NUREG/CR-0776
LA-7793-MS

Informal Report

University of California

DASH:

A Multicomponent Time-Dependent Concentration Diffusion with Radioactive Decay Program

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos, New Mexico 87545

An Affirmative Action/Equal Opportunity Employer

NUREG/CR-0776
LA-7793-MS
Informal Report

R-8

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

358 114

CONTENTS

CONTENTS (cont)

TABLES

358 116

FIGURES

358 118

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

by

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

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.[4] 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

$$\frac{\partial \vec{C}}{\partial t} = \nabla \cdot D \nabla \vec{C} - \lambda \vec{C} + \vec{S}, \tag{1}$$

where D is an n x n square positive definite diffusion matrix (cm^2/s), $\vec{C}$ is an n-component column vector representing isotopic concentrations ($atoms/cm^3$), λ is the decay matrix including branching rations[5] (1/s), and $\vec{S}$ 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 $\vec{C}(r, t) = \vec{C}(r, 0)$ and either homogeneous Newman ($D\nabla\vec{C} = 0$) or inhomogeneous or homogeneous Dirichlet ($\vec{C} = \vec{\sigma}$ or $\vec{C} = 0$) 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

$$V_k \frac{d}{dt} \vec{C}_k = -A_{k+1/2} \vec{J}_{k+1/2} + A_{k-1/2} \vec{J}_{k-1/2} - \lambda_k V_K \vec{C}_K + V_k \vec{S}_k. \tag{2}$$

In Eq. (2), $\vec{C}_k$ is the concentration vector averaged over the k^{th} cell, V_k is the volume of the k^{th} cell (diagonal matrix), and $\vec{A}_{k \pm 1/2}$ and $\vec{J}_{k \pm 1/2}$ are the area elements and current at the boundary between cell k 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, λ_k, is cell dependent only if neutron processes are included in addition to β decay.

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

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

The mesh spacing (Δr_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[6]

TABLE I
GEOMETRIC VARIABLES

Geometry	Δr_k	$A_{k + \frac{1}{2}}$	V_k
Slab	$r_{k + \frac{1}{2}} - r_{k - \frac{1}{2}}$	1	Δr_k
Cylinder	$r_{k + \frac{1}{2}} - r_{k - \frac{1}{2}}$	$2\pi r_{k + \frac{1}{2}}$	$\pi(r^2_{k + \frac{1}{2}} - r^2_{k - \frac{1}{2}})$
Sphere	$r_{k + \frac{1}{2}} - r_{k - \frac{1}{2}}$	$4\pi r^2_{k + \frac{1}{2}}$	$\frac{4\pi}{3}(r^3_{k + \frac{1}{2}} - r^3_{k - \frac{1}{2}})$

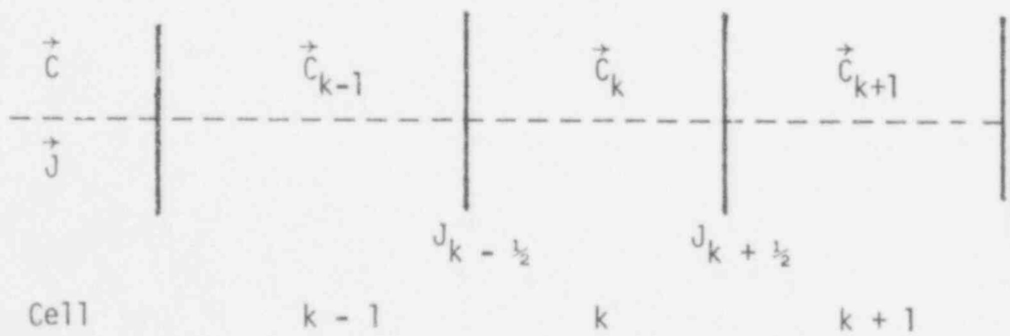

Fig. 1. Discrete mesh function representation.

358 121

$$\int_{r_k}^{r_{k+\frac{1}{2}}} \vec{J}\, dr = \int_{r_{k+\frac{1}{2}}}^{r_{k+1}} \vec{J}\, dr \tag{4}$$

is used or substituting $- D\nabla C$ for $\vec{J}$,

$$\int_{C_k}^{C_{k+\frac{1}{2}}} D\, d\vec{C} = \int_{C_{k+\frac{1}{2}}}^{C_{k+1}} D\, d\vec{C} \quad , \tag{5}$$

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

$$\vec{C}_{k+\frac{1}{2}} = (D_k + D_{k+1})^{-1} (D_k \vec{C}_k + D_{k+1} \vec{C}_{k+1}), \tag{6}$$

where 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

$$- \vec{J}_{k+\frac{1}{2}} = 4\, D_k (\vec{C}_{k+\frac{1}{2}} - \vec{C}_k)/(\Delta r_k + \Delta r_{k+1})$$

$$= 4\, D_k (D_k + D_{k+1})^{-1} D_{k+1} (\vec{C}_{k+1} - \vec{C}_k)/(\Delta r_k + \Delta r_{k+1}) \cdot \tag{7}$$

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

$$- \vec{J}_{k-\frac{1}{2}} = 4\, D_{k-1} (D_k + D_{k-1})^{-1} D_k (\vec{C}_k - \vec{C}_{k-1})/(\Delta r_{k-1} + \Delta r_k). \tag{8}$$

Note that $(D_k + D_{k+1})^{-1}$ represents a matrix inverse of a positive definite diffusion coefficient matrix. Substituting Eq. (7) and (8) into Eq. (2) results in

358 122

$$V_k \frac{d\vec{C}_k}{dt} = \bar{A}_k \vec{C}_{k+1} + \bar{K}_k \vec{C}_k + \bar{B}_k \vec{C}_{k-1} + V_k \vec{S}_k, \quad (9)$$

where the coefficient matrices are given by

$$\bar{A}_k = 4\, A_{k+1/2}\, D_k\, (D_k + D_{k+1})^{-1}\, D_{k+1} / (\Delta r_k + \Delta r_{k+1}),$$

$$\bar{B}_k = 4\, A_{k-1/2}\, D_{k-1}\, (D_{k-1} + D_k)^{-1}\, D_k / (\Delta r_{k-1} + \Delta r_k), \text{ and}$$

$$\bar{K}_k = -A_k - B_k - \lambda_k V_k. \quad (10)$$

Since $\bar{A}_k = \bar{B}_{k+1}$, a reciprocity relationship exists.[7]

The spatial boundary conditions treated are reflection ($\vec{J} = 0$), 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 $\vec{C}_0$ or $\vec{C}_{K+1}$, a [illegible]rresponds 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 h llow 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

$$\vec{J}_{k_0 - \frac{1}{2}} = 2\, D_{k_0} \left(\vec{C}_{k_0} - \vec{C}_{k_0 - \frac{1}{2}} \right) \Big/ \Delta r_{k_0}, \quad (11)$$

where the concentration vector $C_{k_0 - \frac{1}{2}}$ 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

$$\bar{A}_{k_0} = 4\, A_{k_0 + 1/2}\, D_{k_0} \left(D_{k_0} + D_{k_0 + 1} \right)^{-1} D_{k_0 + 1} / (\Delta r_{k_0} + \Delta r_{k_0 + 1}),$$

358 123

$$\bar{B}_{k_o}' = 2\,A_{k_o} - 1/2\ \ D_{k_o}/\Delta r_{k_o}, \qquad (12)$$

$$\bar{K}_{k_o}' = -\bar{A}_{k_o} - \bar{B}_{k_o}' - \lambda_{k_o} V_{k_o}.$$

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.

$$V_{k_o} \vec{S}_{k_o}' = V_{k_o} \vec{S}_{k_o} + \bar{B}_{k_o}' \vec{C}_{k_o - 1/2} \ . \qquad (13)$$

Similarily, 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).

$$\bar{A}_K' = 2\,A_K + 1/2\ D_K/\Delta r_K,$$

$$\bar{B}_K = 4\,A_K - 1/2\ D_{K-1}\ (D_{K-1} + D_K)^{-1}\ D_K/(\Delta r_{K-1} + \Delta r_K)\ , \qquad (14)$$

$$\bar{K}_K' = -\bar{A}_K' - \bar{B}_K - \lambda_K V_K.$$

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

$$V_K \vec{S}_K' = V_K \vec{S}_K + \bar{A}_K' \vec{C}_{K+1/2} \ . \qquad (15)$$

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

$$V \frac{d}{dt} \vec{C} = A\vec{C} + V\vec{S}, \qquad (16)$$

where

358 124

$$\vec{C} = \begin{bmatrix} \vec{C}_1 \\ \vec{C}_2 \\ . \\ . \\ \vec{C}_{K-1} \\ \vec{C}_K \end{bmatrix}, \quad \vec{S} = \begin{bmatrix} \vec{S}_1' \\ \vec{S}_2 \\ . \\ . \\ \vec{S}_{K-1} \\ \vec{S}_K' \end{bmatrix} \tag{17}$$

$$V = \begin{bmatrix} V_1 & & & & & \\ & V_2 & & & & \\ & & . & & & \\ & & & . & & \\ & & & & . & \\ & & & & & V_K \end{bmatrix} \tag{18}$$

and

$$A = \begin{bmatrix} \bar{K}_1' & \bar{A}_1 & & & & \\ \bar{B}_2 & \bar{K}_2 & \bar{A}_2 & & & \\ & . & & & & \\ & & . & & & \\ & & & . & & \\ & & & \bar{B}_{K-1} & \bar{K}_{K-1} & \bar{A}_{K-1} \\ & & & & \bar{B}_K & \bar{K}_K' \end{bmatrix} . \tag{19}$$

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

$$V_k = I\,V, \tag{20}$$

358 125

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,[1] such techniques are limited in time-step size by spectral considerations. Instead, an operator method is used.[4,5]

By defining

$$\vec{X} = V\vec{C}, \quad \vec{g} = V\vec{S}, \quad \text{and} \quad B = AV^{-1} \tag{21}$$

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

$$\frac{d\vec{X}}{dt} = B\vec{X} + \vec{g}, \tag{22}$$

which has the solution[4,5]

$$\vec{X}(t) = e^{Bt} \ \vec{X}(0) + tD(Bt)\ \vec{g}, \tag{23}$$

where

$$D(Bt) = (Bt)^{-1} (e^{Bt} - I), \tag{24}$$

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

$$\vec{C}(t) = V^{-1} e^{Bt} \ V\vec{C}(0) + V^{-1} tD(Bt) \ V\vec{S}, \tag{25}$$

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.

358 126

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

$$C = \lambda \int_0^t C' e^{-\lambda t'} dt' + C' e^{\lambda t'}, \qquad (26)$$

where λ is the radioactive decay constant (1/s), 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 ($0 \leq x \leq \ell$) with a uniform initial disitribution and different saturated surface concentrations is[8]

$$C' = C_1 + (C_2 - C_1)\frac{x}{\ell} + \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{C_2\cos(n\pi) - C_1}{n}\sin\frac{n\pi x}{\ell}\exp\left(\frac{-Dn^2\pi^2 t}{\ell^2}\right)$$

$$+ \frac{4C_o}{\pi}\sum_{m=0}^{\infty}\frac{1}{2m+1}\sin\left(\frac{(2m+1)\pi x}{\ell}\right)\exp\left(\frac{-D(2m+1)^2\pi^2 t}{\ell^2}\right) \quad (27)$$

where C_o 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 specie 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 ($0 \le r \le a$) with a uniform initial distribution and a constant concentration at the outer radius is given analytically by[8]

TABLE II

DATA FOR VALIDATION TESTS

GEOMETRY	DIFFUSION COEFFICIENT ($cm^2\ s^{-1}$)	DECAY CONSTANT (s^{-1})	UNIFORM INITIAL CONCENTRATION (atoms/cm^3)	BOUNDARY CONDITIONS Left (atoms/cm^3)	Right (atoms/cm^3)	DIMENSIONS (cm)
Slab	7.234×10^{-6}	8.0225×10^{-7}	0.0	1.0×10^{10}	1.0×10^{10}	1 cm thick
Solid Cylinder	7.234×10^{-6}	8.0225×10^{-7}	0.0	REFLECTED	1.0×10^{10}	1 cm radius
Hollow Cylinder	7.234×10^{-6}	8.0225×10^{-7}	0.0	1.0×10^{10}	1.0×10^{10}	0.5 cm I.D., 2.0 cm o.d.
Solid Sphere	7.234×10^{-6}	8.0225×10^{-7}	0.0	REFLECTED	1.0×10^{10}	1 cm radius
Hollow Sphere	7.234×10^{-6}	8.0225×10^{-7}	0.0	1.0×10^{10}	1.0×10^{10}	0.5 cm I.D., 2.0 cm o.d.

358 128

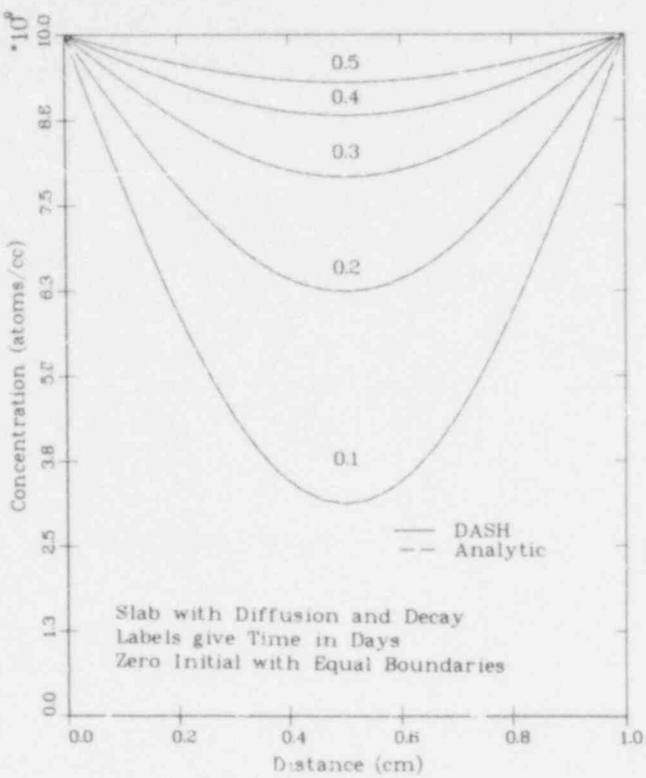

Fig. 2. Slab validation problem results.

$$C = C_1 + (C_0 - C_1)\,\frac{2}{a}\sum_{n=1}^{\infty}\frac{\exp(-D\alpha_n^2 t)\;J_0(r\alpha_n)}{\alpha_n J_1\,(a\,\alpha_n)}\;, \tag{28}$$

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 α_n's are roots of the Bessel function of the first kind of order zero,

$$J_0(a\,\alpha_n) = 0, \tag{29}$$

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. <u>Hollow Cylinder.</u> The analytic solution to the problem of flow through a cylinder wall ($a \le r \le b$) is[8]

$$C = \frac{C_1 \ln \frac{b}{r} + C_2 \ln \frac{r}{a}}{\ln (b/a)} + \pi C_o \sum_{n=1}^{\infty} \frac{J_o(a\alpha_n) U_o(r\alpha_n) \exp(-D\alpha_n^2 t)}{J_o(a\alpha_n) + J_o(b\alpha_n)}$$

$$+ \pi \sum_{n=1}^{\infty} \frac{\left\{C_2 J_o(a\alpha_n) - C_1 J_o(b\alpha_n)\right\} J_o(a\alpha_n) U_o(r\alpha_n)}{J_o^2(a\alpha_n) - J_o^2(b\alpha_n)} \exp(-D\alpha_n^2 t) \,, \qquad (30)$$

where C_o 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_o is given by

$$U_o(r\alpha_n) = J_o(r\alpha_n) Y_o(b\alpha_n) - J_o(b\alpha_n) Y_o(r\alpha_n). \qquad (31)$$

The values of α_n are the positive roots of

$$U_o(a\,\alpha_n) = 0, \qquad (32)$$

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.

