY
CONDITIONS
IJ = SPACING IN THE F AND W TABLES
A, B, C = ARRAYS OF DIMENSION N USED FOR TEMPORARY STORAGE

DIMENSION IOP(2), X(4), F(2), W(2), A(2), B(2), C(2), COMM(6)
DATA (COMM(J), J = 1, 6) / 6HSP1D1 N, 6H LESS THAN 4. RE, 6HSULTS IN,
1 8HC0RRECT., 8H
K = N - 1
A(2) = (X(2) - X(1)) / 6.
B(2) = (X(3) - X(1)) / 3.
W(IJ + 1) = (F(2*IJ + 1) - F(IJ + 1)) / (X(3) - X(2)) - (F(IJ + 1) - F(1)) / (X(2) - X(1))
IF (N < 3) 10, 30, 10
10 DO 20 I = 3, K
M = I - 1
J = I + 1
CON = (X(IJ + 1) - X(I - 1)) / 3.
DO = (X(I) - X(I - 1)) / 6.
B(I) = CON - (DON**2) / B(I - 1)
E = (F(J) - F(M)) / (X(I) - X(I - 1)) - (F(M) - F(J - 2)) / (X(I) - X(I - 1))
W(I - 1) = E / (DON**2) / B(I - 1)
20 A(I - 1) = DON**2 / B(I - 1)
30 K = N - 2
C(N - 1) = (X(N) - X(N - 1)) / 6. / B(N - 1)
W(K - 1) = W(K - 1) / B(N - 1)
A(N - 1) = A(N - 1) / B(N - 1)
K2 = K - 1
IF (N < 3) 40, 60, 40
40 DO 50 I = 2, K2
J = N - 1
CON = (X(J + 1) - X(J)) / 6.
A(J) = (A(J) - CON**2) / B(J)
C(J) = (CON**2) / B(J)
K3 = J + 1
M = K3 + 1
50 W(K3) = W(K3) - DON**2 / B(J)
60 A(N) = A(N) / B(J)
IF (IOP(2) - 5) 70, 90, 70
70 C1 = W(1)
IF (IOP(1) - 5) 70, 90, 70
80 C2 = W(K4)
GO TO 130

65
90 IF (N-4) 570, 100, 100
100 A1=X(1)-X(2)
    A2=X(1)-X(3)
    A3=X(1)-X(4)
    A4=X(2)-X(3)
    A5=X(2)-X(4)
    A6=X(3)-X(4)
    W(1)*I+1/A2*4*A4+A4*F(I+1)/(I+1)
    GO TO 70
110 IF (N-4) 570, 120, 120
120 B1=X(N)-X(N-3)
    B2=X(N)-X(N-2)
    B3=X(N)-X(N-1)
    B4=X(N-1)-X(N-3)
    B5=X(N-1)-X(N-2)
    B6=X(N-2)-X(N-3)
    L1=K4-IJ
    L2=K4-IJ
    L3=K4-IJ
    1*(B4*B5*B3)*F(K4)*(1./B1+1./B2+1./B3)
    GO TO 80
130 DO 160 I=1,K
    M=(I-1)*I+1
    GO TO 70
140 IF (I-1) 150, 160, 150
150 W(1)=W(1)-C0*B0*K(N)
    W(K4)=W(K4)-K(N)*W(M)
    A(N)=A(N)-K(N)*A(I)
    C(N)=C(N)-K(N)*C(I)
    CONTINUE
    GO TO 550
160 MK=IOP(1)
    GO TO (180, 210, 260, 310, 260), MK
170 IF (I-1) 200, 190, 200
180 A(I)=1.
    C(I)=0.
    GO TO 340
190 BOB=0.
    GO TO 340
200 IF (I-1) 230, 220, 230
210 A(I)=1.
    C(I)=0.
    W(1)=0.
    GO TO 340
220 A(I)=1.
    C(I)=0.
    GO TO 340
230 IF (I-2) 240, 240, 250
240 BOB=0.
    GO TO 340
250 BOB=0.
    GO TO 340
260 IF (I-1) 280, 270, 280
270 A(I)=-X(2)-X(3))/3.
    C(I)=0.
    W(1)=C1*(F(I+1)-F(1))/(X(2)-X(1))
    GO TO 340
280 IF (I-2) 300, 290, 300
290 BOB=(X(2)-X(3))/6.
    GO TO 340
300 BOB=0.
    GO TO 340
310 IF (I-1) 330, 320, 330
320 A(I)=1.
    C(I)=1.
    W(1)=0.
    GO TO 340
330 BOB=0.
340 ML=IOP(2)
    GO TO (350, 380, 430, 480, 430), ML
SUBROUTINE SPL1D2 (N,X,F,W,IJ,Y,TAB)
WHERE N = NUMBER OF POINTS IN THE INTERPOLATION
X = ORIGIN OF TABLE OF THE INDEPENDENT VARIABLE
F = ORIGIN OF TABLE OF THE DEPENDENT VARIABLE
W = ORIGIN OF TABLE OF SECOND DERIVATIVES AS CALCULATED BY
   SPL1D1
IJ = SPACING IN THE TABLES F AND W
Y = THE POINT AT WHICH INTERPOLATION IS DESIRED
TAB = AN ARRAY OF DIMENSION 3 WHICH CONTAINS THE FUNCTION
      VALUE, FIRST DERIVATIVE, AND SECOND DERIVATIVE AT Y
DIMENSION X(3), F(3), W(3), TAB(3)

LOCATE Y IN THE X TABLE

IF (Y-X(1)) 10,10,20
10 I=1
GO TO 50
20 IF (Y-X(N)) 40,30,30
30 I=N-1
GO TO 50
40 CALL SEARCH (Y,X,N,NDX,MFLAG)
50 MI=(I-1)*I+1
   K1=MI+I
   FLK=X(I+1)-X(I)
C
   CALCULATE F(Y)
C
   A=(W(MI)^*(X(I+1)-Y)^2+W(K1)^*(Y-X(I))^2)/(6.*FLK)
   B=(F(K1)-F(MI))/FLK
   C=FLK^*(W(MI)-W(K1))/6.
   TAB(1)=A+B+C
C
   CALCULATE THE FIRST DERIVATIVE AT Y
C
   A=(W(K1)^*(Y-X(I))^2-W(MI)^*(X(I+1)-Y)^2)/(2.*FLK)
   B=(F(K1)-F(MI))/FLK
   C=FLK^*(W(MI)-W(K1))/6.
   TAB(2)=A+B+C
C
   CALCULATE THE SECOND DERIVATIVE AT Y
C
   TAB(3)=(W(MI)^*(X(I+1)-Y)^2+W(K1)^*(Y-X(I)))^2)/FLK
RETURN
END

IDENT SEARCH
ENTRY SEARCH

CALL SEARCH(X,XT,N,NDX,MFLAG)

BINARY SEARCH WITH MEMORY OF ARRAY XT (LENGTH N) FOR
VALUE X. RESULT IS RETURNED IN NDX, AND A FLAG IS SET SO THAT
MFLAG = 0 IF X=XT(NDX)
MFLAG = 1 IF XT(NDX) LT X LT XT(NDX+1)
MFLAG = 2 IF X LT XT(NDX) WHERE NDX=1
OR X GT XT(NDX) WHERE NDX=N
XT MAY BE FIXED POINT, FLOATING POINT OR CHARACTER VALUES,
AND MUST BE EITHER MONOTONIC INCREASING OR DECREASING.

IF FLOATING POINT VALUES USED, THEN (BASE 10) EXPONENTS
ARE RESTRICTED TO LESS THAN 150 IN ABSOLUTE VALUE
IF CHARACTERS ARE USED, OCTAL DISPLAY CODE SHOULD BE LESS
THAN 40B (THAT IS MAY BE ALL ALPHABETICS AND NUMBERS 0-4)

ILO IS BEGINNING SEARCH VALUE
SET TO 1 UPON FIRST ENTRY
SET TO LAST NDX UPON EACH SUCCEEDING ENTRY.
ILO DATA 1

THIS IS RETURN BRANCH IF XT(ILO) LT X LT XT(ILO+1)
 DONE11  S6X  B7
       EQ  DONE12
 DONE1  S6X  B6
       BX  B1
       NG  B5,DONE11
 DONE12  SA6  A5
       SA7  A0
       SA6  B4
       IFI EQ *F,2
       CALED BY FTN
       SEARCH  SA1  TEMP
       SAO  X1
       ENDT
IN ORDER TO DO A BINARY SEARCH, X MUST BE IN AN INTERVAL

SINCE WE SAVED VALUE OF ILO, WE MUST FORCE INTERVAL WHERE

XT(ILO) LT X LT XT(IHI), BY MOVING ILO AND IHI UP OR DOWN XT.

**DATA**

1. IF EQ X1, XT(ILO) LT X LT XT(IHI) RETURN

2. IF N E5, IHI IS GT N

3. IF X4, SWIT YES, XT(N) LT XT(IHI) Switch

4. IF X IS MONOTONIC DECREASING BRANCH TO SWITCH VALUES

5. CHECK STARTING POINTS TO SEE IF X IS IN AN INTERVAL OF XT(ILO)

6. AND XT(IHI). MUST BE TO BEGIN ACTUAL BINARY SECTION.

7. IF XT IS MONOTONIC DECREASING BRANCH TO SWITCH VALUES

**ENDIF**

**ELSE**

1. X3 CONTAINS X

2. X4 CONTAINS N

3. STORE ADDRESS OF MFLAG

4. KEEP VALUE OF I

5. PICK UP BEGINNING VALUE OF ILO

6. B2 NOW REFERENCES XT FOR INDEXES

7. ISTEP = 1

**ENDIF**

**E6, B6 - 135 DECREASE ILO TO GET THIS STEP**

**LT**

1. ILO IS ZERO OR LESS NOT VALID

2. IHI = ILO

3. ISTEP = 2*ISTEP

**DECR**

1. NEXT DECREASING ILO

2. X7, DONE07 X GT XT(IHI) MUST BRING INTERVAL UP

3. X = XT(ILO)

4. XT(ILO) LT X LT XT(ILO + 1) SO DONE

**SMLXW**

1. DECREASE ILO TO GET THIS STEP

2. ILO IS ZERO OR LESS NOT VALID

3. GET NEXT VALUE

4. IHI = ILO

5. ISTEP = 2*ISTEP

6. NEXT DECREASING ILO
XT WAS MONOTONIC DECREASING

SWIT
SA1 A2
SB7 B6
SB6 B6+B1
SB5 -B5
SA2 A2+B1
LE B6,B3,DEC
SB6 B1

ILO2BIG
PL X4,BSRCH
SA1 B2+B1
SB7 B6
SA2 B2+B3
SB6 B3

IX6 X1-X3
IX4 X3-X2
IFEQ *F,2
NG X6,SEARCH1
ELSE
NG X6,SEARCH
X NOT IN TABLE ALL DONE
ENDIF
PL X4,BSRCH
SA7 B4
SA7 A5
IFEQ *F,2
EQ SEARCH1
ELSE
EQ SEARCH
ALL DONE
ENDIF

ILO2SML
IX4 X3-X2
SA2 B2+B1
SA7 B4
IFEQ *F,2
NG X4,SEARCH1
ELSE
NG X4,SEARCH
X IS LT XT(I1) ALL DONE
EQ BSRCH
INTERVAL SET TRY BINARY

IHI2BIG
SA1 B2+B3
SB6 B3-B1
SA2 A1-B1

NEXT INCREASING WILL BE OUT
N = NEXT STEP(DECREASING)
X-XT(ILO)
INTERVAL NOT FOUND MUST REPEAT
INTERVAL FORMED PERFORM BINARY.
SA7      B1       1
SA6      A5       STORE ILO = 1
SA7      A0       STORE MFLAG = 2
SA7      B4       STORE NDX = 1

* IX4      X1-X3   X-XT(N)
SB5      B3       PLACE ANSWER IN B5
IX6      X3-X2   X-XT(N-1)
IFEQ *F,2    CALLED BY FTN
LE       B3,B1,SEARCH1
ELSE
LE       B3,B1,SEARCH    XT NOT AN ARRAY    ALL DONE
ENDIF
ZR       X4,DONE00  HAVE ANSWER BRANCH TO RETURN VALUES
SB5      B1       RESTORE ISTEP
NG       X4,HIHTN  OPPS OUT RETURN RIGHT VALUES
ZR       X6,DONE0  FOUND X EXACT
PL       X6,DONE1  X BETWEEN XT(N-1) AND XT(N)
EQ       SMLXW    STILL MUST FORM INTERVAL

* X WAS GREATER THAN BEGINNING HI. NEED TO MOVE INTERVAL DOWN
* XT TO DO BINARY SEARCH. MOVE INTERVAL, DOUBLING STEP SIZE
* AT EACH STEP UNTIL XT(ILO) LT X LT XT(IHI)

BIGXW  SA1   A1+B5   X1 = XT(IHI)
SA6   B7    ILO = IHI
SB7   B7+B5   IHI = IHI+ISTEP
SB5   B5+B5   ISTEP = I*ISTEP
NO
SX7   B7+B5   NEXT INCREASING VALUE
SX6   B3    KEEP VALUE OF N
ZR       X7,HIHTL1 NEXT DECREASING VALUE IS OUTP
NG       X7,HIHTL1 NEXT DECREASING VALUE IS OUT
IX7       X7-X7   N = NEXT STEP(INCREASING)
IX4       X7-X3   X-XT(IHI)
NG       X7,HIHTL1 NEXT STEP OUT OF INTERVAL
NG       X4,BIGXW    STILL DO NOT HAVE INTERVAL REPEAT
ZR       X4,DONE07

* HAVE INTERVAL MUST SET UP A MID POINT FOR BINARY

BSRCH  ZR       X4,DONE0   FOUND ANSWER EXACTLY
SX6   B6+B7   IHI + ILO
SB5   B7    MID = IHI
AX6   1     NEXT MID POINT MAYBE
SA2   B2+X6   X2 = XT(MID)

* FINALLY   THIS IS ACTUAL BINARY SEARCH SECTION

BSRCH1 SB7   B5   IHI = MID
SB5   X6    MID POINT
SX0   B6+X6   X0 = ILO+MID
SX1   B7+X6   X1 = IHI+MID
AX6   B1,X0   MID 1 IF X BETWEEN XT(ILO) AND XT(MID)
AX1   1      MID 2 IF X BETWEEN XT(IHI) AND XT(MID)
IX0   X2-X3   XT(MID)-X
IX5   X1-X6   IHI - ILO
SA2   B2+X6   X2 = XT(MID 1)
ZR       X5,DONE  MID MATCHES BOUNDARY RETURN ANSWERS
SA4   B2+X1   X4 = XT(MID 2)

* LOOP
ZR       X0,DONE00  X MATCHES XT(MID) RETURN VALUE
PL       X0,BSRCH1  X BETWEEN XT(MID) AND XT(IHI)

* SB6   B5   ILO = MID
SB5   X1    NEW MID POINT
SX6   X1+B6   X6 = MID + ILO
SX1   X1+B7   X1 = MID + IHI
AX6   1      MID 1 X BETWEEN XT(MID) AND XT(ILO)
AX1   1      MID 2 X BETWEEN XT(MID) AND XT(IHI)
IX0 X4-X3                    XT(MID)-X
IX5 X1-X6                    IHI-IL0
SA2 B2+X6                    X2 = XT(MID 1)
SA4 B2+X1                    X4 = XT(MID 2)
NZ X5, LOOP                  HAVE NOT FOUND VALUE TRY AGAIN