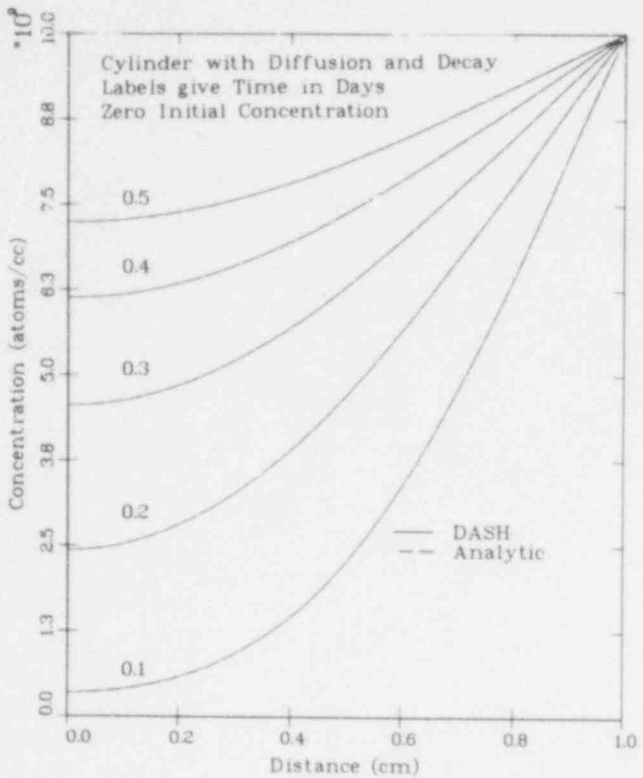

Fig. 3. Solid cylinder validation problem results.

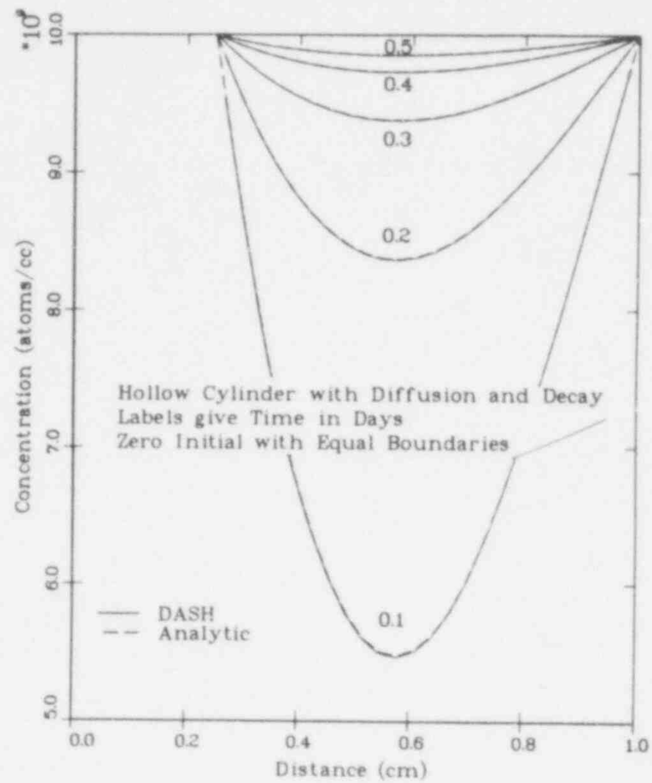

Fig. 4. Hollow cylinder validation problem results.

3. Spherical Problems

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

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

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.

358 131

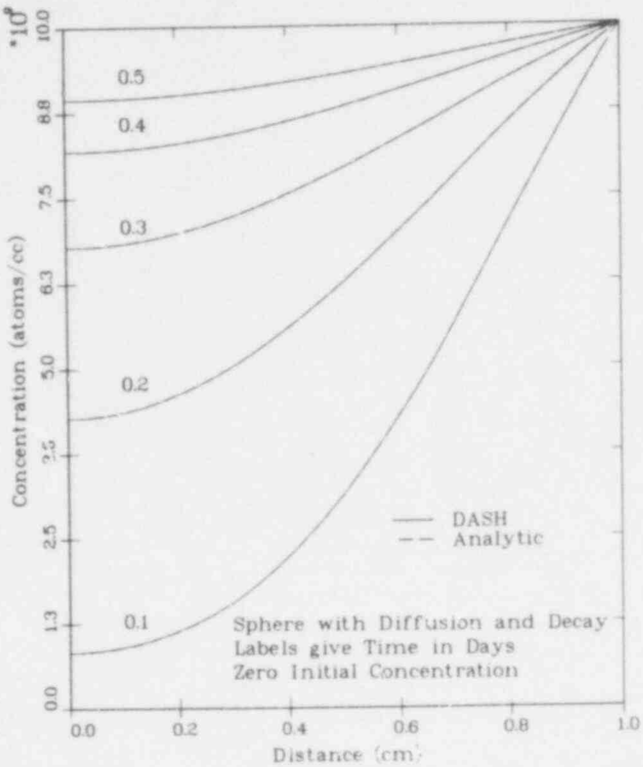

Fig. 5. Solid sphere validation problem results.

b. Hollow Sphere. The analytic solution for flow through a spherical wall ($a \le r \le b$) is[8]

$$C = \frac{aC_1}{r} + \frac{\left(bC_2 - aC_1\right)(r - a)}{r\,(b - a)}$$

$$+ \frac{2}{r\pi} \sum_{n=1}^{\infty} \frac{b(C_2 - C_o)\cos(n\pi) - a(C_1 - C_o)}{n} \sin \frac{n\pi(r - a)}{b - a} \exp\left(\frac{-Dn^2\pi^2 t}{(b - a)^2}\right), \tag{34}$$

where C_o 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.

358 132

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.[10] 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.

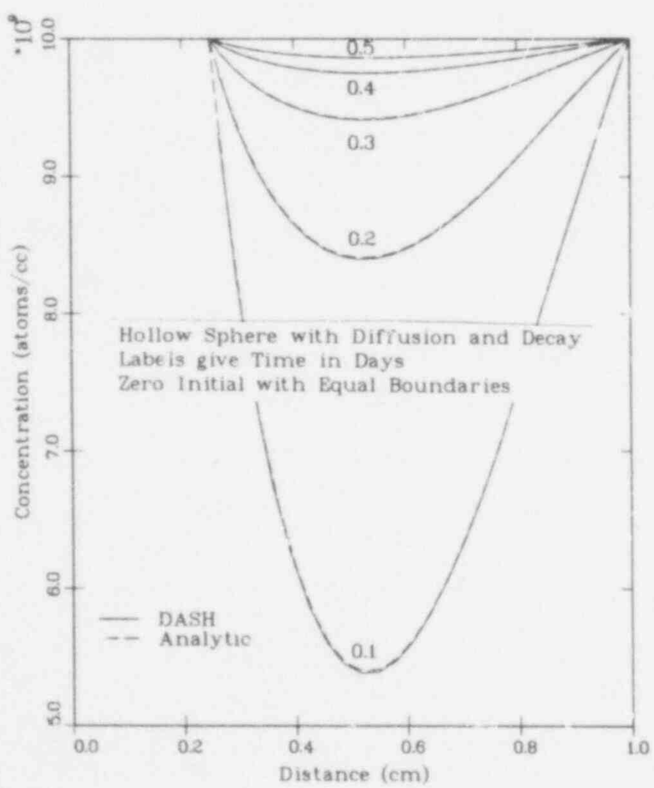

Fig. 6. Hollow sphere validation problem results.

358 133

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_∞ 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,[10] 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 III

TWO GROUP VALIDATION TEST DATA

	Group 1		Group 2	
	Material 1	Material 2	Material 1	Material 2
Diffusion Coefficient (cm)	1.13	1.13	0.16	0.16
Absorption Cross-Section (cm^{-1})	0.0419	0.0419	0.06	0.0197
Fission Cross-Section (cm^{-1})	0.0	0.0	0.040258	0.0

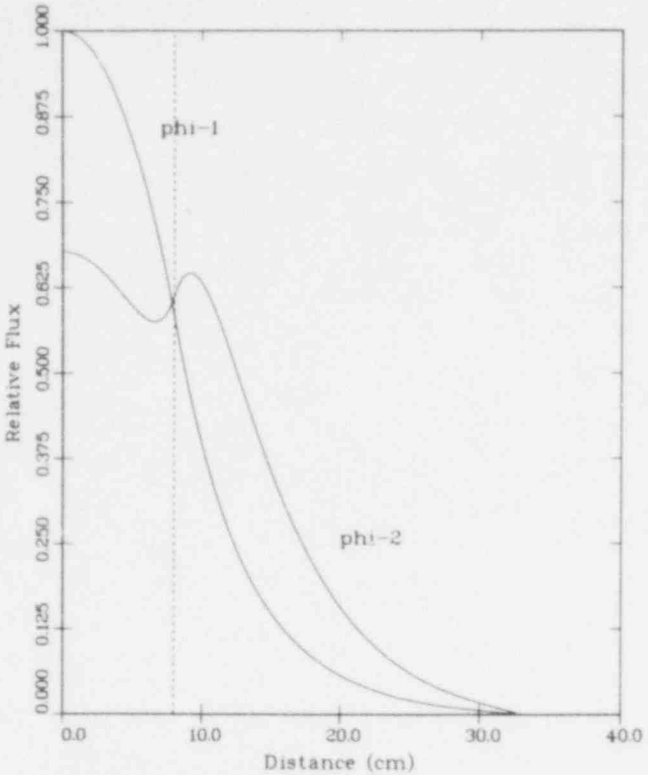

Fig. 7. Critical slab analytic results.

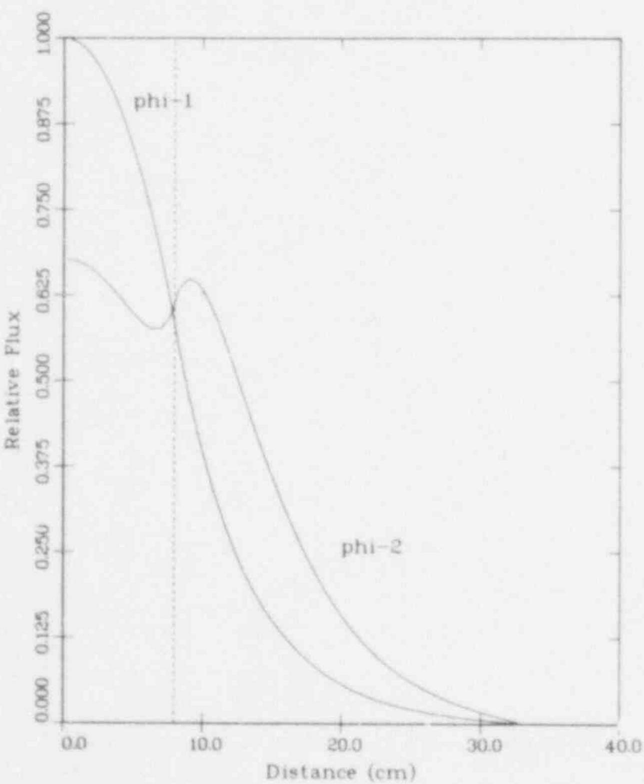

Fig. 8. Critical slab DASH results.

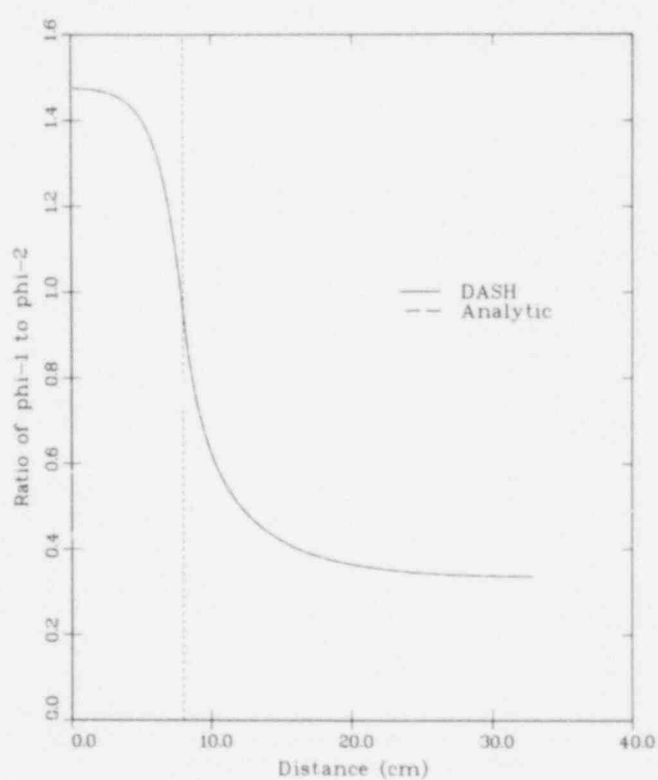

Fig. 9. Slab flux ratio comparison.

358 135

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.

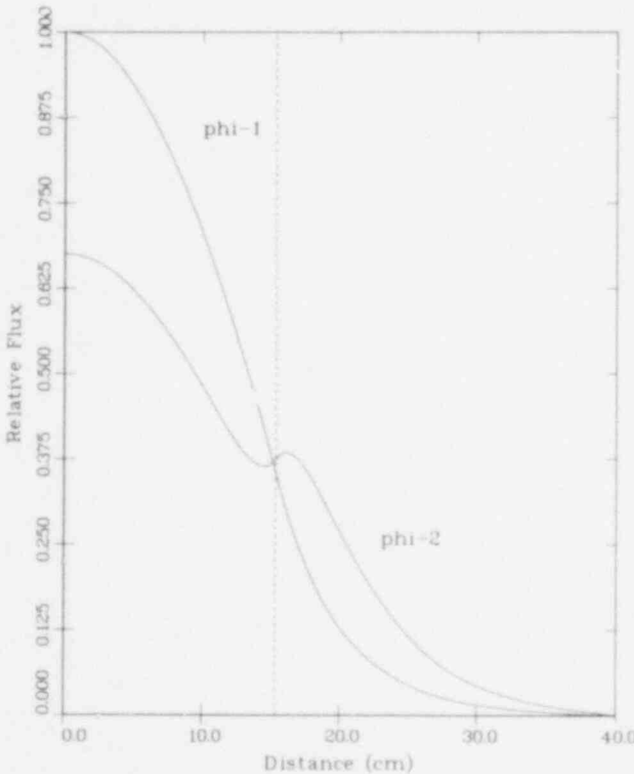

Fig. 10. Critical cylinder analytic results.

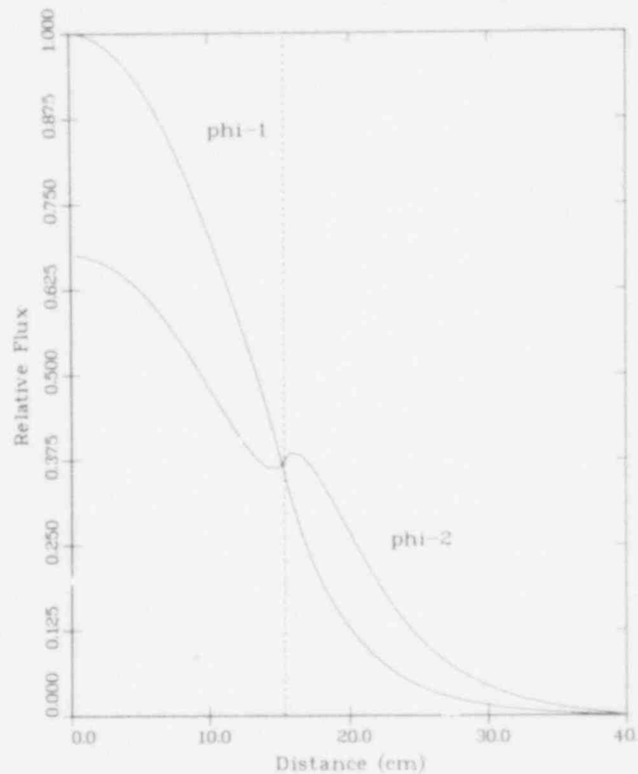

Fig. 11. Critical cylinder DASH results.

358 136

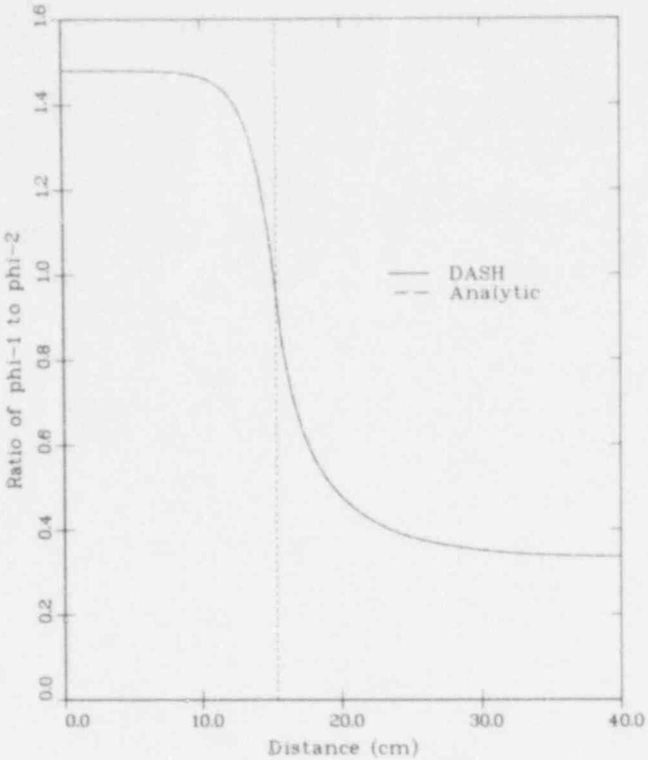

Fig. 12. Cylinder flux ratio comparison.

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.

358 137

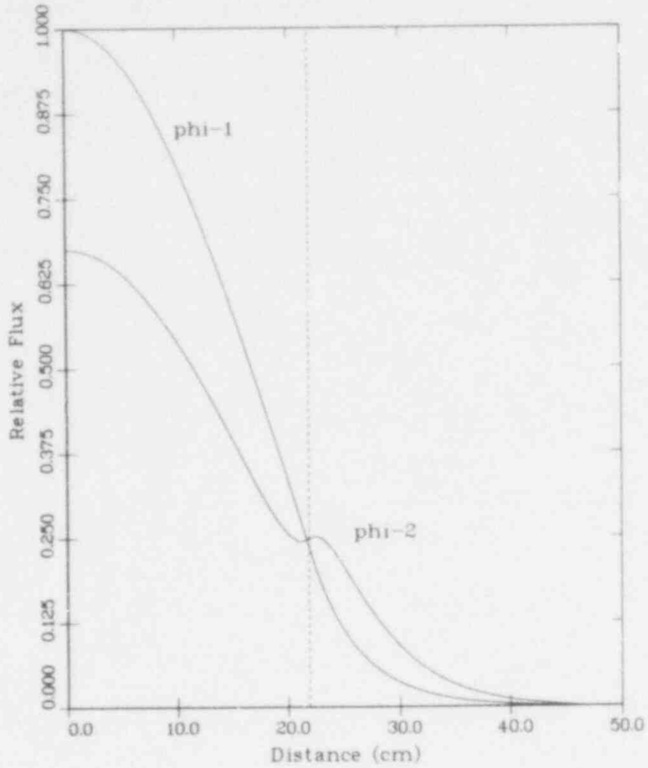

Fig. 13. Critical sphere analytic results.

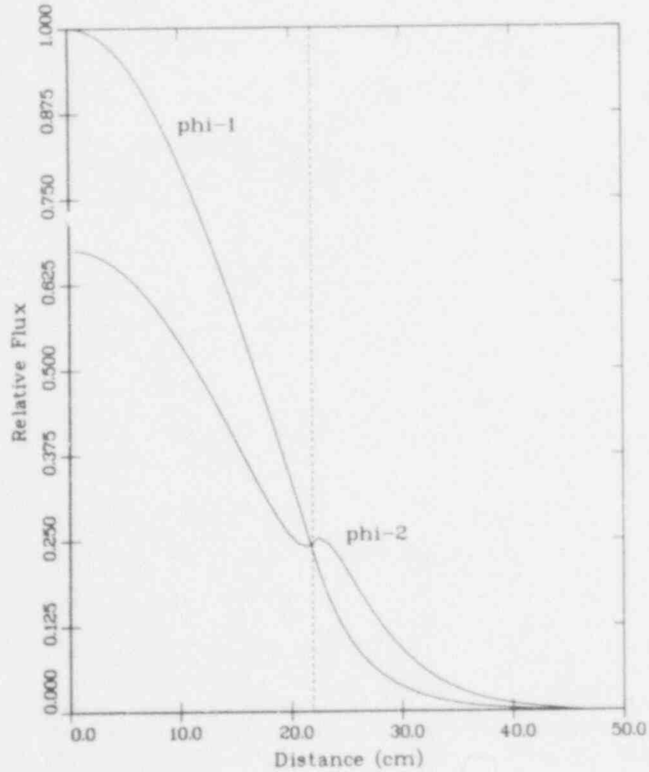

Fig. 14. Critical sphere DASH results.

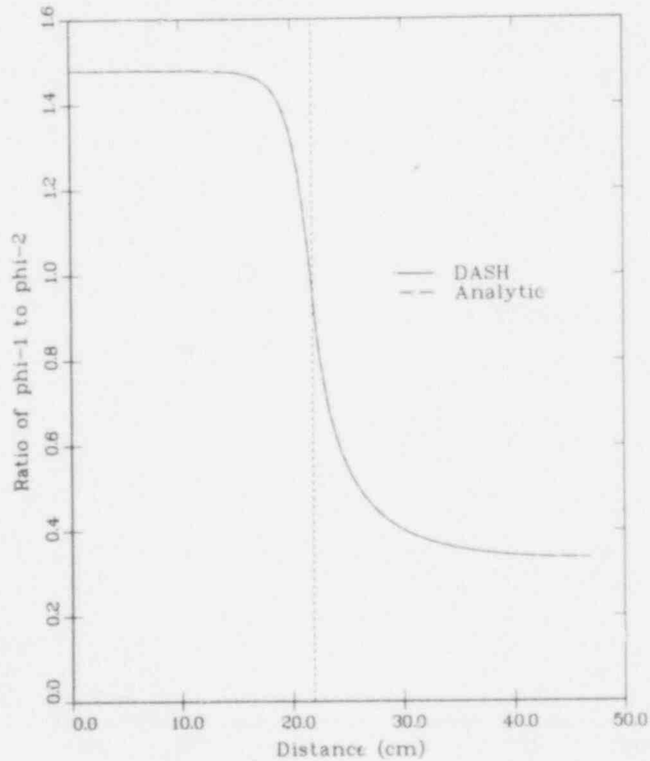

Fig. 15. Spherical flux ratio comparison.

358 138

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.

$$\varepsilon = kh^2 , \tag{35}$$

where

ε = inherent error
k = proportionality constant
h = 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.

$$\varepsilon = (kL^2) \frac{1}{n^2} . \tag{36}$$

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.1 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 εn^2 is approximately constant

358 139

TABLE IV

SPATIAL DIFFERENCING ERROR

n	ε	n^2	εn^2	Normalized ε
5	0.061320	25	1.53	1.000
10	0.016860	100	1.69	0.275
15	0.007778	225	1.75	0.127
20	0.004370	400	1.75	0.071
25	0.002820	625	1.76	0.046

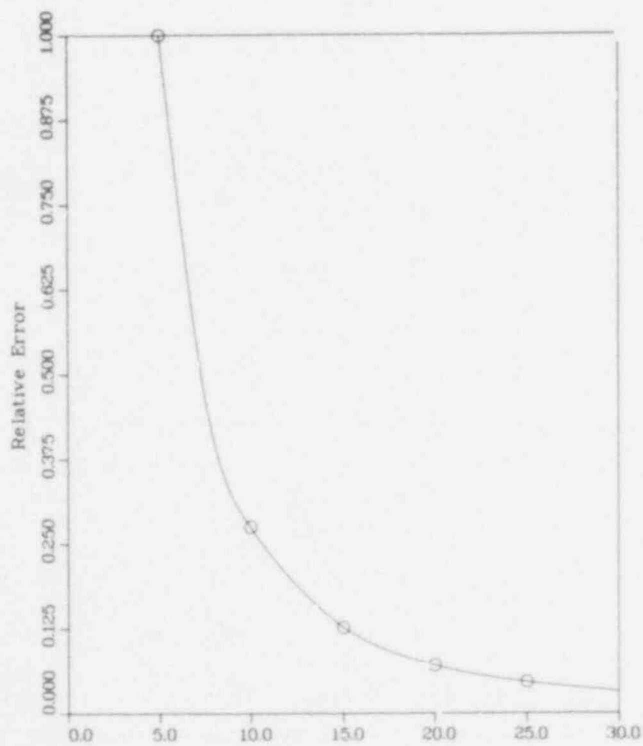

Fig. 16. Relative inherent differencing error.

358 140

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 Δr 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

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

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 7.3×10^9 atoms/($cm^3 \cdot s$) 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.

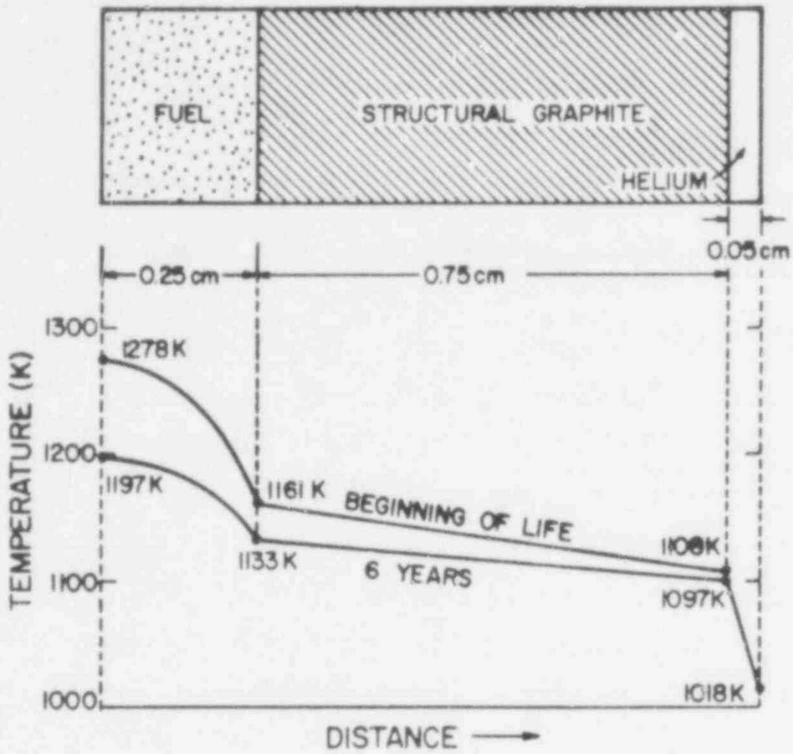

Fig. 17. Fuel-graphite-helium calculational model and beginning-of-life and six-year temperature profiles.

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 V

DATA FOR MASS-90 DECAY CHAIN

ISOTOPE	YIELD %	DECAY CONSTANT/s
^{90}Sr	5.77	7.844×10^{-10}
^{90}Y	5.77	2.994×10^{-6}
^{90}Zr	0.0	1.0×10^{-20}

$$-\log_{10} D = A + 1000\ B/T, \qquad (37)$$

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

Appel and Roos[11] assume that the concentration of ^{90}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} cm thick) at the boundary and adjusting the diffusion coefficient of the region introduced until the ratio of ^{90}Sr concentrations is 300. Except for this boundary region, the mesh spacing is taken as 0.05 cm throughout.

To compare with the work of Appel and Roos, the concentrations of ^{90}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

SPECIE	A	B
^{90}Sr	0.34	6.5
^{90}Y	0.74	14.2
^{90}Zr	1.19	22.8

TABLE VII

COMPARISON OF ^{90}Sr CONCENTRATIONS AT ONE YEAR

			DASH Results	
POSITION (cm)	MATERIAL	Results from Ref. 11 (atoms/cm^3)	Ref. 11 Coefficients	Ref. 12 Coefficients
0.125	Fuel	1.7×10^{18}	2.27×10^{17}	1.80×10^{17}
0.30	Graphite	3.0×10^{15}	5.41×10^{14}	5.68×10^{14}
0.50	Graphite	6.0×10^{14}	1.06×10^{14}	4.30×10^{14}
0.75	Graphite	1.0×10^{13}	4.62×10^{12}	2.23×10^{14}

It is apparent in looking at Table VII that the ^{90}Sr concentration in the fuel matrix as given by Appel and Roos is larger than that which a source of 7.3×10^9 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 7.3×10^{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 ($S_o = 7.3 \times 10^9$ atoms/cm^3.s) increases with time such that S_o is used for the first year, $2S_o$ for the second year, $3S_o$ for the third year, etc. Numerical results for ^{90}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}Y concentration profiles are shown in Fig. 19. Comparison of the amount of ^{90}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}Y does not diffuse significantly but decays into ^{90}Zr.

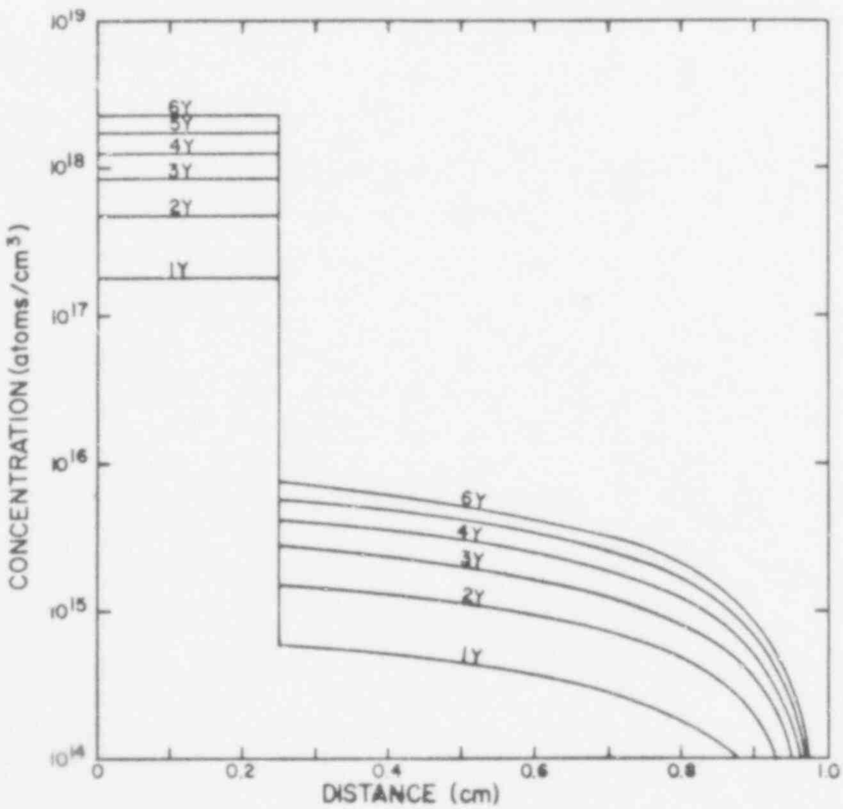

Fig. 18. ^{90}Sr concentration profiles.

TABLE VIII

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

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

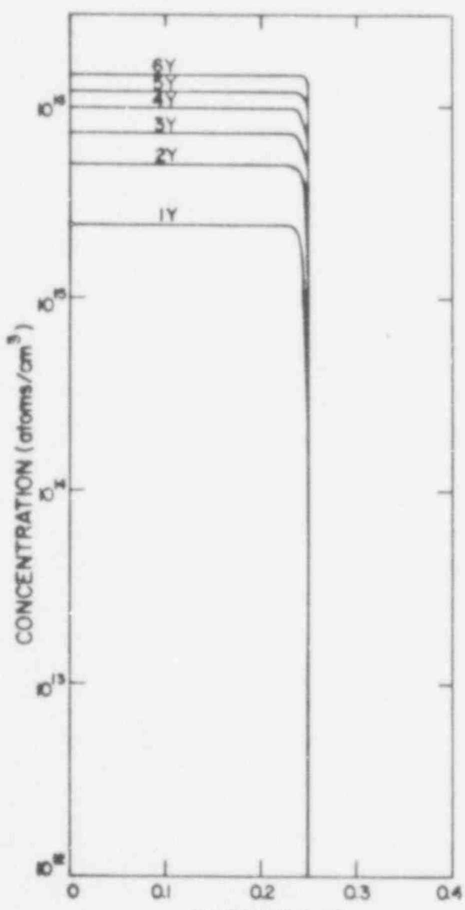

Fig. 19. ^{90}Y concentration profiles.

358 145

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 soubroutine 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 $\bar{B}_k$, Eq. (12), is evaluated for Eq. (9). The modified source, Eq. (13), due to the left boundary is also determined in this routine.

358 146

h. MAKLAM. The routine MAKLAM utilizes the nuclear data from INPA to construct the radioactive decay matrix, Eq. (2).

i. BIGEL. The routine BIGEL constructs all the matrices necessary for the matrix A, Eq. (19), except $\bar{K}_k'$. This determination is carried out on a cell-by-cell basis.

j. MAKEB. 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.

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

l. SOLVER. 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.

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

n. 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.

o. CONCPLT. 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

- GPLOT - 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.

358 148

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.