* * RIGHT ANSWER FOUND IN INTERVAL XT(X6) LT X LT XT(X6+1)
* * DONE ZR X0,DONE0         FOUND RIGHT ANSWER EXACTLY
SX7 B1                      1
SA6 B4                      STORE ANSWER IN NDX
SA7 A0                      STORE 1 IN MFLAG
SA6 A5                      STORE ANSWER IN IL0
IFEQ *F, 2                  CALLED BY FTN
ELSE EQ SEARCH1            ALL DONE
ENDIF

* * RIGHT ANSWER FOUND EXACTLY
* * DONE0 SX6 B6                 ANSWER IN B6
MX7 0                       0
SA6 B4                      STORE ANSWER IN NDX
SA7 A0                      STORE 0 IN MFLAG
SA6 A5                      STORE ANSWER IN IL0
IFEQ *F, 2                  CALLED BY FTN
ELSE EQ SEARCH1            ALL DONE
ENDIF

DONE00 SX6 B5                 ANSWER IN B5
MX7 0                       SET TO 0
SA6 B4                      NDX = ANSWER
SA7 A0                      MFLAG = 0
SA6 A5                      IL0 = ANSWER
IFEQ *F, 2                  CALLED BY FTN
ELSE EQ SEARCH1            ALL DONE
ENDIF

* * DONE07 SX6 B7              CALLED BY FTN
MX7 0                       
SA6 B4                      
SA7 A0                      
SA6 A5                      
IFEQ *F, 2
ELSE EQ SEARCH             ALL DONE
ENDIF

IHIGHTN SA1 B2+B3             X1 = XT(N)
ZR X4,DONE07                FOUND INTERVAL
PL X4,ESRCH                 
SB6 B3                       

* SX6 B3                  N
SX7 B1+B1                  2
SA6 A5                  STORE IL0 = N
SA7 A0                  STORE MFLAG = 2
IX4 X1-X3                  XT(N)-X
SA6 B4                  STORE NDX = N

* IFEQ *F, 2                 CALLED BY FTN
NG X4,SEARCH1           X OUT OF RANGE ALL DONE
ELSE NG X4,SEARCH          MUST KEEP POINTERS RIGHT
ENDIF
LT B5,B0,ESRCH
**IHILT1**

**X4, DONEO**

FOUND RIGHT ANSWER

**X4 = XT(1)**

OR WONT ZERO IN ON RIGHT SECTION

**X4, BSRCB**

FOUND INTERVAL USE BINARY

**ILO = 1**

**SA7**

STORE ILO = 1

**SA6**

STORE MFLAG = 2

**B4, BSRCB**

STORE NDX = 1

CALLED BY FTN

**SA7**

MONTONIC DECREASING INTERVAL

**X4, SEARCH**

FOUND RIGHT ANSWER

MONTONIC DECREASING INTERVAL

CALLED BY FTN

**TEMP**

THE FOLLOWING SUBROUTINE IS A REPLACEMENT FOR THE

ROUTINE SEARCH FOR USE WITH MACHINES THAT DO

NOT COMPIL COMPASS.

SUBROUTINE SEARCH(X,XT,N,NDX,MFLAG)

SEARCH FINDS A GIVEN VALUE X IN A MONOTONIC SERIES

DIMENSION XT(1)

10 PRINT XT(N) - XT(1)) 140, 10, 20

CALL EXIT

20 IF (XT(N) - X) 40, 30, 50

30 NDX = N

MFLAG = 0

RETURN

40 NDX = N

MFLAG = 2

RETURN

50 IF (XT(1) - X) 80, 60, 70

60 NDX = 1

MFLAG = 0

RETURN

70 NDX = 1

MFLAG = 2

RETURN

80 NDX = N/2.0 + 0.5

MO = 1

M2 = N

90 IF (XT(NDX) - X) 110, 100, 130

100 MFLAG = 0

RETURN

110 MO = NDX

C IF (MO = NDX) 90, 120, 90

120 MFLAG = 1

RETURN

130 M2 = NDX

C IF (MO = NDX) 90, 120, 90

140 IF (X - XT(1)) 170, 150, 160

END
SUBROUTINE SGECO(A,LDA,N,IPVT,RCOND,Z)

INTEGER LDA,N,IPVT(1)
REAL A(LDA,N),Z(I)
REAL RCOND

SGECO FACTORS A REAL MATRIX BY GAUSSIAN ELIMINATION
AND ESTIMATES THE CONDITION OF THE MATRIX.

IF RCOND IS NOT NEEDED, SGECFA IS SLIGHTLY FASTER.
TO SOLVE A*X = B, FOLLOW SGECO BY SGESL.
TO COMPUTE INVERSE(A)*C, FOLLOW SGECO BY SGESL.
TO COMPUTE DETERMINANT(A), FOLLOW SGECO BY SGEDI.
TO COMPUTE INVERSE(A), FOLLOW SGECO BY SGEDI.

ON ENTRY

A REAL(LDA, N)
THE MATRIX TO BE FACTORED.

LDA INTEGER
THE LEADING DIMENSION OF THE ARRAY A .

N INTEGER
THE ORDER OF THE MATRIX A .

ON RETURN

A AN UPPER TRIANGULAR MATRIX AND THE MULTIPLIERS
WHICH WERE USED TO OBTAIN IT.
THE FACTORIZATION CAN BE WRITTEN A = L*U WHERE
L IS A PRODUCT OF PERMUTATION AND UNIT LOWER
TRIANGULAR MATRICES AND U IS UPPER TRIANGULAR.

IPVT INTEGER(N)
AN INTEGER VECTOR OF PIVOT INDICES.

RCOND REAL
AN ESTIMATE OF THE RECIPROCAL CONDITION OF A .
FOR THE SYSTEM A*X = B, RELATIVE PERTURBATIONS
IN A AND B OF SIZE EPSILON MAY CAUSE
RELATIVE PERTURBATIONS IN X OF SIZE EPSILON/RCOND .
IF RCOND IS SO SMALL THAT THE LOGICAL EXPRESSION 
1.0 + RCOND .EQ. 1.0 
IS TRUE, THEN A MAY BE SINGULAR TO WORKING 
PRECISION. IN PARTICULAR, RCOND IS ZERO IF 
EXACT SINGULARITY IS DETECTED OR THE ESTIMATE 
UNDERFLOWS.

Z
REAL(N)
A WORK VECTOR WHOSE CONTENTS ARE USUALLY UNIMPORTANT.
IF A IS CLOSE TO A SINGULAR MATRIX, THEN Z IS 
AN APPROXIMATE NULL VECTOR IN THE SENSE THAT 
\|A\|Z\| = RCOND\|A\|\|Z\| .

LINPACK. THIS VERSION DATED 07/14/77.
CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.

SUBROUTINES AND FUNCTIONS
LINPACK SGEE'A
BLAS SAXPY,SDOT,SSCAL,SASUM
FORTRAN ABS,AMAX1,SIGN

INTERNAL VARIABLES
REAL SDOT,EK,T,WK,WKM
REAL ANORM,S,SASUM,SM,YNORM
INTEGER INFO,J,K,KB,KP1,L

REAL SIGN

COMPUTE 1-NORM OF A
ANORM = 0.0EO
DO 10 J = 1, N 
ANORM = AMAX1(ANORM,SASUM(N,A(1,J),1))
10 CONTINUE

FACTOR
CALL SGEFA(A,LDA,N,IPVT,INFO)

RCOND = 1/(NORM(A)*(ESTIMATE OF NORM(INVERSE(A))))
ESTIMATE = NORM(Z)/NORM(Y) WHERE A*Z = Y AND TRANS(A)*Y = E.
TRANS(A) IS THE TRANSPOSE OF A. THE COMPONENTS OF E ARE 
CHosen TO CAUSE MAXIMUM LOCAL GROWTH IN THE ELEMENTS OF W WHERE 
TRANS(U)*W = E. THE VECTORS ARE FREQUENTLY RESCALED TO AVOID 
OVERFLOW.