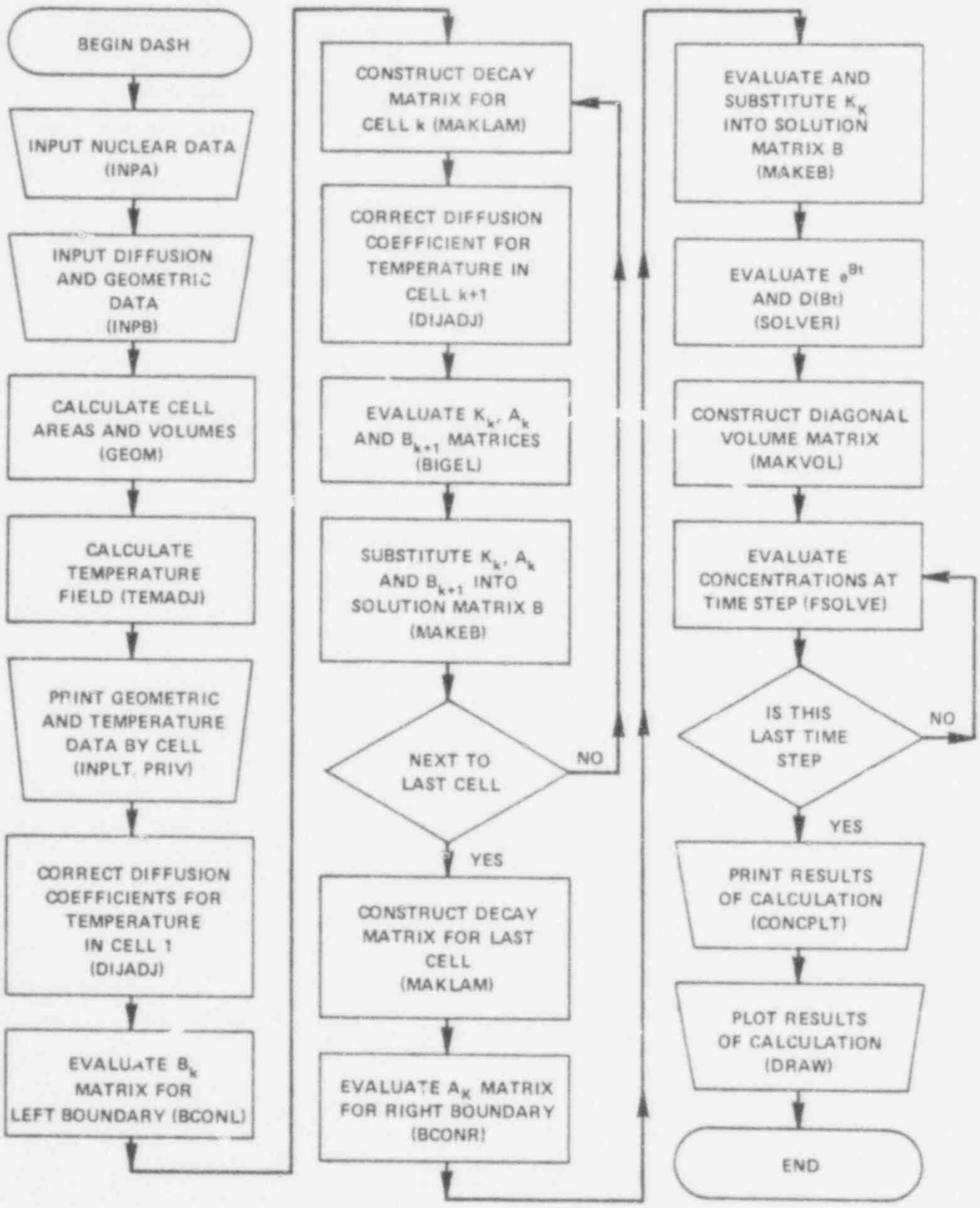

Fig. 20. DASH flow diagram.

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:

$$D = D_o e^{-\frac{A}{RT}} \tag{38}$$

= DIJO*EXP(- AIJO/(R*T))

for each element of DIJO and AIJO. Values must be supplied for each isotope in each material.

358 150

TABLE IX

DASH INPUT INSTRUCTIONS

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 -			BASIC NUCLIDE DATA
	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,3-4)-			BRANCHING RATIO
	1	BRV(IBR)	E12.5	BRANCHING RATIO
	-- BLANK CARD AFTER LAST SET OF CARDS 1 and 2 -- ISO = NUMBER OF NUCLIDE CARDS			
3				CROSS SECTION TITLE CARD
	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-			CROSS SECTION DATA MICROSCOPIC CROSS SECTIONS (cm^2)
	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

358 151

TABLE IX (cont)

CARD	WORD	SYMBOL	FORMAT	DESCRIPTION
--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 1/2 reflected/concentration specified
	4	NBCR	I4	Right boundary condition 1/2 reflected/concentration specified
	5	NTEMPS	I4	Number of entries for specifying temperature field
	6	IMATS	I4	Number of materials
	7	IGP	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				DIMENSIONS
	1	DIST(1)	SPECIAL	0.0
	2	DIST(2)	SPECIAL	First cell right boundary (cm)
	3	DIST(3)	SPECIAL	Second cell right boundary (cm)
	NCELLS+1	DIST(NCELLS+1)	SPECIAL	Last cell right boundary (cm)
8				ASSIGN MATERIALS
	1	MATS(1)	SPECIAL	Material ID for cell 1
	2	MATS(2)	SPECIAL	Material ID for cell 2
	.			
	.			
	.			
	NCELLS	MATS(NCELLS)	SPECIAL	Material ID for cell NCELLS
9				Dependent value for temperature field
	1	TEMPS(1)	SPECIAL	Temperature 1 (K)
	2	TEMPS(2)	SPECIAL	Temperature 2 (K)
	NTEMPS	TEMPS(NTEMPS)	SPECIAL	Temperature NTEMPS (K)
10				Independent value for temperature field
	1	TEMCOR(1)	SPECIAL	Coordinate of temperature 1 (cm)
	2	TEMCOR(2)	SPECIAL	Coordinate of temperature 2 (cm)
	NTEMPS	TEMCOR(NTEMPS)	SPECIAL	Coordinate of temperature NTEMPS (cm)

358 152

TABLE IX (cont)

CARD	WORD	SYMBOL	FORMAT	DESCRIPTION
11	- One set of cards 11 and 12 for each of IMATS materials -			Diffusion Matrix (cm^2/s)
	1	DIJO(1,1,1)	SPECIAL	Material 1 element (1,1)
	2	DIJO(2,1,1)	SPECIAL	Material 1 element (1,2)
	.	.	.	.
	ISO	DIJO(ISO,1,1)	SPECIAL	Material 1 element (1,ISO)
	ISO + 1	DIJO(1,2,1)	SPECIAL	Material 1 element (2,1)
	.	.	.	.
	ISO*ISO	DIJO(ISO,ISO,1)	SPECIAL	Material 1 element (ISO,ISO)
	ISO*ISO+1	DIJO(1,1,2)	SPECIAL	Material 2 element (1,1)
	.	.	.	.
	.	.	.	.
	ISO*ISO*N	DIJO(ISO,ISO,N)	SPECIAL	Material N element (ISO,ISO)
12	- One set of cards 11 and 12 for each of IMATS materials			Activation Energy Matrix(cal/mole)
	1	AIJS(1,1,1)	SPECIAL	Material 1 element (1,1)
	2	AIJS(2,1,1)	SPECIAL	Material 1 element (1,2)
	.	.	.	.
	.	.	.	.
	ISO	AIJS(ISO,1,1)	SPECIAL	Material 1 element (1,ISO)
	ISO + 1	AIJS(1,2,1)	SPECIAL	Material 1 element (2,1)
	.	.	.	.
	.	.	.	.
	ISO*ISO	AIJS(ISO,ISO,1)	SPECIAL	Material 1 element (ISO,ISO)
	ISO*ISO+1	AIJS(1,1,2)	SPECIAL	Material 2 element (1,1)
	.	.	.	.
	.	.	.	.
	ISO*ISO*N	AIJS(ISO,ISO,N)	SPECIAL	Material N element (ISO,ISO)
13	- One continuous set of card 13 for IGP groups - - Supply only if cross sections are present -			Fluxes
	1	PHI(1,N)	SPECIAL	Group N flux in cell 1(n/cm^2-s)
	2	PHI(2,N)	SPECIAL	Group N flux in cell 2(n/cm^2-s)
	.	.	.	.
	.	.	.	.
	NCELLS	PHI(NCELLS,N)	SPECIAL	Group N flux in cell NCELLS (n/cm^2-s)
	NCELLS + 1	PHI(1,N+1)	SPECIAL	Group N + 1 flux in cell 1 (n/cm^2-s)
	.	.	.	.
	.	.	.	.

358 153

CARD	WORD	SYMBOL	FORMAT	DESCRIPTION
14	- Supply only if NBCL = 2			Left boundary concentrations
	1	CONBOU(1,1)	SPECIAL	Specie 1 left boundary concentration (atoms/cc)
	2	CONBOU(2,1)	SPECIAL	Specie 2 left boundary concentration (atoms/cc)
	.	.	.	.
	ISO	CONBOU(ISO,1)	SPECIAL	Specie ISO left boundary concentration (atoms/cc)
15	- Supply only if NBCR = 2 -			Right boundary concentrations
	1	CONBOU(1,2)	SPECIAL	Specie 1 right boundary concentrations (atoms/cc)
	2	CONBOU(2.2)	SPECIAL	Specie 2 right boundary concentrations (atoms/cc)
	.	.	.	.
	ISO	CONBOU(ISO,2)	SPECIAL	Specie ISO right boundary concentrations (atoms/cc)
16				Initial concentration
	1	CONINT(1)	SPECIAL	Initial concentration cell 1 specie 1 (atoms/cc)
	2	CONINT(2)	SPECIAL	Initial concentration cell 1 specie 2 (atoms/cc)
	.	.	.	.
	ISO	CONINT(ISO)	SPECIAL	Initial concentration cell 1 specie ISO (atoms/cc)
	ISO + 1	CONINT(ISO+1)	SPECIAL	Initial concentration cell 2 specie 1
	.	.	.	.
	ISO*NCELLS	CONINT(ISO*NCELLS)	SPECIAL	Initial concentration cell NCELLS specie ISO
17	1	SOURCE(1)	SPECIAL	Source cell 1 specie 1(atoms/s)
	2	SOURCE(2)	SPECIAL	Source cell 1 specie 2(atoms/s)
	.	.	.	.
	ISO	SOURCE(ISO)	SPECIAL	Source cell 1 specie ISO(atoms/s)
	ISO+1	SOURCE(ISO+1)	SPECIAL	Source cell 2 specie 1(atoms/s)
	.	.	.	.
	ISO*NCELLS	SOURCE(ISO*NCELLS)	SPECIAL	Source cell NCELLS specie ISO (atoms/s)

358 154

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

Value of I1	Description
0/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 the data block. A 3 must follow last data word of all blocks.

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

DIFFUSANT	DECAY CONSTANT (s^{-1})	DIFFUSION COEFFICIENT (cm^2/s)			BOUNDARY CONCENTRATIONS	
		1	2	3	Left	Right
A	8.0225×10^{-7}	5.426×10^{-6}	1.266×10^{-5}	1.808×10^{-6}	1.0×10^{10}	0.0
B	1.6045×10^{-6}	2.713×10^{-6}	6.330×10^{-6}	9.042×10^{-7}	5.0×10^{9}	0.0

A

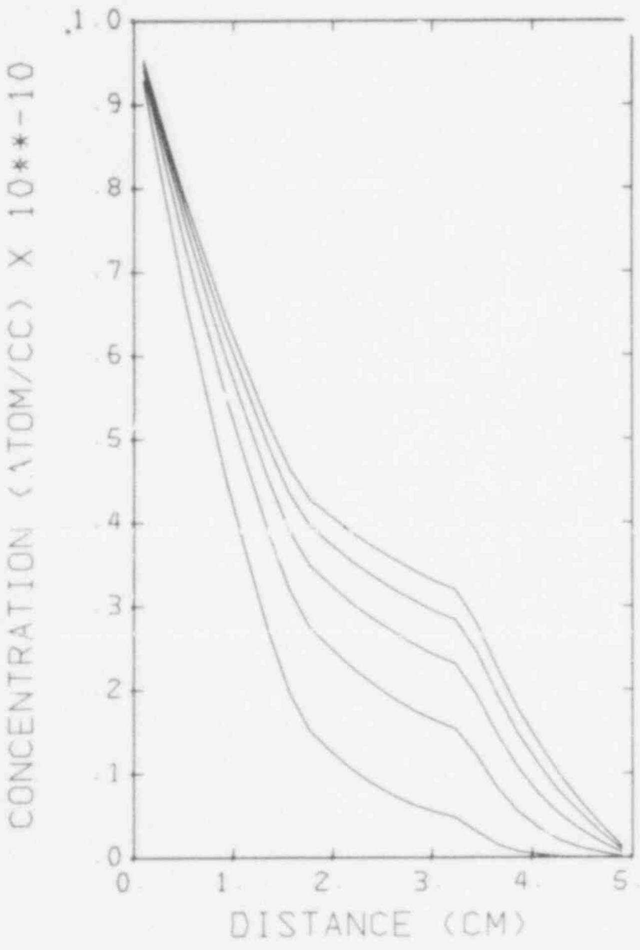

Fig. 21. Sample problem results for Diffusant A.

B

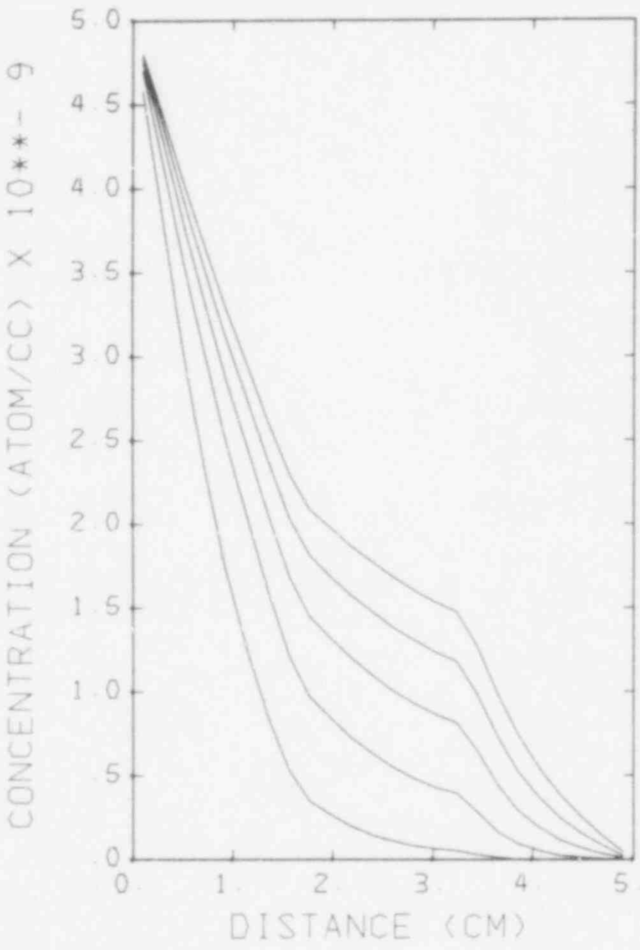

Fig. 22. Sample problem results for Diffusant B.

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

1. S. R. DeGroot and P. Mazur, Non-Equilibrium Thermodynamics (North Holland Publ. Co., Amsterdam, 1962).

2. S. Nakamura, Computational Methods in Engineering and Science (Wiley Interscience Publ., New York, 1977), pp. 157-164.

3. H. S. Carslaw and J. C. Jaeger, Conduction of Heat in Solids (The Clarendon Press, Oxford, 1959).

4. C. E. Lee, "The Calculation of Isotopic Mass and Energy Production by a Matrix Operator Method," Los Alamos Scientific Laboratory report LA-6483-MS (September 1976).

5. C. E. Lee, C. E. Apperson, and J. E. Foley, "Fission Product Release Calculations from a Reactor Containment Building," Nuclear Science and Engineering 64, 266-275 (1977).

6. Robert N. Thorn and Burton Wendroff, "Implicit Radiation Diffusion," Los Alamos Scientific Laboratory report LAMS-2960 (October 1963).

7. C. E. Lee and P. M. Stone, "Numerical Methods for Solving Linear Systems and Applications to Elliptic Difference Equations," Los Alamos Scientific Laboratory report LA-2314 (October 1959).

8. J. Crank, The Mathematics of Diffusion (The Clarendon Press, Oxford, 1975).

9. P. V. Danckwerts, "Absorption by Simultaneous Diffusion and Chemical Reaction into Particles of Various Shapes and into Falling Drops," Trans. Faraday Society 47, 1014-1023 (1951).

10. J. R. Lamarsh, Introduction to Nuclear Reactor Theory (Addison-Wesley Publ., Reading, Massachusetts, 1966).

11. J. Appel and B. Roos, "A Study of the Release of Radioactive Metallic Isotopes from High Temperature Gas-Cooled Reactors," Nuclear Science and Engineering 34, 201-213 (1968).