SOLVE TRANS(U)*W = E
EK = 1.0EO
DO 20 J = 1, N 
Z(J) = 0.0EO
20 CONTINUE

DO 100 K = 1, N
IF (Z(K) .NE. 0.0EO) EK = SIGN(EK,-Z(K))
IF (ABS(EK-Z(K)) .LE. ABS(A(K,K))) GO TO 30
S = ABS(A(K,K))/ABS(EK-Z(K))
CALL SSCL(L,N,S,Z,1)
EK = S*EK
30 CONTINUE
WK = EK - Z(K)
WKM = -EK - Z(K)
S = ABS(WK)
SM = ABS(WKM)
IF (A(K,K) .EQ. 0.0EO) GO TO 40
WK = WK/A(K,K)
WKM = WKM/A(K,K)
GO TO 50
40 CONTINUE
WK = 1.0EO
WKM = 1.0EO

75
CONTINUE
KP1 = K + 1
IF (KP1 .GT. N) GO TO 90
DO 60 J = KP1, N
   SM = SM + ABS(Z(J)+WKM*A(K,J))
   Z(J) = Z(J) + WKM*A(K,J)
   S = S + ABS(Z(J))
60 CONTINUE
IF (S .GE. SM) GO TO 80
   T = WKM - WK
   WK = WKM
   DO TO J = KP1, N
   Z(J) = Z(J) + T*A(K,J)
   CONTINUE
70 CONTINUE
90 CONTINUE
Z(K) = WK
100 CONTINUE
   S = 1.0E0/SASUM(N,Z,1)
   CALL SSCAL(N,S,Z,1)
   CALL SOLVE TRANS(L)*Y = V
   DO 120 KB = 1, N
      K = N - 1 - KB
      IF (K .LT. N) Z(K) = Z(K) + SDOT(N-K,A(K+1,K),1,Z(K+1),1)
      IF (ABS(Z(K)) .LE. 1.0E0) GO TO 110
      S = 1.0E0/ABS(Z(K))
      CALL SSCAL(N,S,Z,1)
110 CONTINUE
      L = IPVT(K)
      T = Z(L)
      Z(L) = Z(K)
      Z(K) = T
      IF (K .LT. N) CALL SAXPY(N-K,T,A(K+1,K),1,Z(K+1),1)
      IF (ABS(Z(K)) .LE. 1.0E0) GO TO 130
      S = 1.0E0/ABS(Z(K))
      CALL SSCAL(N,S,Z,1)
      YNORM = S*YNORM
130 CONTINUE
140 CONTINUE
   S = 1.0E0/SASUM(N,Z,1)
   CALL SSCAL(N,S,Z,1)
   CALL SOLVE L*V = Y
   DO 160 K = 1, N
      L = IPVT(K)
      T = Z(L)
      Z(L) = Z(K)
      Z(K) = T
      IF (K .LT. N) CALL SAXPY(N-K,T,A(K+1,K),1,Z(K+1),1)
      IF (ABS(Z(K)) .LE. 1.0E0) GO TO 150
      S = ABS(A(K,K))/ABS(Z(K))
      CALL SSCAL(N,S,Z,1)
      YNORM = S*YNORM
150 CONTINUE
   IF (A(K,K) .NE. 0.0E0) Z(K) = Z(K)/A(K,K)
   IF (A(K,K) .EQ. 0.0E0) Z(K) = 1.0E0
   T = -Z(K)
   CALL SAXPY(K-1,T,A(1,K),1,Z(1),1)
160 CONTINUE

76
MAKE ZNORM = 1.0
S = 1.0*E0/SASUM(N,Z,1)
CALL SSCAL(N,S,2,1)
YNORM = S*YNORM

IF (ANORM .LT. 0.0) RCOND = YNORM/ANORM
IF (ANORM .EQ. 0.0) RCOND = 0.0
RETURN
END
SUBROUTINE SGEFA(A,LDA,N,IPVT,INFO)
INTEGER LDA,N,IPVT(N),INFO
REAL A(LDA,N)
SGEFA FACTORS A REAL MATRIX BY GAUSSIAN ELIMINATION.
SGEFA IS USUALLY CALLED BY SGECO, BUT IT CAN BE CALLED
DIRECTLY WITH A SAVING IN TIME IF RCOND IS NOT NEEDED.
(TIME FOR SGECO) = (1 + 9/N)*(TIME FOR SGEFA).

ON ENTRY
A REAL(LDA, N)
THE MATRIX TO BE FACTORED.
LDA INTEGER
THE LEADING DIMENSION OF THE ARRAY A.
N INTEGER
THE ORDER OF THE MATRIX A.

ON RETURN
A AN UPPER TRIANGULAR MATRIX AND THE MULTIPLIERS
WHICH WERE USED TO OBTAIN IT.
THE FACTORIZATION CAN BE WRITTEN A = L*U WHERE
L IS A PRODUCT OF PERMUTATION AND UNIT LOWER
TRIANGULAR MATRICES AND U IS UPPER TRIANGULAR.
IPVT INTEGER(N)
AN INTEGER VECTOR OF PIVOT INDICES.
INFO INTEGER
= 0 NORMAL VALUE.
= K IF U(K,K) .EQ. 0.0 THIS IS NOT AN ERROR
CONDITION FOR THIS SUBROUTINE, BUT IT DOES
INDICATE THAT SGESL OR SGEDI WILL DIVIDE BY ZERO
IF CALLED. USE RCOND IN SGECO FOR A RELIABLE
INDICATION OF SINGULARITY.

LINPACK. THIS VERSION DATED 07/14/77.
CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.
SUBROUTINES AND FUNCTIONS
BLAS SAXPY,SSCAL,ISAMAX
INTERNAL VARIABLES
REAL T
INTEGER ISAMAX,J,K,KP1,L,NM1
GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
INFO = 0
NM1 = N - 1
IF (NM1 .LT. 1) GO TO 70
DO 60 K = 1, NM1
    KP1 = K + 1
    FIND L = PIVOT INDEX
L = ISAMAX(N-K+1,A(K,K),1) + K - 1
IPVT(K) = L

ZERO PIVOT IMPLIES THIS COLUMN ALREADY TRIANGULARIZED
IF (A(L,K) .EQ. 0.0E0) GO TO 40

INTERCHANGE IF NECESSARY
IF (L .EQ. K) GO TO 10
T = A(L,K)
A(L,K) = A(K,K)
A(K,K) = T
CONTINUE

10 COMPUTE MULTIPLIERS
T = -1.0E0/A(K,K)
CALL SSCAL(N-K,T,A(K+1,K),1)

ROW ELIMINATION WITH COLUMN INDEXING
DO 30 J = KP1, N
   T = A(L,J)
   IF (L .EQ. K) GO TO 20
   A(L,J) = A(K,J)
   A(K,J) = T
   CONTINUE
   CALL SAXPY(N-K,T,A(K+1,K),1,A(K+1,J),1)
30 CONTINUE
GO TO 50
40 CONTINUE
INFO = K
CONTINUE
60 CONTINUE
CONTINUE
IPVT(N) = N
IF (A(N,N) .EQ. 0.0E0) INFO = N
RETURN
END

SUBROUTINE SGEDI(A,LDA,N,IPVT,DET,WORK,JOB)
INTEGER LDA,N,IPVT(*),JOB
REAL A(LDA,*)

SGEDI COMPUTES THE DETERMINANT AND INVERSE OF A MATRIX
USING THE FACTORS COMPUTED BY SGECO OR SGEFA.