12. M. Schwartz, D. Sedgley, and M. Mendonca, "SORS: Computer Programs for Analyzing Fission Product Release from HTGR Cores during Transient Temperature Excursions," General Atomic Co. report GA-A12462, pp. 5-15 (April 1974).

13. L. M. Carruthers and C. E. Lee, "LARC-1: A Los Alamos Release Calculation Program for Fission Product Transport in HTGRs during the LOFC Accident," Los Alamos Scientific Laboratory report LA-NUREG-6563-MS (November 1976).

14. F. R. Gantmacher, The Theory of Matrices (Chelsea Publishing Co., New York, 1960), pp. 125-135; 185-190.

15. E. Bodewig, Matric Calculus (Interscience Publishers, New York, 1963).

APPENDIX A
MATRIX OPERATOR EVALUATION

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

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

If the matrix B is constant in the time interval (0, t), we may construct the matricant Ω_0^t (B), Eq. (A-2), using the Volterra method of the multiplicative integral.[4,14,15]

$$\Omega_0^t (B) = \exp \left[\int_0^t B(s)\, ds \right] = \exp [Bt] . \quad \text{(A-2)}$$

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

$$\vec{X}(t) = \Omega_0^t (B)\, \vec{X}(0) + \int_0^t dt' \, K(t, t')\, \vec{g}(t'), \quad \text{(A-3)}$$

where

$$K(t,t') = \Omega_0^t (B) \left[\Omega_0^{t'} (B) \right]^{-1}. \quad \text{(A-4)}$$

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

$$\vec{X}(t) = e^{Bt} \vec{X}(0) + e^{Bt} \int_0^t dt' \, e^{-Bt'} \vec{g}(t'). \quad \text{(A-5)}$$

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

$$\vec{X}(t) = e^{Bt} \vec{X}(0) + \vec{g}^{-1} (e^{Bt} - I)\, \vec{g}. \quad \text{(A-6)}$$

Defining the matrix operator D(C) by[4]

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

358 158

or

$$tD(Bt) = B^{-1} (e^{Bt} - I), \tag{A-8}$$

Eq. (A-6) becomes

$$\vec{X}(t) = \vec{X}(0) + tBD(Bt)\,\vec{X}(0) + tD(Bt)\,\vec{g} \tag{A-9}$$

$$= \vec{X}(0) + tD(Bt)\left[B\vec{X}(0) + \vec{g}\right].$$

Note that the matrix operator D(C) defined by

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

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

$$H = 2^{-p} C, \tag{A-11}$$

where p is determined by

$$||H|| < \frac{1}{2} \tag{A-12}$$

or[4,15]

$$p > \ln\left(\Sigma_{ij} |C_{ij}|^2\right) / (2 \ln 2). \tag{A-13}$$

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

$$D^M(H) \approx \sum_{n=0}^{M} \frac{H^n}{(n+1)!} \tag{A-14}$$

The value of M is chosen such that the excluded terms have an error less than ε,[4] or

$$\frac{(||H||)^{M + 1}}{(M + 2)!} < \frac{1}{2^{M + 1}(M + 2)!} < \varepsilon. \qquad \text{(A-15)}$$

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

$$D(2^{p + 1}H) = D(2^p H) \left[I + \tfrac{1}{2} (2^p H) D(2^p H) \right] . \qquad \text{(A-16)}$$

The recursion relation is readily proven by induction. Define

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

and

$$C = 2^p H. \qquad \text{(A-18)}$$

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

$$\begin{aligned} D(C) &= D(2H) = (2H)^{-1} (e^{2H} - I) \\ &= H^{-1} (e^H - I) \frac{(e^H + I)}{2} \\ &= D(H) \left[I + \tfrac{1}{2} H D(H) \right] . \end{aligned} \qquad \text{(A-19)}$$

Induction based on Eq. (A-19) yields

$$D(2^p H) = D(2^{p - 1}H) \left[I + \tfrac{1}{2} (2^{p - 1}H) D(2^{p - 1}H) \right] . \qquad \text{(A-20)}$$

Assume Eq. (A-20), which is true for p = 0 and 1, is true for p = n. Evaluate $D(2^{n + 1}H)$ as

$$D(2^{n+1}H) = (2^{n+1}H)^{-1}\left(e^{2^{n+1}H} - I\right)$$

$$= (2^{n}H)^{-1}\left(e^{2^{n}H} - I\right) \tfrac{1}{2}\left(e^{2^{n}H} + I\right) \qquad \text{(A-21)}$$

$$= D(2^{n}H)\left[I + \tfrac{1}{2}(2^{n}H)D(2^{n}H)\right] .$$

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)
C
C     DASH - A MULTICOMPONENT TIME DEPENDENT CONCENTRATION DIFFUSION
C     WITH RADIOACTIVE DECAY PROGRAM.
C
C     BY  COURTNEY E. APPERSON, JR.
C         LUCY M. CARRUTHERS
C         JUDITH F. SHINN
C         ENERGY DIVISION
C         LOS ALAMOS SCIENTIFIC LABORATORY
C         AND
C         CLARENCE E. LEE
C         TEXAS A AND M UNIVERSITY
C         MARCH 1979
C
C     THE PROGRAM DASH CALCULATES THE TRANSIENT CONCENTRATION OF
C     MULTIPLY DIFFUSING SPECIE WITH RADIOACTIVE DECAY USING FINITE
C     DIFFERENCE AND EXPONENTIAL OPERATOR TECHNIQUES.
C
C     THIS IS THE FOURTH VERSION OF DASH.  IT WAS CREATED
C     ON 15 MARCH 1979.
C
C     RECOGNIZING THAT GRAPHICS HARDWARE AND SOFTWARE ARE USUALLY
C     UNIQUE TO A PARTICULAR INSTALLATION THE GRAPHICS PACKAGE IN
C     DASH CAN BE READILY DELETED WITHOUT EFFECTING THE REMAINDER
C     OF THE CODE.
C
C     INPUT INSTRUCTIONS FOR THE CODE ARE--
C
C     CARD  WORD        SYMBOL      FORMAT           DESCRIPTION
C
C      0                                     PRINT OPTIONS
C
C            1          NPRINT        I4     0/1 LINE PRINTER/TERMINAL
C            2          NPLOT         I4     0/1   NO PLOT/PLOT
```

```
C
C   1  --ONE CARD PER NUCLIDE--              BASIC NUCLIDE DATA
C
C          1      NANMAT(I,1)        A7      NUCLIDE NAME
C          2      NANMAT(I,2)        I4      ID NUMBER
C          3      NANMAT(I,3)        I4      DECAY PARENT 1
C          4      NANMAT(I,4)        I4      DECAY PARENT 2
C          5      NANMAT(I,5)        I4      CAPTURE PARENT 1
C          6      NANMAT(I,6)        I4      CAPTURE PARENT 2
C          7      NANMAT(I,7)        I4      N-2N PARENT
C          8      NANMAT(I,8)        I4      N-ALPHA PARENT
C          9      NANMAT(I,9)        I4      N-P PARENT
C         10      ANMAT(I,1)       E12.5     DECAY CONSTANT (1/S)
C
C   2  --ONE CARD PER BRANCH FOR EACH NEGATIVE VALUE
C           OF NANMAT(I,3-9)--
C
C          1       BRV(IBR)        E12.5     BRANCHING RATIO
C
C      --BLANK CARD AFTER LAST SET OF CARDS 1 AND 2--
C
C   3                                        CROSS SECTION TITLE CARD
C
C          1      NXSEC(II,1)        A6      TITLE 1
C          2      NXSEC(II,2)        A6      TITLE 2
C          3      NXSEC(II,3)        I4      NUMBER OF GROUPS
C          4      NXSEC(II,4)        I4      NUCLIDE ID
C
C   4  --NXSEC(II,3) CARDS--                 CROSS SECTION DATA
C
C          1     XSEC(II,KX,1)     E12.5     SIGMA  N-GAMMA
C          2     XSEC(II,KX,2)     E12.5     SIGMA  N-2N
C          3     XSEC(II,KX,3)     E12.5     SIGMA  N-ALPHA
C          4     XSEC(II,KX,4)     E12.5     SIGMA  N-P
C
C      --BLANK CARD AFTER LAST SET OF CARDS 3 AND 4--
C
C   5                                        PROBLEM RELATED DATA
C
C          1         NCELLS          I4      NUMBER OF CELLS IN PROBLEM
C          2         NGEOM           I4      1/2/3 SLAB/CYLINDER/SPHERE
C          3         NBCL            I4      LEFT BOUNDARY CONDITION
C                                            1/2 REFLECTED/CONCENTRATION
C                                                SPECIFIED
C          4         NBCR            I4      RIGHT BOUNDARY CONDITION
C                                            1/2 REFLECTED/CONCENTRATION
C                                                SPECIFIED
C          5         NTEMPS          I4      NUMBER OF ENTRIES FOR
C                                            SPECIFYING TEMPERATURE FIELD
C          6         IMATS           I4      NUMBER OF MATERIALS
C          7         IGP             I4      NUMBER OF NEUTRON ENERGY
C                                            GROUPS
C
C   6                                        TIME STEP DATA
C
C          1         TINT          E12.6     INITIAL TIME (DAYS)
C          2         TINC          E12.6     NUMBER OF TIME STEPS
C          3         TIMAX         E12.6     TIME AT END OF PROBLEM
C                                            (DAYS)
C
C   7                                        GEOMETRY (CM)
C
C          1         DIST(1)      SPECIAL    0.0
C          2         DIST(2)      SPECIAL    FIRST CELL RIGHT BOUNDARY
C          3         DIST(3)      SPECIAL    SECOND CELL RIGHT BOUNDARY
C          .            .            .          .
C   NCELLS+1  DIST(NCELLS+1)      SPECIAL    LAST CELL RIGHT BOUNDARY
C
```

```
C
C      8                                       ASSIGN MATERIALS
C
C           1         MATS(1)      SPECIAL    MATERIAL ID FOR CELL 1
C           2         MATS(2)      SPECIAL    MATERIAL ID FOR CELL 2
C           .            .            .                 .
C        NCELLS     MATS(NCELLS)  SPECIAL    MATERIAL ID FOR CELL NCELLS
C
C      9                                       DEPENDENT VALUE FOR
C                                              TEMPERATURE FIELD (K)
C
C           1         TEMPS(1)     SPECIAL    TEMPERATURE 1
C           2         TEMPS(2)     SPECIAL    TEMPERATURE 2
C           .            .            .             .
C        NTEMPS     TEMPS(NTEMPS) SPECIAL    TEMPERATURE NTEMPS
C
C     10                                       INDEPENDENT VALUE FOR
C                                              TEMPERATURE FIELD (CM)
C
C           1         TEMCOR(1)    SPECIAL    COORDINATE OF TEMPERATURE 1
C           2         TEMCOR(2)    SPECIAL    COORDINATE OF TEMPERATURE 2
C           .            .            .             .
C       NTEMPS    TEMCOR(NTEMPS)   SPECIAL    COORDINATE OF TEMPERATURE
C                                              NTEMPS
C
C     11   --ONE SET OF CARDS 11 AND 12 FOR EACH OF IMATS MATERIALS--
C
C                                              DIFFUSION MATRIX (CM**2/S)
C           1        DIJO(1,1,N)   SPECIAL    MATERIAL N ELEMENT(1,1)
C           2        DIJO(2,1,N)   SPECIAL    MATERIAL N ELEMENT(1,2)
C           .            .            .             .
C          ISO      DIJO(ISO,1,N)  SPECIAL    MATERIAL N ELEMENT(1,ISO)
C         ISO+1      DIJO(1,2,N)   SPECIAL    MATERIAL N ELEMENT(2,1)
C           .            .            .             .
C        ISO*ISO DIJO(ISO,ISO,N)SPECIAL       MATERIAL N ELEMENT(ISO,ISO)
C
C     12   --ONE SET OF CARDS 11 AND 12 FOR EACH OF IMATS MATERIALS--
C
C                                              ACTIVATION ENERGY MATRIX
C                                              (CAL/MOLE)
C           1        AIJS(1,1,N)   SPECIAL    MATERIAL N ELEMENT(1,1)
C           2        AIJS(2,1,N)   SPECIAL    MATERIAL N ELEMENT(1,2)
C           .            .            .             .
C          ISO      AIJS(ISO,1,N)  SPECIAL    MATERIAL N ELEMENT(1,ISO)
C         ISO+1      AIJS(1,2,N)   SPECIAL    MATERIAL N ELEMENT(2,1)
C           .            .            .             .
C        ISO*ISO AIJS(ISO,ISO,N)SPECIAL       MATERIAL N ELEMENT(ISO,ISO)
C
C     13   --ONE CONTINUOUS SET OF CARD 13 FOR IGP GROUPS--
C          --SUPPLY ONLY IF CROSS SECTIONS ARE PRESENT--
C
C                                              FLUXES (NEUTRONS/CM**2-S)
C           1        PHI(1,N)      SPECIAL    GROUP N FLUX IN CELL 1
C           2        PHI(2,N)      SPECIAL    GROUP N FLUX IN CELL 2
C           .            .            .             .
C        NCELLS    PHI(NCELLS,N)   SPECIAL    GROUP N FLUX IN CELL NCELLS
C       NCELLS+1    PHI(1,N+1)     SPECIAL    GROUP N+1 FLUX IN CELL 1
C           .            .            .             .
C
C     14   --SUPPLY ONLY IF NBCL = 2--        LEFT BOUNDARY CONCENTRATIONS
C                                              (ATOMS/CC)
C
C           1       CONBOU(1,1)    SPECIAL    SPECIE 1 LEFT BOUNDARY CONC
C           2       CONBOU(2,1)    SPECIAL    SPECIE 2 LEFT BOUNDARY CONC
C           .            .            .             .
C          ISO     CONBOU(ISO,1)   SPECIAL    SPECIE ISO LEFT BOUNDARY
C                                              CONC
C
```

```
C     15  --SUPPLY ONLY IF NBCR = 2--      RIGHT BOUNDARY CONCENTRA-
C                                          TIONS
C                                          (ATOMS/CC)
C
C         1     CONBOU(1,2)     SPECIAL    SPECIE 1 RIGHT BOUNDARY CONC
C         2     CONBOU(2,2)     SPECIAL    SPECIE 2 RIGHT BOUNDARY CONC
C         .          .             .
C        ISO    CONBOU(ISO,2)   SPECIAL    SPECIE ISO RIGHT BOUNDARY
C                                          CONC
C
C     16                                   INITIAL CONCENTRATION
C                                          (ATOMS/CC)
C
C         1     CONINT(1)       SPECIAL    INITIAL CONC CELL 1 SPECIE 1
C         2     CONINT(2)       SPECIAL    INITIAL CONC CELL 1 SPECIE 2
C         .          .             .
C        ISO    CONINT(ISO)     SPECIAL    INITIAL CONC CELL 1 SPECIE
C                                          ISO
C      ISO+1    CONINT(ISO+1)   SPECIAL    INITIAL CONC CELL 2 SPECIE 1
C         .          .             .
C     ISO*NCELLS                SPECIAL    INITIAL CONC CELL NCELLS
C            CONINT(ISO*NCELLS)            SPECIE ISO
C
C     17                                   SOURCE
C                                          (ATOMS/SEC)
C
C         1     SOURCE(1)       SPECIAL    SOURCE CELL  1 SPECIE 1
C         2     SOURCE(2)       SPECIAL    SOURCE CELL  1 SPECIE 2
C         .          .             .
C        ISO    SOURCE(ISO)     SPECIAL    SOURCE CELL  1 SPECIE ISO
C      ISO+1    SOURCE(ISO+1)   SPECIAL    SOURCE CELL  2 SPECIE 1
C         .          .             .
C      ISO*NCELLS               SPECIAL    SOURCE CELL NCELLS SPECIE IS
C               SOURCE(ISO*NCELLS)
C
C     SPECIAL FORMATS
C
C     THERE ARE TWO SPECIAL READ FORMATS.  ONE IS FOR INTEGER DATA
C     6(I1,I2,I9) AND ONE IS FOR FLOATING POINT DATA 6(I1,I2,E9.3).
C     IN EACH WORD OF BOTH THESE FORMATS, THE FIRST INTEGER FIELD,I1,
C     DESIGNATES THE OPTIONS LISTED BELOW.  THE SECOND INTEGER FIELD,
C     I2, CONTROLS THE EXECUTION OF THE OPTION, AND THE REMAINDER OF
C     THE FIELD, I9 OR E9.3, IS FOR THE INPUT DATA.  ALL DATA BLOCKS
C     READ WITH THESE FORMATS MUST BE ENDED WITH A 3 IN THE I1 FIELD
C     AFTER THE LAST WORD OF THE BLOCK.
C
C     --OPTIONS FOR SPECIAL READ FORMATS--
C
C     VALUE OF I1          NATURE OF OPTION
C
C     0 OR BLANK           NO ACTION
C          1               REPEAT DATA WORD IN 9 FIELD NUMBER OF TIMES
C                          INDICATED IN I2 FIELD.
C          2               PLACE NUMBER OF LINEAR INTERPOLANTS INDICATED
C                          IN I2 FIELD BETWEEN DATA WORD IN 9 FIELD AND
C                          DATA WORD IN NEXT 9 FIELD.  NOT ALLOWED FOR
C                          INTEGERS.
C          3               TERMINATE READING OF DATA BLOCK.  A 3 MUST
C                          FOLLOW LAST DATA WORD OF ALL BLOCKS.
C
C     DIMENSION NANMAT(NISO1,9),ANMAT(NISO1,2),BRV(NBR),NXSEC(NXSP,4)
C    1,XSEC(NXS,NGP,4),NPP(NCELL)
C     DIMENSION DIST(NCELL1),MATS(NCELL),TEMPS(NTEM),TEMCOR(NTEM),
C    1 DIJO(NISO,NISO,NMATS),AIJS(NISO,NISO,NMATS),DUM1(NTEM),DUM2(NTEM)
C     DIMENSION DUM11(NN),DUM22(NN),ALAM(NISO,NISO),BB(NISO,NISO)
C     WHERE NN IS THE GREATER OF NISO*NISO AND NGP*NCELL
C     DIMENSION DELR(NCELL),AREA(NCELL),VOL(NCELL),RBAR(NCELL),
C    1 PHI(NCELL,NGP),W(NTEM),DUM3(NTEM),DIFFK(NISO,NISO),
C    2 DIFFK1(NISO,NISO),DAPLOT(NISO,NCELL,NTIME)
C     COMMON /SOLS/ BIGB(MM,MM),B(MM,MM),C(MM,MM),D(MM,MM),E(MM,MM),
C    1 F(MM,MM)
```


358 164

```
C      DIMENSION SOURCE(MM),DIFDUM1(NISO,NISO),DIFDUM2(NISO,NISO),
C     1 CONBOU(NISO,2),IPVT(NISO),AK(NISO,NISO),BK(NISO,NISO),
C     2 CKK(NISO,NISO),CONINT(MM),CONCEN(MM),DUM33(MM)
C      WHERE MM=NISO*NCELL
       DIMENSION NANMAT(6,9), ANMAT(6,2), BRV(10), NXSEC(11,4), XSEC(10,4
      1 ,4), NPP(25)
       DIMENSION DIST(25), 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 /SOLS/ BIGB(125,125), B(125,125), C(125,125), D(125,125), E
      1 (125,125), F(125,125)
       DIMENSION SOURCE(125), DIFDUM1(5,5), DIFDUM2(5,5), CONBOU(5,2),
      1 IPVT(5), AK(5,5), BK(5,5), CKK(5,5), CONINT(125), CONCEN(125),
      2 DUM33(125)
       LEVEL 2, BIGB, B, C, D, E, F
       COMMON /IO/ NINP, NOUT, IER, NPRINT
       COMMON /NUCDAT/ ISO, IBR, IXS, IGP
       COMMON /TIMES/ TINT, TINC, TIMAX
C      TAPE5 IS INPUT UNIT
       NINP=5
C      TAPE6 IS OUTPUT UNIT
       NOUT=6
       READ (NINP,100) NPRINT,NPLOT
       NISO=5
       NISO1=NISO+1
       NN=100
C      NN IS THE GREATER OF NISO*NISO AND NGP*NCELL
       NBR=10
       NXS=10
       NXSP=NXS+1
       NGP=4
       NCELL=25
       NCELL1=NCELL+1
       MM=NISO*NCELL
       NTEM=25
C      NTEM MUST BE GREATER THAN OR EQUAL TO NCELL
       NMATS=5
       NTIME=11
       CALL INPA (NANMAT,ANMAT,BRV,NXSEC,XSEC,NGP,NXS,NBR,NISO1,NXSP)
       CALL INPB (NCELLS,NGEOM,NBCR,NBCL,NTEMPS,IMATS,DIST,MATS,PHI,TEMPS
      1 ,TEMCOR,DIJO,AIJS,NCELL1,NCELL,NTEM,NISO,NMATS,NGP,DUM11,DUM22,NN
      2 ,CONBOU,CONINT,SOURCE,MM)
       N=NCELLS+1
       CALL GEOM (NGEOM,DIST,DELR,AREA,VOL,RBAR,N,NCELLS)
       CALL INPLT (NCELL,NCELL1,NBCL,NBCR,MATS,DIST,NGEOM,NCELLS)
C      CALL PRIV(DIST,N,NCELL1,10H DISTANCES,10H           )
C      CALL PRIV(DELR,NCELLS,NCELL1,10H DELR      ,10H          )
C      CALL PRIV(AREA,NCELLS,NCELL1,10H AREA      ,10H          )
C      CALL PRIV(VOL,NCELLS,NCELL1,10H VOLUME    ,10H          )
       CALL PRIV (RBAR,NCELLS,NCELL1,10H RBAR      ,10H          )
       CALL TEMADJ (TEMPS,TEMCOR,NTEMPS,RBAR,NCELLS,W,DUM1,DUM2,DUM3,NTEM
      1 )
       CALL PRIV (TEMPS,NCELLS,NTEM,10H MESH TEMP,10HERATURES  )
C      IF(IXS.EQ.0) GO TO 5
C      CALL PRIM(PHI,NCELLS,IGP,NCELL,NGP,10H FLUXES    ,10H          )
C    5 CONTINUE
       DO 20 I=1,MM
       DO 10 J=1,MM
       BIGB(I,J)=0.0
    10 CONTINUE
    20 CONTINUE
       II=1
C
C      MAKE ADJUSTMENTS FOR HOLLOW CYLINDER AND SPHERE
C
       IF (NBCL.EQ.2.AND.NGEOM.GT.1) II=2
       MAT=MATS(II)
       CALL DIJADJ (DIJO,AIJS,TEMPS,DIFFK,NISO,NMATS,NCELL,ISO,II,MAT)
C      CALL PRIM(DIFFK,ISO,ISO,ISO,ISO,10H DIFFK     ,10H          )
```


358 165

```
      NIC=ISO*(NCELLS+1-II)
      CALL BCONL (BK,AREA,DIFFK,DELR,SOURCE,CONBOU,NCELLS,ISO,NBCL,NGEOM
     1 ,NISO,NIC,DUM11,DUM33)
C     CALL PRIM(BK,ISO,ISO,ISO,ISO,10H BK        ,10H          )
      NM1=NCELLS-1
      ICOL=1
      DO 30 I=II,NM1
      CALL MAKLAM (NANMAT,ANMAT,XSEC,NXSEC,PHI,BRV,ALAM,BB,NISO1,NXS
     1 ,NXSP,NBR,NGP,ISO,IXS,IGP,NCELL,I)
C     CALL PRIM(ALAM,ISO,ISO,ISO,ISO,10H LAMBDA    ,10H          )
      K=I+1
      MAT=MATS(K)
      CALL DIJADJ (DIJO,AIJS,TEMPS,DIFFK1,NISO,NMATS,NCELL,ISO,K,MAT)
C     CALL PRIM(DIFFK1,ISO,ISO,ISO,ISO,10H DIFFK1    ,10H          )
      CALL BIGEL (DIFFK,DIFFK1,DIFDUM1,DIFDUM2,ALAM,AK,BK,CKK,ISO,AREA
     1 ,VOL,DELR,NCELLS,IPVT,DUM11,I)
C     CALL PRIM(CKK,ISO,ISO,ISO,ISO,10H KK        ,10H          )
C     CALL PRIM(AK,ISO,ISO,ISO,ISO,10H AK        ,10H          )
C     CALL PRIM(BK,ISO,ISO,ISO,ISO,10H BK        ,10H          )
      CALL MATMOV (ISO,ISO,DIFFK1,ISO,DIFFK,ISO)
      CALL MAKEB (BIGB,AK,BK,CKK,NIC,ISO,VOL,NCELLS,ICOL,II)
   30 CONTINUE
      CALL MAKLAM (NANMAT,ANMAT,XSEC,NXSEC,PHI,BRV,ALAM,BB,NISO1,NXS
     1 ,NXSP,NBR,NGP,ISO,IXS,IGP,NCELL,NCELLS)
C     CALL PRIM(ALAM,ISO,ISO,ISO,ISO,10H LAMBDA    ,10H          )
      CALL BCONR (AK,AREA,DIFFK,DELR,SOURCE,CONBOU,NCELLS,ISO,NBCR,NISO
     1 ,NIC,DUM11,DUM22,BK,CKK,ALAM,VOL,DIFDUM1)
C     CALL PRIM(CKK,ISO,ISO,ISO,ISO,10H KK        ,10H          )
C     CALL PRIM(AK,ISO,ISO,ISO,ISO,10H AK        ,10H          )
C     CALL PRIV(SOURCE,NIC,NIC,10H SOURCE    ,10H          )
      CALL MAKEBL (BIGB,AK,BK,CKK,NIC,ISO,VOL,NCELLS,ICOL,II)
C     CALL PRIMES(BIGB,NIC,NIC,NIC,NIC,10H BIGB      ,10H          )
      DTIME=(TIMAX/TINC)*24.*3600.
      NSTEP=TINC
      CALL SOLVER (BIGB,B,C,D,E,F,DTIME,NIC,NIC)
C     CALL PRIMES(E,NIC,NIC,NIC,NIC,10H E         ,10H          )
C     CALL PRIMES(D,NIC,NIC,NIC,NIC,10H D         ,10H          )
      NP=NCELLS-II+1
      DO 50 J=1,NP
      DO 40 I=1,ISO
      I1=ISO*(J-1)+I
      DAPLOT(I,J,1)=CONINT(I1)
   40 CONTINUE
   50 CONTINUE
      CALL MAKVOL (B,NIC,VOL,NCELLS,II,ISO)
C     CALL PRIMES(B,NIC,NIC,NIC,NIC,10H VOLMAT    ,10H          )
      DO 90 I=1,NSTEP
      CALL FSOLVE (B,C,D,E,DUM33,DTIME,NIC,CONINT,SOURCE,CONCEN)
      DO 70 J=1,NP
      DO 60 JI=1,ISO
      I1=ISO*(J-1)+JI
      DAPLOT(JI,J,I+1)=CONCEN(I1)
   60 CONTINUE
   70 CONTINUE
      DO 80 J=1,NIC
      CONINT(J)=CONCEN(J)
   80 CONTINUE
   90 CONTINUE
      NTIM=NSTEP+1
      CALL CONCPLT (NANMAT,CONBOU,DAPLOT,ISO,NCELL,NISO1,NISO,NTIME,NP
     1 ,NTIM,NPP,NBCL,NBCR,VOL)
      IF (NPLOT.NE.0) CALL DRAW (NANMAT,RBAR,DAPLOT,NTIM,NP,CONCEN,DUM33
     1 ,NISO,NCELLS,NCELL,NTIME,ISO,NISO1,CONBOU,DIST,NCELL1,NBCL,NBCR)
      STOP
C
  100 FORMAT (2I4)
      END
```

358 166

```
      SUBROUTINE INPA (NANMAT,ANMAT,BRV,NXSEC,XSEC,NGP,NXS,NBR,NISO1
     1 ,NXSP)
C     INPUTA READS AND PRINTS THE NUCLEAR DATA
      DIMENSION NANMAT(NISO1,9), ANMAT(NISO1,2), BRV(NBR), NXSEC(NXSP,4)
     1 , HBRP(3,10), XSEC(NXS,NGP,4)
      COMMON /IO/ NINP, NOUT, IER, NPRINT
      COMMON /NUCDAT/ ISO, IBR, IXS, IGP
      DATA NHJ /6H      /
      I=1
      IBR=1
      II=1
C     READ NUCLEAR DATA
   10 READ (NINP,130) (NANMAT(I,J),J=1,9),ANMAT(I,1)
      IF (NANMAT(I,1).EQ.NHJ) GO TO 40
C     TEST FOR BRANCHING RATIOS
      DO 30 J=3,6
      IF (NANMAT(I,J)) 20,30,30
   20 READ (NINP,140) BRV(IBR)
      HBRP(1,IBR)=IABS(NANMAT(I,J))
      HBRP(2,IBR)=I
      HBRP(3,IBR)=BRV(IBR)
      IBR=IBR+1
   30 CONTINUE
      I=I+1
      GO TO 10
   40 I=I-1
      IBR=IBR-1
      IGP=0
C     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
   70 II=II-1
C     PRINT DECAY DATA
      WRITE (NOUT,160)
      WRITE (NOUT,180)
      WRITE (NOUT,190)
      WRITE (NOUT,180)
      WRITE (NOUT,200)
      WRITE (NOUT,170)
      LCNT=13
      DO 80 J=1,I
      WRITE (NOUT,210) (NANMAT(J,JJ),JJ=1,9),ANMAT(J,1)
      LCNT=LCNT+2
      IF (LCNT.GE.60) WRITE (NOUT,160)
   80 CONTINUE
C     PRINT BRANCHING RATIOS
      IF (IBR.EQ.0) GO TO 100
      WRITE (NOUT,160)
      WRITE (NOUT,180)
      WRITE (NOUT,220)
      DO 90 J=1,IBR
      WRITE (NOUT,230) HBRP(1,J),HBRP(2,J),HBRP(3,J)
   90 CONTINUE
  100 CONTINUE
C     PRINT CROSS SECTIONS
      IF (II.EQ.0) GO TO 120
      WRITE (NOUT,160)
      WRITE (NOUT,180)
      WRITE (NOUT,240)
      DO 110 J=1,II
      WRITE (NOUT,250) (NXSEC(J,JN),JN=1,4)
      DO 110 JJ=1,IGP
  110 WRITE (NOUT,260) JJ,(XSEC(J,JJ,JN),JN=1,4)
  120 CONTINUE
      ISO=I
```

```fortran
      IXS=II
      RETURN
C
  130 FORMAT (A7,8I4,E12.5)
  140 FORMAT (6(E12.5))
  150 FORMAT (2A6,2I4)
  160 FORMAT (1H1)
  170 FORMAT (/)
  180 FORMAT (/////)
  190 FORMAT (29X,37HDECAY CHAINS AND NUCLIDE RELATED DATA)
  200 FORMAT (24X,5HDECAY,7X,7HCAPTURE,28X,5HDECAY,/,24X,6HPARENT,6X,6HP
     1ARENT,4X,21HN-2N  N-ALPHA  N-P   ,3X,8HCONSTANT,/,5X,7HNUCLIDE,5X
     2 ,2HID,2(5X,7H1     2),/)
  210 FORMAT (5X,A7,1X,6(2X,I4),3X,I4,4X,I4,3X,1PE12.5)
  220 FORMAT (9X,15HBRANCHING RATIO,/,5X,4HFROM,4X,2HTO,6X,5HRATIO,/)
  230 FORMAT (5X,F4.0,3X,F4.0,3X,F7.4)
  240 FORMAT (10X,42HCROSS SECTIONS FOR TRANSMUTATION REACTIONS,///)
  250 FORMAT (//,10X,2A6,2I4,/,4X,5HGROUP,8X,7HN-GAMMA,11X,4HN-2N,12X,7H
     1N-ALPHA,12X,3HN-P)
  260 FORMAT (5X,I3,1X,4(5X,1PE12.5))
      END
      SUBROUTINE INPB (NCELLS,NGEOM,NBCR,NBCL,NTEMPS,IMATS,DIST,MATS,PHI
     1 ,TEMPS,TEMCOR,DIJO,AIJS,NCELL1,NCELL,NTEM,NISO,NMATS,NGP,DUM11
     2 ,DUM22,NN,CONBOU,CONINT,SOURCE,MM)
C     READ PROBLEM RELATED DATA
      DIMENSION DIST(NCELL1), MATS(NCELL), TEMPS(NTEM), TEMCOR(NTEM),
     1 DIJO(NISO,NISO,NMATS), AIJS(NISO,NISO,NMATS), DUM11(NN), DUM22(NN
     2 ), PHI(NCELL,NGP), CONBOU(NISO,2), CONINT(MM), SOURCE(MM)
      DIMENSION IGEOM(3)
      COMMON /IO/ NINP, NOUT, IER, NPRINT
      COMMON /NUCDAT/ ISO, IBR, IXS, IGP
      COMMON /TIMES/ TINT, TINC, TIMAX
      DATA IGEOM(1), IGEOM(2), IGEOM(3) /9HSLAB.    ,9HCYLINDER.,9HSPHER
     1E.  /
      READ (NINP,130) NCELLS,NGEOM,NBCL,NBCR,NTEMPS,IMATS,IGP
      PRINT (NOUT,140) IGEOM(NGEOM),NBCL,NBCR
      READ (NINP,150) TINT,TINC,TIMAX
      NRADI=NCELLS+1
      CALL REAG (DIST,NRADI,6HRADII ,6H      )
      CALL REAI (MATS,NCELLS,6HMATERI,6HALS   )
      CALL REAG (TEMPS,NTEMPS,6HTEMPER,6HATURES)
      CALL REAG (TEMCOR,NTEMPS,6HTEMP R,6HADII  )
      INDEX=ISO*ISO
      DO 30 I=1,IMATS
      CALL REAG (DUM11,INDEX,6HDIJ-0 ,6H      )
      CALL REAG (DUM22,INDEX,6HAIJ   ,6H      )
      DO 20 J=1,ISO
      DO 10 JJ=1,ISO
      IND=(J-1)*ISO+JJ
      DIJO(JJ,J,I)=DUM11(IND)
      AIJS(JJ,J,I)=DUM22(IND)
   10 CONTINUE
   20 CONTINUE
   30 CONTINUE
      IF (IXS.EQ.0) GO TO 60
      INDEX=IGP*NCELLS
      CALL REAG (DUM11,INDEX,6HCELL F,6HLUXES )
      DO 50 I=1,NCELLS
      DO 40 JJ=1,IGP
      IND=(I-1)*IGP+JJ
      PHI(I,JJ)=DUM11(IND)
   40 CONTINUE
   50 CONTINUE
   60 DO 80 I=1,2
      DO 70 J=1,ISO
      CONBOU(J,I)=0.0
   70 CONTINUE
   80 CONTINUE
      IF (NBCL.NE.2) GO TO 100
      CALL REAG (DUM11,ISO,6HLEFT C,6HONCEN )
      DO 90 I=1,ISO
```

358 168