ON ENTRY
A REAL(LDA, N)
THE OUTPUT FROM SGECO OR SGEFA.
LDA INTEGER
THE LEADING DIMENSION OF THE ARRAY A .
N INTEGER
THE ORDER OF THE MATRIX A .
IPVT INTEGER(N)
THE PIVOT VECTOR FROM SGECO OR SGEFA.
WORK REAL(N)
WORK VECTOR. CONTENTS DESTROYED.
JOB INTEGER
= 11 BOTH DETERMINANT AND INVERSE. 
= 01 INVERSE ONLY.
= 10 DETERMINANT ONLY.
ON RETURN

A   INVERSE OF ORIGINAL MATRIX IF REQUESTED.
    OTHERWISE UNCHANGED.

DET   REAL(2)
DETERMINANT OF ORIGINAL MATRIX IF REQUESTED.
    OTHERWISE NOT REFERENCED.
    DETERMINANT = DET(1) * 10.0**DET(2)
    WITH 1.0 .LE. ABS(DET(1)) .LT. 10.0
    OR DET(1) .EQ. 0.0.

ERROR CONDITION

A DIVISION BY ZERO WILL OCCUR IF THE INPUT FACTOR CONTAINS
A ZERO ON THE DIAGONAL AND THE INVERSE IS REQUESTED.
IT WILL NOT OCCUR IF THE SUBROUTINES ARE CALLED CORRECTLY
AND IF SGECO HAS SET RCOND .GT. 0.0 OR SGEFA HAS SET
INFO .EQ. 0.

LINPACK. THIS VERSION DATED 07/14/77.
CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.

SUBROUTINES AND FUNCTIONS

ELAS SAXPY, SSCAL, SSWAP
FORTRAN ABS, MOD

INTERNAL VARIABLES

REAL T
REAL TEN
INTEGER I, J, K, KB, KP1, L, NM1

COMPUTE DETERMINANT

IF (JOB/10 .EQ. 0) GO TO 70
DET(1) = 1.0E0
DET(2) = 0.0E0
TEN = 10.0E0
DO 50 I = 1, N
   IF (IPVT(I) .NE. I) DET(1) = -DET(1)
   DET(1) = A(I, I) * DET(1)
   ...EXIT

10   IF (DET(1) .EQ. 0.0E0) GO TO 60
   IF (ABS(DET(1)) .GE. 1.0E0) GO TO 20
      DET(1) = TEN * DET(1)
      DET(2) = DET(2) - 1.0E0
   GO TO 10
   CONTINUE
20   IF (ABS(DET(1)) .LT. TEN) GO TO 40
      DET(1) = DET(1) / TEN
      DET(2) = DET(2) + 1.0E0
   GO TO 30
   CONTINUE
30   CONTINUE
40   CONTINUE
50   CONTINUE
60   CONTINUE
70   CONTINUE

COMPUTE INVERSE(U)

IF (MOD(JOB, 10) .EQ. 0) GO TO 150
DO 100 K = 1, N
   A(K, K) = 1.0E0 / A(K, K)
   T = -A(K, K)
   CALL SSCAL(K-1, T, A(1, K), 1)
   KP1 = K + 1
   IF (N .LT. KP1) GO TO 90
   DO 80 J = KP1, N
      A(K, J) = T
80   CONTINUE
   GO TO 90
100  CONTINUE
90   CONTINUE
79
A(K,J) = 0.0E0
CALL SAXPY(K,T,A(1,K),1,A(1,J),1)

CONTINUE
CONTINUE
CONTINUE
CONTINUE

FORM INVERSE(U)*INVERSE(L)

NM1 = N - 1
IF (NM1 .LT. 1) GO TO 140
DO 130 KB = 1, NM1
  K = N - KB
  KP1 = K + 1
  DO 110 I = KP1, N
    WORK(I) = A(I,K)
    A(I,K) = 0.0E0
  CONTINUE
  DO 120 J = KP1, N
    T = WORK(J)
    CALL SAXPY(N,T,A(1,J),1,A(1,K),1)
  CONTINUE
  L = IPVT(K)
  IF (L .NE. K) CALL SSWAP(N,A(1,K),1,A(1,L),1)
CONTINUE
CONTINUE
CONTINUE
CONTINUE
RETURN
END

INTEGER FUNCTION ISAMAX (N,SX,INCX)

FINDS THE INDEX OF ELEMENT HAVING MAX. ABSOLUTE VALUE.
JACK DONGARRA, LINPACK, 3/11/78.

REAL SX(1), SMAX
INTEGER I, INCX, IX, N

ISAMAX=0
IF (N.LT.1) RETURN
ISAMAX=1
IF (N.EQ.1) RETURN
IF (INCX.EQ.1) GO TO 30

CODE FOR INCREMENT NOT EQUAL TO 1

IX=1
SMAX=ABS(SX(1))
IX=IX+INCX
DO 20 I=2,N
  IF (ABS(SX(IX)).LE.SMAX) GO TO 10
  ISAMAX=I
  SMAX=ABS(SX(IX))
20 CONTINUE
CONTINUE
RETURN

CODE FOR INCREMENT EQUAL TO 1

30 SMAX=ABS(SX(1))
DO 40 I=2,N
  IF (ABS(SX(I)).LE.SMAX) GO TO 40
  ISAMAX=I
  SMAX=ABS(SX(I))
40 CONTINUE
RETURN
END

REAL FUNCTION SASUM (N,SX,INCX)

TAKES THE SUM OF THE ABSOLUTE VALUES.
USES UNROLLED LOOPS FOR INCREMENT EQUAL TO ONE.
JACK DONGARRA, LINPACK, 3/11/78.
REAL SX(1), STEM
INTEGER I, INCX, M, MP1, N, NINCX

C
SASUM=0.0EO
STEMP=0.0EO
IF (N.LE.0) RETURN
IF (INCX.EQ.1) GO TO 20

CODE FOR INCREMENT NOT EQUAL TO 1

NINCX=N*INCX
DO 10 I=1,NINCX,INCX
STEMP=STEMP+ABS(SX(I))
CONTINUE
SASUM=STEMP
RETURN

CODE FOR INCREMENT EQUAL TO 1
CLEAN-UP LOOP

20 M=MOD(N,6)
IF (M.EQ.0) GO TO 40
DO 30 I=1,M
STEMP=STEMP+ABS(SX(I))
30 CONTINUE
NINCX=N*INCX
DO 10 I=1,NINCX,INCX
STEMP=STEMP+ABS(SX(I))+ABS(SX(I+1))+ABS(SX(I+2))+ABS(SX(I+3))+ABS(SX(I+4))+ABS(SX(I+5))
CONTINUE
60 SASUM=STEMP
RETURN
END

SUBROUTINE SAXPY (N,SA,SX,INCX,SY,INCY)
CONSTANT TIMES A VECTOR PLUS A VECTOR.
USES UNROLLED LOOP FOR INCREMENTS EQUAL TO ONE.
JACK DONGARRA, LINPACK, 3/11/78.
REAL SX(1), SY(1), SA
INTEGER I, INCX, INCY, IX, IY, M, MP1, N

IF (N.LE.0) RETURN
IF (SA.EQ.0.0) RETURN
IF (INCX.EQ.1.AND.INCY.EQ.1) GO TO 20

CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
NOT EQUAL TO 1

IX=1
IY=1
IF (INCX.LT.0) IX=(-N+1)*INCX+1
IF (INCY.LT.0) IY=(-N+1)*INCY+1
DO 10 I=1,N
SY(IY)=SY(IY)+SA*SX(IX)
IX=IX+INCX
IY=IY+INCY
CONTINUE
RETURN

CODE FOR BOTH INCREMENTS EQUAL TO 1
CLEAN-UP LOOP

20 M=MOD(N,4)
IF (M.EQ.0) GO TO 40
DO 30 I=1,M
SY(I)=SY(I)+SA*SX(I)
30 CONTINUE
IF (N.LT.4) RETURN
40 MP1=M+1
   DO 50 I=MP1,N, M
      SY(I)=SX(I)+SA*SX(I)
      SY(I+1)=SY(I+1)+SA*SX(I+1)
      SY(I+2)=SY(I+2)+SA*SX(I+2)
      SY(I+3)=SY(I+3)+SA*SX(I+3)
   CONTINUE
CONTINUE
RETURN
END

REAL FUNCTION SDOT (N,SX,INCX,SY,INCY)

FORMS THE DOT PRODUCT OF TWO VECTORS.
USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.
JACK DONGARRA, LINPACK, 3/11/78.