```
      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
      CONBCU(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,6HL CONC)
      INDEX=ISO*NCELLS
      CALL REAG (SOURCE,INDEX,6HSOURCE,6H INPUT)
      RETURN
C
  130 FORMAT (18I4)
  140 FORMAT (///,1X,*THE GEOMETRY FOR THIS PROBLEM IS A *,A9,/,2X,*THE
     1LEFT BOUNDARY CONDITION IS =*,I2,/,2X*THE RIGHT BOUNDARY CONDITION
     2 IS =*,I2,//)
  150 FORMAT (3E12.6,I12)
      END
      SUBROUTINE GEOM (NGEOM,DIST,DELR,AREA,VOL,RBAR,N,NCELLS)
C     SET UP GEOMETRY
C     CALCULATE CELL AREAS AND VOLUMES
      DIMENSION DIST(N), DELR(NCELLS), AREA(NCELLS), VOL(NCELLS), RBAR
     1 (NCELLS)
C     N=NCELLS+1
C     NGEOM=1,2,3 - SLAB,CYLINDER,SPHERE
      PI=3.1415926
      IF (NGEOM-2) 10,30,50
   10 DO 20 I=1,NCELLS
      DELR(I)=DIST(I+1)-DIST(I)
      RBAR(I)=0.5*(DIST(I+1)+DIST(I))
      AREA(I)=1.0
      VOL(I)=DELR(I)
   20 CONTINUE
      RETURN
   30 DO 40 I=1,NCELLS
      DELR(I)=DIST(I+1)-DIST(I)
      RBAR(I)=0.5*(DIST(I+1)+DIST(I))
      AREA(I)=2*PI*DIST(I+1)
      VOL(I)=2.*PI*DELR(I)*RBAR(I)
   40 CONTINUE
      RETURN
   50 DO 60 I=1,NCELLS
      DELR(I)=DIST(I+1)-DIST(I)
      RBAR(I)=(DIST(I+1)**2+DIST(I+1)*DIST(I)+DIST(I)**2)/3.0
      AREA(I)=4.0*PI*DIST(I+1)**2
      VOL(I)=4.0*PI*RBAR(I)*DELR(I)
      RBAR(I)=SQRT(RBAR(I))
   60 CONTINUE
      RETURN
      END
      SUBROUTINE INPLT (NCELL,NCELL1,NBCL,NBCR,MATS,DIST,NGEOM,NCELLS)
C     DRAW A MAP OF THE PROBLEM GEOMETRY
      DIMENSION MATS(NCELL), DIST(NCELL1), NBOU(2), IGEOM(3)
      COMMON /IO/ NINP, NOUT, IER, NPRINT
      DATA ISTR /1H*/, NBOU(1), NBOU(2) /1H1,1H2/
      DATA IGEOM(1), IGEOM(2), IGEOM(3) /9HSLAB.    ,9HCYLINDER.,9HSPHER
     1E.  /
C     TERMINAL OUTPUT
      NBOX=8
      IF (NPRINT.EQ.1) GO TO 10
C     LINE PRINTER OUTPUT
      NBOX=12
      WRITE (NOUT,180)
      WRITE (NOUT,190) IGEOM(NGEOM)
   10 CONTINUE
      WRITE (NOUT,170)
      IF (MOD(NCELLS,NBOX)) 20,30
   20 LOOP=(NCELLS/NBOX)+1
```

358 169

```
      GO TO 40
 3    LOOP=(NCELLS/NBOX)
 40   ICELL=NCELLS
      ICOUNT=0
      DO 120 I=1,LOOP
      ICOUNT=ICOUNT+1
      IS=1
      IE=MINO(NBOX.ICELL)
      IMM=IE-IS
      IMM1=IMM-1
      IM2=(IMM+1)*8-1
      IM3=IMM*8-1
      NDX=(I*NBOX-NBOX)+1
      NDS=IMM+(I*NBOX-NBOX)
      NDRS=NDS+1
      NDRE=NDRS+1
      ICELL=ICELL-NBOX
      IF (LOOP.EQ.1) GO TO 100
      IF (I.EQ.1) GO TO 50
      IF (ICOUNT.EQ.LOOP) GO TO 70
      WRITE (NOUT,150) ISTR,(ISTR,J=1,IM2)
      WRITE (NOUT,130) ISTR,(ISTR,J=1,IMM)
      WRITE (NOUT,140) ISTR,(MATS(J),ISTR,J=NDX,NDS),MATS(NDS+1)
      WRITE (NOUT,130) ISTR,(ISTR,J=1,IMM)
      WRITE (NOUT,150) ISTR,(ISTR,J=1,IM2)
      WRITE (NOUT,160) (DIST(J),J=NDX,NDRS)
      WRITE (NOUT,170)
      GO TO 120
 50   IF ((NGEOM.NE.1).AND.(NBCL.EQ.2)) GO TO 60
      WRITE (NOUT,150) NBOU(NBCL),(ISTR,J=1,IM2)
      WRITE (NOUT,130) NBOU(NBCL),(ISTR,J=1,IMM)
      WRITE (NOUT,140) NBOU(NBCL),(MATS(J),ISTR,J=NDX,NDS),MATS(NDS+1)
      WRITE (NOUT,130) NBOU(NBCL),(ISTR,J=1,IMM)
      WRITE (NOUT,150) NBOU(NBCL),(ISTR,J=1,IM2)
      WRITE (NOUT,160) (DIST(J),J=NDX,NDRS)
      WRITE (NOUT,170)
      GO TO 120
 60   WRITE (NOUT,200) (ISTR,J=1,8),NBOU(NBCL),(ISTR,J=1,IM3)
      WRITE (NOUT,130) ISTR,NBOU(NBCL),(ISTR,J=1,IMM1)
      WRITE (NOUT,210) ISTR,MATS(1),NBOU(NBCL),(MATS(J),ISTR,J=2,NDS)
     1 ,MATS(NDS+1)
      WRITE (NOUT,130) ISTR,NBOU(NBCL),(ISTR,J=1,IMM1)
      WRITE (NOUT,200) (ISTR,J=1,8),NBOU(NBCL),(ISTR,J=1,IM3)
      WRITE (NOUT,160) (DIST(J),J=NDX,NDRS)
      WRITE (NOUT,170)
      GO TO 120
 70   WRITE (NOUT,150) ISTR,(ISTR,J=1,IM2),NBOU(NBCR)
      IF (IMM.EQ.0) GO TO 80
      WRITE (NOUT,130) ISTR,(ISTR,J=1,IMM),NBOU(NBCR)
      WRITE (NOUT,140) ISTR,(MATS(J),ISTR,J=NDX,NDS),MATS(NDS+1),NBOU
     1 (NBCR)
      WRITE (NOUT,130) ISTR,(ISTR,J=1,IMM),NBOU(NBCR)
      GO TO 90
 80   WRITE (NOUT,130) ISTR,NBOU(NBCR)
      WRITE (NOUT,140) ISTR,MATS(NDS+1),NBOU(NBCR)
      WRITE (NOUT,130) ISTR,NBOU(NBCR)
 90   WRITE (NOUT,150) ISTR,(ISTR,J=1,IM2),NBOU(NBCR)
      WRITE (NOUT,160) (DIST(J),J=NDX,NDRE)
      WRITE (NOUT,170)
      GO TO 120
 100  IF ((NGEOM.NE.1).AND.(NBCL.EQ.2)) GO TO 110
      WRITE (NOUT,150) NBOU(NBCL),(ISTR,J=1,IM2),NBOU(NBCR)
      WRITE (NOUT,130) NBOU(NBCL),(ISTR,J=1,IMM),NBOU(NBCR)
      WRITE (NOUT,140) NBOU(NBCL),(MATS(J),ISTR,J=NDX,NDS),MATS(NDS+1)
     1 ,NBOU(NBCR)
      WRITE (NOUT,130) NBOU(NBCL),(ISTR,J=1,IMM),NBOU(NBCR)
      WRITE (NOUT,150) NBOU(NBCL),(ISTR,J=1,IM2),NBOU(NBCR)
      WRITE (NOUT,160) (DIST(J),J=NDX,NDRE)
      WRITE (NOUT,170)
      GO TO 120
 110  WRITE (NOUT.200) (ISTR.J=1,8),NBOU(NBCL),(ISTR.J=1,IM3),NBOU(NBCR)
```


358 170

```
      WRITE (NOUT,130) ISTR,NBOU(NBCL),(ISTR,J=1,IMM1),NBOU(NBCR)
      WRITE (NOUT,210) ISTR,MATS(1),NBOU(NBCL),(MATS(J),ISTR,J=2,NDS)
     1 ,MATS(NDS+1),NBOU(NBCR)
      WRITE (NOUT,130) ISTR,NBOU(NBCL),(ISTR,J=1,IMM1),NBOU(NBCR)
      WRITE (NOUT,200) (ISTR,J=1,8),NBOU(NBCL),(ISTR,J=1,IM3),NBOU(NBCR)
      WRITE (NOUT,160) (DIST(J),J=NDX,NDRE)
      WRITE (NOUT,170)
  120 CONTINUE
      RETURN
C
  130 FORMAT (5X,A1,15(7X,A1))
  140 FORMAT (5X,A1,15(I4,3X,A1))
  150 FORMAT (5X,A1,15(8A1))
  160 FORMAT (3X,16(F7.3,1X))
  170 FORMAT (/)
  180 FORMAT (1H1,//)
  190 FORMAT (54X,13HGEOMETRIC MAP,/50X,20H(NOT DRAWN TO SCALE),//43X,35
     1 HTHE GEOMETRY FOR THIS PROBLEM IS A  ,A9,//37X,47HBOUNDARY CONDITI
     2ON TYPE INDICATED ON BOUNDARIES,/40X,41HMATERIAL TYPE INDICATED IN
     3 CENTER OF CELL,/39X,43HRADII GIVEN FROM CENTER LINE IN CENTIMETER
     4S,//)
  200 FORMAT (5X,8A1,A1,14(8A1))
  210 FORMAT (5X,A1,3X,I1,3X,A1,14(I4,3X,A1))
      END
      SUBROUTINE CONCPLT (NANMAT,CONBOU,DAPLOT,ISO,NCELL,NISO1,NISO
     1 ,NTIME,NP,NTIM,NPP,NBCL,NBCR,VOL)
C     PRINT RESULTS OF PROBLEM
      DIMENSION NANMAT(NISO1,9), CONBOU(NISO,2), DAPLOT(NISO,NCELL,NTIME
     1 ), NPP(NCELL), VOL(NCELL)
      COMMON /IO/ NINP, NOUT, IER, NPRINT
      COMMON /TIMES/ TINT, TINC, TIMAX
      DATA BNDRY /10HREFLECTED /
      TIME=0.0
C     TERMINAL OUTPUT
      NBOX=4
C     LINE PRINTER OUTPUT
      IF (NPRINT.EQ.0) NBOX=6
      NBOX1=NBOX+1
      NP1=NP-NBOX
      IF (MOD(NP1,NBOX1)) 20,10,10
   10 LOOP=NP1/NBOX1+2
      GO TO 30
   20 LOOP=NP1/NBOX1+1
   30 DO 40 I=1,NP
   40 NPP(I)=I
      IF (NBCL.EQ.2) GO TO 60
      DO 50 K=1,ISO
   50 CONBOU(K,1)=BNDRY
   60 IF (NBCR.EQ.2) GO TO 80
      DO 70 K=1,ISO
   70 CONBOU(K,2)=BNDRY
   80 DO 470 I=1,NTIM
      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
      ICOUNT=ICOUNT+1
      IE=MIN0(NBOX,INP)
      NDX=(II*NBOX-NBOX)+1
      NDS=IE+(II*NBOX-NBOX)
      INP=INP-NBOX
      IF (LOOP.EQ.1) GO TO 240
      IF (II.EQ.1) GO TO 120
      IF (ICOUNT.EQ.LOOP) GO TO 160
      IE=IE+1
      NDX=NDX+II-2
```


358 171

```
      NDS=NDS+II-1
      INP=INP-1
      WRITE (NOUT,550) IE,(NPP(J),J=NDX,NDS)
      DO 110 III=1,ISO
  110 WRITE (NOUT,560) NANMAT(III,1),(DAPLOT(III,J,I),J=NDX,NDS)
      GO TO 440
  120 WRITE (NOUT,510) IE,(NPP(J),J=NDX,NDS)
      IF (NBCL.EQ.1) GO TO 140
      DO 130 III=1,ISO
  130 WRITE (NOUT,520) NANMAT(III,1),CONBOU(III,1),(DAPLOT(III,J,I),J
     1 =NDX,NDS)
      GO TO 440
  140 DO 150 III=1,ISO
  150 WRITE (NOUT,630) NANMAT(III,1),CONBOU(III,1),(DAPLOT(III,J,I),J
     1 =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) NANMAT(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) NANMAT(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) NANMAT(III,1),CONBOU(III,2)
      GO TO 440
  220 DO 230 III=1,ISO
  230 WRITE (NOUT,650) NANMAT(III,1),CONBOU(III,2)
      GO TO 440
  240 IF (IE.EQ.NBOX) 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 (NBCL.EQ.1) GO TO 260
      IF (NBCR.EQ.1) GO TO 280
      DO 250 III=1,ISO
  250 WRITE (NOUT,600) NANMAT(III,1),CONBOU(III,1),IE,(DAPLOT(III,J,I),J
     1 =NDX,NDS),CONBOU(III,2)
      GO TO 440
  260 DO 270 III=1,ISO
  270 WRITE (NOUT,660) NANMAT(III,1),CONBOU(III,1),IE,(DAPLOT(III,J,I),J
     1 =NDX,NDS),CONBOU(III,2)
      GO TO 440
  280 DO 290 III=1,ISO
  290 WRITE (NOUT,670) NANMAT(III,1),CONBOU(III,1),IE,(DAPLOT(III,J,I),J
     1 =NDX,NDS),CONBOU(III,2)
      GO TO 440
  300 DO 310 III=1,ISO
  310 WRITE (NOUT,680) NANMAT(III,1),CONBOU(III,1),IE,(DAPLOT(III,J,I),J
     1 =NDX,NDS),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 (NBCL.EQ.1) GO TO 350
      IF (NBCR.EQ.1) GO TO 380
      DO 330 III=1,ISO
  330 WRITE (NOUT,520) NANMAT(III,1),CONBOU(III,1),(DAPLOT(III,J,I),J
     1 =NDX,NDS)
      WRITE (NOUT,530)
      WRITE (NOUT,610)
      DO 340 III=1,ISO
  340 WRITE (NOUT,620) NANMAT(III,1),CONBOU(III,2)
      GO TO 440
  350 DO 360 III=1,ISO
```


358 172