REAL SX(1), SY(1), STEMP
INTEGER I, INCX, INCY, IX, IY, M, MP1, N

STEMP=0.0E0
SDOT=0.0E0
IF (N.LE.0) RETURN
IF (INCX.EQ.1.AND.INCY.EQ.1) GO TO 20

CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS NOT EQUAL TO 1

IX=1
IY=1
IF (INCY.LT.0) IX=(-N+I)*INCX+1
IF (INCY.LT.0) IX=(-N+I)*INCX+1
DO 10 I=1,N
STEMP=STEMP+SX(IX)*SY(IY)
IX=IX+INCX
IY=IY+INCY
10 CONTINUE
SDOT=STEMP
RETURN

CODE FOR BOTH INCREMENTS EQUAL TO 1

CLEAN-UP LOOP

20 M=MOD(N,5)
   IF (M.EQ.0) GO TO 40
   DO 30 I=1,M
      STEMP=STEMP+SX(I)*SY(I)
   CONTINUE
30 CONTINUE
   IF (N.LT.5) GO TO 60
40 MP1=M+1
   DO 50 I=MP1,N,5
      STEMP=STEMP+SX(I)*SY(I+1)*SY(I+1)+SX(I+2)*SY(I+2)+SX(I+3)*SY
         I+1)+SX(I+2)+SX(I+3)+SX(I+4)+SY(I+4)
   CONTINUE
50 CONTINUE
60 SDOT=STEMP
RETURN
END

SUBROUTINE SSCAL (N,SA,SX,INCX)

SCALE A VECTOR BY A CONSTANT.
USES UNROLLED LOOPS FOR INCREMENT EQUAL TO 1.
JACK DONGARRA, LINPACK, 3/11/78.

REAL SA, SX(1)
INTEGER I, INCX, M, MP1, N, NINCX

IF (N.LE.0) RETURN
IF (INCX.EQ.1) GO TO 20

CODE FOR INCREMENT NOT EQUAL TO 1
SUBROUTINE SSWAP (N, SX, INCX, SY, INCY)
INTERCHANGES TWO VECTORS.
USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO 1.
JACK DONGARRA, LINPACK, 3/11/78.
REAL SX(I), SY(I), STEMP
INTEGER I, INCX, INCY, IX, IY, M, MP1, N
IF N.LE.0 RETURN
IF INCX.EQ.1.AND.INCY.EQ.1) GO TO 20
CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS NOT EQUAL
TO 1
IX=1
IY=1
IF (INCX.LT.0) IX=(-N+1)*INCX+1
IF (INCY.LT.0) IY=(-N+1)*INCY+1
DO 10 I=1,N
STEMP=SX(IX)
SX(IX)=SY(IY)
SY(IY)=STEMP
IX=IX+INCX
IY=IY+INCY
10 CONTINUE
RETURN
CODE FOR BOTH INCREMENTS EQUAL TO 1
CLEAN-UP LOOP
20 M=MOD(N,3)
IF (M.EQ.0) GO TO 40
DO 30 I=1,M
STEMP=SX(I)
SX(I)=SY(I)
SY(I)=STEMP
30 CONTINUE
IF (N.LT.3) RETURN
40 MP1=M+1
DO 50 I=MP1,N,3
STEMP=SX(I)
SX(I)=SY(I)
SY(I)=STEMP
50 CONTINUE
RETURN
APPENDIX C

DASH TEST PROBLEM (with Output)

Problem input:

```
1 0
A 1 0 0 0 0 0 0 0 0 8.0225E-07
B 2 1 0 0 0 0 0 0 0 1.6045E-06
24 1 2 2 4 3 1
0.0 5.0 10.0
223 0.0 5.03 3
108 1108 33 2108
1000.0 1000.0 1000.0 1000.0 3
0.0 2.0 3.0 5.0 3
5.426E-06 0.0 0.0 2.713E-06 3
1.266E-05 0.0 0.0 6.330E-06 3
104 0.0 3
1.808E-06 0.0 0.0 9.042E-07 3
104 0.03
1.00E+10 5.00E+09
102 0.0 3
148 0.0 3
148 0.0 3
10000.0 1000.0 1000.0 1000.0 3
0.0 2.0 3.0 5.0 3
5.426E-06 0.0 0.0 2.713E-06 3
1.266E-05 0.0 0.0 6.330E-06 3
104 0.0 3
1.808E-06 0.0 0.0 9.042E-07 3
104 0.03
1.00E+10 5.00E+09
102 0.0 3
148 0.0 3
148 0.0 3
10000.0 1000.0 1000.0 1000.0 3
0.0 2.0 3.0 5.0 3
5.426E-06 0.0 0.0 2.713E-06 3
1.266E-05 0.0 0.0 6.330E-06 3
104 0.0 3
1.808E-06 0.0 0.0 9.042E-07 3
104 0.03
1.00E+10 5.00E+09
102 0.0 3
148 0.0 3
148 0.0 3
10000.0 1000.0 1000.0 1000.0 3
0.0 2.0 3.0 5.0 3
5.426E-06 0.0 0.0 2.713E-06 3
1.266E-05 0.0 0.0 6.330E-06 3
104 0.0 3
1.808E-06 0.0 0.0 9.042E-07 3
104 0.03
1.00E+10 5.00E+09
102 0.0 3
148 0.0 3
148 0.0 3
10000.0 1000.0 1000.0 1000.0 3
0.0 2.0 3.0 5.0 3
5.426E-06 0.0 0.0 2.713E-06 3
1.266E-05 0.0 0.0 6.330E-06 3
104 0.0 3
1.808E-06 0.0 0.0 9.042E-07 3
104 0.03
1.00E+10 5.00E+09
102 0.0 3
148 0.0 3
148 0.0 3
```
Problem output:

DECAY CHAINS AND NUCLIDE RELATED DATA

<table>
<thead>
<tr>
<th>NUCLIDE ID</th>
<th>DECAY PARENT 1</th>
<th>DECAY PARENT 2</th>
<th>CAPTURE PARENT 1</th>
<th>CAPTURE PARENT 2</th>
<th>N-2N</th>
<th>N-ALPHA</th>
<th>N-P</th>
<th>DECAY CONSTANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8.0225E-07</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.6045E-06</td>
</tr>
</tbody>
</table>

THE GEOMETRY FOR THIS PROBLEM IS A SLAB.
THE LEFT BOUNDARY CONDITION IS = 2
THE RIGHT BOUNDARY CONDITION IS = 2

RADIUS

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>MATERIALS</th>
<th>TEMPERATURES</th>
<th>TEMP RADIUS</th>
<th>DIJ-0</th>
<th>AIJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>1</td>
<td>.1000E+04</td>
<td>.2000E+01</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>.1250E+01</td>
<td>1</td>
<td>.1000E+04</td>
<td>.3000E+01</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>.2500E+01</td>
<td>1</td>
<td>.1000E+04</td>
<td>.5000E+01</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>.3750E+01</td>
<td>1</td>
<td>.1000E+04</td>
<td>.27130E-05</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>.5000E+01</td>
<td>1</td>
<td>.1000E+04</td>
<td>.27130E-05</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

DECAY CONSTANT

<table>
<thead>
<tr>
<th>DECAY CONSTANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.0225E-07</td>
</tr>
<tr>
<td>1.6045E-06</td>
</tr>
</tbody>
</table>
LEFT CONCEN

\[1.0000E+11 \quad 5.0000E+10\]

RIGHT CONCEN

0.

INITIAL CONC

48

0.

SOURCE INPUT

48

0.

SOURCE INPUT

48

0.

MESH TEMPERATURES

\[1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \]

RBAR

\[1.0417E+00 \quad 3.1250E+00 \quad 5.2083E+00 \quad 7.2917E+00 \quad 9.3750E+00 \quad 1.1458E+01 \]

\[1.3542E+01 \quad 1.5625E+01 \quad 1.7708E+01 \quad 1.9792E+01 \quad 2.1875E+01 \quad 2.3958E+01 \]

\[2.6042E+01 \quad 2.8125E+01 \quad 3.0208E+01 \quad 3.2292E+01 \quad 3.4375E+01 \quad 3.6458E+01 \]

\[3.8542E+01 \quad 4.0625E+01 \quad 4.2708E+01 \quad 4.4792E+01 \quad 4.6875E+01 \quad 4.8958E+01 \]

MESH TEMPERATURES

\[1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \quad 1.0000E+04 \]
<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>LEFT BOUNDARY</th>
<th>CELL 1</th>
<th>CELL 2</th>
<th>CELL 3</th>
<th>CELL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>1.00000E+10</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>5.00000E+09</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 5</th>
<th>CELL 6</th>
<th>CELL 7</th>
<th>CELL 8</th>
<th>CELL 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 10</th>
<th>CELL 11</th>
<th>CELL 12</th>
<th>CELL 13</th>
<th>CELL 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 15</th>
<th>CELL 16</th>
<th>CELL 17</th>
<th>CELL 18</th>
<th>CELL 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 20</th>
<th>CELL 21</th>
<th>CELL 22</th>
<th>CELL 23</th>
<th>CELL 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>RIGHT BOUNDARY</th>
<th>CELL 2</th>
<th>CELL 7</th>
<th>CELL 12</th>
<th>CELL 17</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

NO. OF ATOMS = 0.

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>LEFT BOUNDARY</th>
<th>CELL 1</th>
<th>CELL 2</th>
<th>CELL 3</th>
<th>CELL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>1.00000E+10</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>5.00000E+09</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 5</th>
<th>CELL 6</th>
<th>CELL 7</th>
<th>CELL 8</th>
<th>CELL 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 10</th>
<th>CELL 11</th>
<th>CELL 12</th>
<th>CELL 13</th>
<th>CELL 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 15</th>
<th>CELL 16</th>
<th>CELL 17</th>
<th>CELL 18</th>
<th>CELL 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 20</th>
<th>CELL 21</th>
<th>CELL 22</th>
<th>CELL 23</th>
<th>CELL 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>RIGHT BOUNDARY</th>
<th>CELL 2</th>
<th>CELL 7</th>
<th>CELL 12</th>
<th>CELL 17</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

NO. OF ATOMS = 1.446715010794452E+10
### CELL CONCENTRATIONS AT 4.00000E+00 DAYS (CONCENTRATIONS IN ATOMS/CC)

<table>
<thead>
<tr>
<th>ISOTOPE BOUNDARY</th>
<th>CELL 1</th>
<th>CELL 2</th>
<th>CELL 3</th>
<th>CELL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.00000E+10</td>
<td>9.42778E+09</td>
<td>8.34731E+09</td>
<td>7.33072E+09</td>
</tr>
<tr>
<td>B</td>
<td>5.00000E+09</td>
<td>4.70421E+09</td>
<td>4.11867E+09</td>
<td>3.55015E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE 5</th>
<th>CELL 5</th>
<th>CELL 6</th>
<th>CELL 7</th>
<th>CELL 8</th>
<th>CELL 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5.48958E+09</td>
<td>4.66485E+09</td>
<td>3.90357E+09</td>
<td>3.20501E+09</td>
<td>2.75003E+09</td>
</tr>
<tr>
<td>B</td>
<td>2.49978E+09</td>
<td>2.03055E+09</td>
<td>1.60328E+09</td>
<td>1.21885E+09</td>
<td>9.74220E+08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE 10</th>
<th>CELL 10</th>
<th>CELL 11</th>
<th>CELL 12</th>
<th>CELL 13</th>
<th>CELL 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2.50296E+09</td>
<td>2.28157E+09</td>
<td>2.08539E+09</td>
<td>1.91402E+09</td>
<td>1.76597E+09</td>
</tr>
<tr>
<td>B</td>
<td>8.44765E+08</td>
<td>7.32089E+08</td>
<td>6.35525E+08</td>
<td>5.54337E+08</td>
<td>4.87757E+08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE 15</th>
<th>CELL 15</th>
<th>CELL 16</th>
<th>CELL 17</th>
<th>CELL 18</th>
<th>CELL 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.64372E+09</td>
<td>1.54371E+09</td>
<td>1.23472E+09</td>
<td>8.34885E+09</td>
<td>5.49025E+09</td>
</tr>
<tr>
<td>B</td>
<td>4.35023E+08</td>
<td>3.95409E+08</td>
<td>2.86757E+08</td>
<td>1.67269E+08</td>
<td>9.49803E+07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE 20</th>
<th>CELL 20</th>
<th>CELL 21</th>
<th>CELL 22</th>
<th>CELL 23</th>
<th>CELL 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3.50278E+08</td>
<td>2.15566E+08</td>
<td>1.25588E+08</td>
<td>6.46198E+07</td>
<td>1.98114E+07</td>
</tr>
<tr>
<td>B</td>
<td>5.26285E+07</td>
<td>2.84312E+07</td>
<td>1.47910E+07</td>
<td>6.97871E+06</td>
<td>2.03949E+06</td>
</tr>
</tbody>
</table>

### RIGHT ISOTOPE BOUNDARY

A 0.
B 0.

**NO. OF ATOMS = 2.022416554881128E+10**

### CELL CONCENTRATIONS AT 6.00000E+00 DAYS (CONCENTRATIONS IN ATOMS/CC)

<table>
<thead>
<tr>
<th>ISOTOPE BOUNDARY</th>
<th>CELL 1</th>
<th>CELL 2</th>
<th>CELL 3</th>
<th>CELL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.00000E+10</td>
<td>9.48391E+09</td>
<td>8.53455E+09</td>
<td>7.60570E+09</td>
</tr>
<tr>
<td>B</td>
<td>5.00000E+09</td>
<td>4.75291E+09</td>
<td>4.26232E+09</td>
<td>3.78171E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE 5</th>
<th>CELL 5</th>
<th>CELL 6</th>
<th>CELL 7</th>
<th>CELL 8</th>
<th>CELL 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5.86166E+09</td>
<td>5.22303E+09</td>
<td>4.53792E+09</td>
<td>3.90483E+09</td>
<td>3.48873E+09</td>
</tr>
<tr>
<td>B</td>
<td>2.87162E+09</td>
<td>2.45057E+09</td>
<td>2.05634E+09</td>
<td>1.69104E+09</td>
<td>1.45176E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE 10</th>
<th>CELL 10</th>
<th>CELL 11</th>
<th>CELL 12</th>
<th>CELL 13</th>
<th>CELL 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3.26026E+09</td>
<td>3.05272E+09</td>
<td>2.86580E+09</td>
<td>2.69920E+09</td>
<td>2.55260E+09</td>
</tr>
<tr>
<td>B</td>
<td>1.32141E+09</td>
<td>1.20432E+09</td>
<td>1.10046E+09</td>
<td>1.00972E+09</td>
<td>9.31905E+08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE 15</th>
<th>CELL 15</th>
<th>CELL 16</th>
<th>CELL 17</th>
<th>CELL 18</th>
<th>CELL 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2.42566E+09</td>
<td>2.31805E+09</td>
<td>1.96336E+09</td>
<td>1.46521E+09</td>
<td>1.07141E+09</td>
</tr>
<tr>
<td>B</td>
<td>8.66799E+08</td>
<td>8.14129E+08</td>
<td>6.51964E+08</td>
<td>4.43919E+08</td>
<td>2.95522E+08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE 20</th>
<th>CELL 20</th>
<th>CELL 21</th>
<th>CELL 22</th>
<th>CELL 23</th>
<th>CELL 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7.63947E+08</td>
<td>5.25345E+08</td>
<td>3.38961E+08</td>
<td>1.89046E+08</td>
<td>6.06463E+07</td>
</tr>
<tr>
<td>B</td>
<td>1.92098E+08</td>
<td>1.21154E+08</td>
<td>7.25530E+07</td>
<td>3.82733E+07</td>
<td>1.19153E+07</td>
</tr>
</tbody>
</table>

### RIGHT ISOTOPE BOUNDARY

A 0.
B 0.

**NO. OF ATOMS = 2.432060341188196E+10**

88
### CELL CONCENTRATIONS AT 8.00000E+00 DAYS

(CONCENTRATIONS IN ATOMS/CC)

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>LEFT BOUNDARY</th>
<th>CELL 1</th>
<th>CELL 2</th>
<th>CELL 3</th>
<th>CELL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>5.00000E+09</td>
<td>4.78213E+09</td>
<td>4.34920E+09</td>
<td>3.92396E+09</td>
<td>3.51057E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 5</th>
<th>CELL 6</th>
<th>CELL 7</th>
<th>CELL 8</th>
<th>CELL 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6.25491E+09</td>
<td>5.57308E+09</td>
<td>4.94004E+09</td>
<td>4.35369E+09</td>
<td>3.96811E+09</td>
</tr>
<tr>
<td>B</td>
<td>3.11256E+09</td>
<td>2.73290E+09</td>
<td>2.37399E+09</td>
<td>2.03777E+09</td>
<td>1.81489E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 10</th>
<th>CELL 11</th>
<th>CELL 12</th>
<th>CELL 13</th>
<th>CELL 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3.75326E+09</td>
<td>3.55795E+09</td>
<td>3.38056E+09</td>
<td>3.22076E+09</td>
<td>3.07825E+09</td>
</tr>
<tr>
<td>B</td>
<td>1.69186E+09</td>
<td>1.57968E+09</td>
<td>1.47843E+09</td>
<td>1.38814E+09</td>
<td>1.30877E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 15</th>
<th>CELL 16</th>
<th>CELL 17</th>
<th>CELL 18</th>
<th>CELL 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2.95271E+09</td>
<td>2.84382E+09</td>
<td>2.47363E+09</td>
<td>1.93189E+09</td>
<td>1.48164E+09</td>
</tr>
<tr>
<td>B</td>
<td>1.24026E+09</td>
<td>1.18248E+09</td>
<td>9.93692E+09</td>
<td>7.32038E+09</td>
<td>5.28611E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 20</th>
<th>CELL 21</th>
<th>CELL 22</th>
<th>CELL 23</th>
<th>CELL 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.10836E+09</td>
<td>7.97676E+09</td>
<td>5.35402E+09</td>
<td>3.07533E+09</td>
<td>1.00231E+09</td>
</tr>
<tr>
<td>B</td>
<td>3.72667E+08</td>
<td>2.53864E+08</td>
<td>1.62631E+08</td>
<td>9.02763E+07</td>
<td>2.88938E+07</td>
</tr>
</tbody>
</table>

### RIGHT ISOTOPE BOUNDARY

| A | 0 |
| B | 0 |

**NO. OF ATOMS = 2.732894842037402E+10**

### CELL CONCENTRATIONS AT 1.00000E+01 DAYS

(CONCENTRATIONS IN ATOMS/CC)

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>LEFT BOUNDARY</th>
<th>CELL 1</th>
<th>CELL 2</th>
<th>CELL 3</th>
<th>CELL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.00000E+10</td>
<td>9.51607E+09</td>
<td>8.61555E+09</td>
<td>7.77410E+09</td>
<td>7.14074E+09</td>
</tr>
<tr>
<td>B</td>
<td>5.00000E+09</td>
<td>4.78213E+09</td>
<td>4.34920E+09</td>
<td>3.92396E+09</td>
<td>3.64930E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 5</th>
<th>CELL 6</th>
<th>CELL 7</th>
<th>CELL 8</th>
<th>CELL 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6.44833E+09</td>
<td>5.80458E+09</td>
<td>5.20681E+09</td>
<td>4.65250E+09</td>
<td>4.28595E+09</td>
</tr>
<tr>
<td>B</td>
<td>3.26681E+09</td>
<td>2.93976E+09</td>
<td>2.61023E+09</td>
<td>2.29977E+09</td>
<td>2.09257E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 10</th>
<th>CELL 11</th>
<th>CELL 12</th>
<th>CELL 13</th>
<th>CELL 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4.03296E+09</td>
<td>3.89564E+09</td>
<td>3.72633E+09</td>
<td>3.57197E+09</td>
<td>3.43314E+09</td>
</tr>
<tr>
<td>B</td>
<td>1.97730E+09</td>
<td>1.87121E+09</td>
<td>1.77439E+09</td>
<td>1.68690E+09</td>
<td>1.60874E+09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 15</th>
<th>CELL 16</th>
<th>CELL 17</th>
<th>CELL 18</th>
<th>CELL 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3.30505E+09</td>
<td>3.20077E+09</td>
<td>2.82418E+09</td>
<td>2.25954E+09</td>
<td>1.77606E+09</td>
</tr>
<tr>
<td>B</td>
<td>1.53988E+09</td>
<td>1.48025E+09</td>
<td>1.27831E+09</td>
<td>9.85283E+08</td>
<td>7.45147E+08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISOTOPE</th>
<th>CELL 20</th>
<th>CELL 21</th>
<th>CELL 22</th>
<th>CELL 23</th>
<th>CELL 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.36081E+09</td>
<td>1.00111E+09</td>
<td>6.84443E+08</td>
<td>3.98429E+08</td>
<td>1.30772E+08</td>
</tr>
<tr>
<td>B</td>
<td>5.49730E+08</td>
<td>3.90608E+08</td>
<td>2.59383E+08</td>
<td>1.47840E+08</td>
<td>4.79860E+07</td>
</tr>
</tbody>
</table>

### RIGHT ISOTOPE BOUNDARY

| A | 0 |
| B | 0 |

**NO. OF ATOMS = 2.953406154721277E+10**
Distribution:

221 - Nuclear Regulatory Commission, Washington, DC.
2 - Technical Information Center, Oak Ridge National Laboratory, Oak Ridge, TN.
50 - Los Alamos Scientific Laboratory, Los Alamos, NM.

273 copies printed
<table>
<thead>
<tr>
<th>Material</th>
<th>$ 2.00</th>
<th>1.51-550</th>
<th>7.25</th>
<th>551-1725</th>
<th>10.75</th>
<th>256-600</th>
<th>13.00</th>
<th>501-125</th>
<th>15.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>504-0255</td>
<td>4.00</td>
<td>151-175</td>
<td>8.00</td>
<td>376-1000</td>
<td>15.00</td>
<td>476-510</td>
<td>13.35</td>
<td>526-150</td>
<td>15.50</td>
</tr>
<tr>
<td>102-0500</td>
<td>4.50</td>
<td>176-200</td>
<td>9.00</td>
<td>301-325</td>
<td>11.75</td>
<td>426-430</td>
<td>14.00</td>
<td>551-175</td>
<td>16.25</td>
</tr>
<tr>
<td>103-0750</td>
<td>5.25</td>
<td>201-225</td>
<td>9.25</td>
<td>236-350</td>
<td>12.00</td>
<td>451-475</td>
<td>14.50</td>
<td>579-600</td>
<td>16.30</td>
</tr>
<tr>
<td>1756-100</td>
<td>6.00</td>
<td>236-250</td>
<td>9.50</td>
<td>351-375</td>
<td>12.50</td>
<td>476-500</td>
<td>15.00</td>
<td>601-up</td>
<td>--1</td>
</tr>
</tbody>
</table>

1. Add $2.50 for each additional 100-page increment from 601 pages up.