```
  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)
  460 CONTINUE
      WRITE (NOUT,480) ATOMS
  470 CONTINUE
      RETURN
C
  480 FORMAT (15H NO. OF ATOMS =,1PE22.15)
  490 FORMAT (1H1,//,37X,23HCELL CONCENTRATIONS AT ,1PE12.5,5H DAYS,/40X
     1 ,28H(CONCENTRATIONS IN ATOMS/CC))
  500 FORMAT (//,20X,23HCELL CONCENTRATIONS AT ,1PE12.5,5H DAYS,/23X,28H
     1(CONCENTRATIONS IN ATOMS/CC))
  510 FORMAT (13X,4HLEFT,/22H ISOTOPE   BOUNDARY   ,=(3X,5HCELL ,I2,3X))
  520 FORMAT (1X,A7,1PE13.5,8(1PE13.5))
  530 FORMAT (/)
  540 FORMAT (1H1,//)
  550 FORMAT (9H ISOTOPE ,=(3X,5HCELL ,I2,3X))
  560 FORMAT (1X,A7,9(1PE13.5))
  570 FORMAT (9X,=(13X),4X,5HRIGHT,/9H ISOTOPE ,=(3X,5HCELL ,I2,3X),10H
     1 BOUNDARY)
  580 FORMAT (1X,A7,=(1PE13.5),1PE13.5)
  590 FORMAT (13X,4HLEFT,4X,=(13X),4X,5HRIGHT,/22H ISOTOPE   BOUNDARY
     1 ,=(3X,5HCELL ,I2,3X),9H BOUNDARY)
  600 FORMAT (1X,A7,1PE13.5,=(1PE13.5),1PE13.5)
  610 FORMAT (13X,5HRIGHT,/19H ISOTOPE   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,A7,3X,A10)
  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),2X,A10)
  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,A7,3X,A10,8(1PE13.5))
  740 FORMAT (1X,A7,3X,A10)
      END
      SUBROUTINE TEMADJ (TEMPS,TEMCOR,NTEMPS,RBAR,NCELLS,W,A,B,C,NTEM)
C     EVALUATE TEMPERATURE FIELD FROM DATA SUPPLIED IN INPE
```


358 173

```
      DIMENSION TEMPS(NTEM), TEMCOR(NTEM), REAR(NCELLS), W(NTEM), A(NTEM
     1 ), B(NTEM), C(NTEM), IOP(2), TAB(3)
      IJ=1
      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
      RETURN
      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 BB(IK,JK)=0.0
      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,NM,MM,IGP,NGP,NXS,NCELL,K)
      IF (NANMAT(IK,IDX)) 70,70,80
   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 (1H0,4X,*CROSS SECTIONS CANNOT BE FOUND FOR NUCLIDE *,I4)
      END
```


358 174

```
      SUBROUTINE DIJADJ (DIJO,AIJS,TEMPS,DIFFK,NISO,NMATS,NCELL,ISO,K,KK
     1 )
C     USE ARRHENIUS RELATION TO ADJUST DIFFUSION COEFFICIENTS
      DIMENSION DIJO(NISO,NISO,NMATS), AIJS(NISO,NISO,NMATS)
      DIMENSION DIFFK(ISO,ISO), TEMPS(NCELL)
C     R=1.987 CAL/K-MOLE
C     T DEGREES KELVIN
C     DIJO CM**2/SEC
C     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)
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE BCONL (BK,AREA,DIFFK,DELR,SOURCE,CONBOU,NCELLS,ISO,NBCL
     1 ,NGEOM,NISO,NIC,DUM11,DUM33)
C     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.NE.1) GO TO 80
      DO 20 I=1,ISO
      DO 10 J=1,ISO
      BK(I,J)=0.0
   10 CONTINUE
   20 CONTINUE
      RETURN
   30 IF (NGEOM.GT.1) GO TO 40
      CON=2./DELR(1)
      CALL SCALAR (DIFFK,CON,BK,ISO,ISO)
      GO TO 50
   40 CON=2.*AREA(1)/DELR(2)
      CALL SCALAR (DIFFK,CON,BK,ISO,ISO)
   50 CONTINUE
      DO 60 I=1,ISO
      DUM11(I)=CONBOU(I,1)
   60 CONTINUE
      CALL MATMPY (ISO,ISO,1,BK,ISO,DUM11,NISO,DUM33,NIC)
      DO 70 I=1,ISO
   70 SOURCE(I)=SOURCE(I)+DUM33(I)
      RETURN
   80 CONTINUE
      PRINT 90
      RETURN
C
   90 FORMAT (5X,*LEFT BOUNDARY CONDITION IMPROPERLY SPECIFIED*)
      END
      SUBROUTINE BCONR (AK,AREA,DIFFK,DELR,SOURCE,CONBOU,NCELLS,ISO,NBCR
     1 ,NISO,NIC,DUM11,DUM22,BK,CKK,ALAM,VOL,DIFDUM1)
C     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 DIFDUM1(ISO,ISO)
      IF (NBCR.EQ.2) GO TO 30
      IF (NBCR.NE.1) GO TO 90
      DO 20 I=1,ISO
      DO 10 J=1,ISO
      AK(I,J)=0.0
   10 CONTINUE
   20 CONTINUE
      GO TO 60
   30 CON=2.*AREA(NCELLS)/DELR(NCELLS)
      CALL SCALAR (DIFFK,CON,AK,ISO,ISO)
      DO 40 I=1,ISO
      DUM11(I)=CONBOU(I,2)
   40 CONTINUE
```

358 175

```
      CALL MATMPY (ISO,ISO,1,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
  100 FORMAT (5X,*RIGHT BOUNDARY CONDITION IMPROPERLY SPECIFIED*)
      END
      SUBROUTINE BIGEL (DIFFK,DIFFK1,DIFDUM1,DIFDUM2,ALAM,AK,BK,CKK,ISO
     1 ,AREA,VOL,DELR,NCELLS,IPVT,Z,I)
C     EVALUATE ELEMENTS OF BIG MATRIX; A, B, AND K
      DIMENSION DIFFK(ISO,ISO), DIFFK1(ISO,ISO), DIFDUM1(ISO,ISO),
     1 DIFDUM2(ISO,ISO), ALAM(ISO,ISO), AK(ISO,ISO), BK(ISO,ISO), CKK
     2 (ISO,ISO), IPVT(ISO), Z(ISO), DET(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 (DIFDUM1,ISO,ISO,IPVT,RCOND,Z)
      CALL SGEDI (DIFDUM1,ISO,ISO,IPVT,DET,Z,01)
      CALL MATMPY (ISO,ISO,ISO,DIFDUM1,ISO,DIFFK1,ISO,DIFDUM2,ISO)
      CALL MATMPY (ISO,ISO,ISO,DIFFK,ISO,DIFDUM2,ISO,DIFDUM1,ISO)
      CON=4.*AREA(I)/(DELR(I)+DELR(I+1))
      CALL SCALAR (DIFDUM1,CON,AK,ISO,ISO)
      CALL SCALAR (ALAM,VOL(I),DIFDUM1,ISO,ISO)
      DO 40 I1=1,ISO
      DO 30 I2=1,ISO
      CKK(I1,I2)=-AK(I1,I2)-BK(I1,I2)-DIFDUM1(I1,I2)
      BK(I1,I2)=AK(I1,I2)
   30 CONTINUE
   40 CONTINUE
      RETURN
C
      END
      SUBROUTINE MAKEB (BIGB,AK,BK,CKK,NIC,ISO,VOL,NCELLS,ICOL,II)
C     CONSTRUCT BIG MATRIX
      DIMENSION BIGB(NIC,NIC), AK(ISO,ISO), BK(ISO,ISO), CKK(ISO,ISO),
     1 VOL(NCELLS)
      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
```


358 176

```
      BIGB(IR1,IC1)=CKK(I,J)/VOL(IVOL)
   30 CONTINUE
   40 CONTINUE
      RETURN
      END
      SUBROUTINE WXSEC (SIGPHI,PHI,XSEC,M,N,IGP,NGP,NXS,NCELL,K)
C     WEIGHT CROSS SECTIONS
      DIMENSION PHI(NCELL,NGP), XSEC(NXS,NGP,4)
      SIGPHI=0.0
      DO 10 J=1,IGP
   10 SIGPHI=SIGPHI+PHI(K,J)*XSEC(M,J,N)
      RETURN
      END
      SUBROUTINE MAKVOL (VOLMAT,NIC,VOL,NCELLS,II,ISO)
C     CONSTRUCT DIAGONAL VOLUME MATRIX
      DIMENSION VOLMAT(NIC,NIC), VOL(NCELLS)
      LEVEL 2, VOLMAT
      DO 20 I=1,NIC
      DO 10 J=1,NIC
      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 SOLVER (A,B,C,D,E,F,TINCD,I,NN)
C     SOLVER EVALUATES D(A) AND I+A*D(A)
C     THESE VALUES ARE RETURNED IN D AND E.
C     THE FOLLOWING ARE REQUIRED ROUTINES
C          IFACT - CALCULATES FACTORIALS
C          SCALAR - MULTIPLIES A SCALAR TIMES A MATRIX
C          GENID - CREATES AN IDENTITY MATRIX
C          MATMOV - SETS TWO MATRICES EQUAL
C          MPYEC3 - MULTIPLIES TWO LCM MATRICES - CALLS SDOT
C          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
   10 CONTINUE
   20 CONTINUE
      SUM=0.0
      DO 40 JJ=1,I
      DO 30 J=1,I
      SUM=SUM+A(J,JJ)*A(J,JJ)
   30 CONTINUE
   40 CONTINUE
C     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
   50 NP=1
      GO TO 70
   60 NP=P+1.0
   70 CONTINUE
      S=1.
      DO 80 NLOOP=1,NP
      S=S*2.
   80 CONTINUE
C     THIS LOOP IS USED IN PLACE OF 2**NP AS THAT WAS SET TO ZERO
C     FOR NP GREATER THAN 48 (CDC-6600)
      T=TINCD/S
      CALL SCALRS (A,T,C,I,NN)
      CALL GENID (D,I,NN)
```

358 177

```
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,I
      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,I
      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,B,NN)
      CALL MOVECS (I,I,B,NN,E,NN)
      DO 120 JJ=1,I
      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,B,NN,C,NN)
  130 CONTINUE
C     C CONTAINS THE ENTIRE PRODUCT
      CALL SCAECS (D,SI,F,I,NN)
      CALL MPYEC3 (I,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,I
      DO 140 J=1,I
      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 FSOLVE (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
C     CALL PRIMES(VOLMAT,NIC,NIC,NIC,NIC,10H VOLMAT    ,10H          )
C     CALL PRIV(CONINT,NIC,NIC,10H CONINT    ,10H          )
      CALL MPYEC1 (NIC,NIC,1,VOLMAT,NIC,CONINT,NIC,CONCEN,NIC)
C     CALL PRIV(CONCEN,NIC,NIC,10H VOLMAT*CO,10HNINT      )
      CALL MPYEC1 (NIC,NIC,1,E,NIC,CONCEN,NIC,CONINT,NIC)
C     CALL PRIV(CONINT,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)
C     CALL PRIV(CONCEN,NIC,NIC,10H FIRST PAR,10HT         )
      CALL MPYEC1 (NIC,NIC,1,D,NIC,SOURCE,NIC,CONINT,NIC)
C     CALL PRIV(CONINT,NIC,NIC,10H D*SOURCE ,10H          )
      CALL SCAECS (VOLMAT,DTIME,C,NIC,NIC)
C     CALL PRIMES(C,NIC,NIC,NIC,NIC,10H DTIME*VOL,10HMAT       )
      CALL MPYEC1 (NIC,NIC,1,C,NIC,CONINT,NIC,DUM33,NIC)
C     CALL PRIV(DUM33,NIC,NIC,10H SECOND PA,10HRT        )
      DO 20 I=1,NIC
      VOLMAT(I,I)=1./VOLMAT(I,I)
      CONCEN(I)=CONCEN(I)+DUM33(I)
   20 CONTINUE
C     CALL PRIV(CONCEN,NIC,NIC,10H ANSWER    ,10H          )
      RETURN
      END
```


358 178

```
      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 SCAECS (A,S,B,N,NDIM)
C     SCAECS 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 10 I=1,N
   10 A(I,J)=0.0
   20 A(J,J)=1.0
      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)=SDOT(M,A(I,1),IA,B(1,J),1)
      RETURN
      END
      SUBROUTINE MPYEC3 (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 10 K=1,M
```

358 179

```
   10 AM=AM+A(I,K)*B(K,J)
   20 C(I,J)=AM
      RETURN
      END
      SUBROUTINE MPYEC1 (N,M,L,A,IA,B,IB,C,IC)
C     MULTIPLY A SCM AND LCM MATRIX
      DIMENSION A(IA,M), B(IB,L), C(IC,L)
      LEVEL 2, A
      DO 20 J=1,L
      DO 20 I=1,N
      AM=0.0
      DO 10 K=1,M
   10 AM=AM+A(I,K)*B(K,J)
   20 C(I,J)=AM
      RETURN
      END
      SUBROUTINE PRIM (A,N1,N2,N1D,N2D,TITLE1,TITLE2)
C     PRINT A MATRIX
      DIMENSION A(N1D,N2D)
      PRINT 20, TITLE1,TITLE2
      DO 10 J=1,N1
   10 PRINT 30, (A(J,JJ),JJ=1,N2)
      RETURN
C
   20 FORMAT (/1X,2A10/)
   30 FORMAT (6E13.5)
      END
      SUBROUTINE PRIMES (A,N1,N2,N1D,N2D,TITLE1,TITLE2)
C     PRINT A LCM MATRIX
      DIMENSION A(N1D,N2D)
      LEVEL 2, A
      PRINT 20, TITLE1,TITLE2
      DO 10 J=1,N1
   10 PRINT 30, (A(J,JJ),JJ=1,N2)
      RETURN
C
   20 FORMAT (/1X,2A10/)
   30 FORMAT (6E13.5)
      END
      SUBROUTINE PRIV (A,N1,N1D,TITLE1,TITLE2)
C     PRINT A VECTOR
      DIMENSION A(N1D)
      PRINT 10, TITLE1,TITLE2
      PRINT 20, (A(I),I=1,N1)
      RETURN
C
   10 FORMAT (/,1X,2A10,/)
   20 FORMAT (6E13.5)
      END
      SUBROUTINE REAC (ARRAY,NCOUNT,HOL1,HOL2)
C
C     READS FLOATING POINT DATA
      DIMENSION ARRAY(NCOUNT), V(12), K(12), IN(12)
      COMMON /IO/ NINP, NOUT, IER, NPRINT
      JFLAG=0
      J=1
   10 IF (JFLAG.EQ.0) GO TO 30
      DO 20 JJ=1,6
      K(JJ)=K(JJ+6)
      IN(JJ)=IN(JJ+6)
   20 V(JJ)=V(JJ+6)
      JFLAG=0
      GO TO 40
   30 READ (NINP,200) (K(I),IN(I),V(I),I=1,6)
   40 DO 160 I=1,6
      L=K(I)+1
      GO TO (50,60,80,170,120), L
C     NO MODIFICATION
   50 ARRAY(J)=V(I)
      J=J+1
      GO TO 160
```


358 180

```
C     REPEAT
   60 L=IN(I)
      DO 70 M=1,L
      ARRAY(J)=V(I)
   70 J=J+1
      GO TO 160
C     INTERPOLATE
   80 IF (I-6) 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)
  110 J=J+1
      GO TO 160
C     INTERPOLATE WITH CONSTANT RATIO
  120 IF (I.LT.6) GO TO 130
      READ (NINP,200) (K(JJ),IN(JJ),V(JJ),JJ=7,12)
      JFLAG=1
  130 L=MAX0(2,IN(I)+1)
      T1=0.
      T2=1.
      DO 140 JJ=1,L
      T1=T1+T2
  140 T2=T2*V(I)
      T2=(V(I+1)-ARRAY(J-1))/T1
      L=MAX0(1,IN(I))
      DO 150 JJ=1,L
      ARRAY(J)=ARRAY(J-1)+T2
      T2=T2*V(I)
  150 J=J+1
  160 CONTINUE
      GO TO 10
C     TERMINATE
  170 J=J-1
      WRITE (NOUT,210) HOL1,HOL2,J,(ARRAY(I),I=1,NCOUNT)
      IF (J-NCOUNT) 180,190,180
  180 WRITE (NOUT,220) HOL1,HOL2
      IER=1
  190 RETURN
C
  200 FORMAT (6(I1,I2,E9.3))
  210 FORMAT (/1X,2A6,I6/(6E13.5))
  220 FORMAT (/33H INCORRECT NUMBER OF INPUT ITEMS ,2A6)
      END
      SUBROUTINE REAI (IARRAY,NCOUNT,HOL1,HOL2)
C
C     READS INTEGER DATA
      DIMENSION IARRAY(NCOUNT), IV(6), K(6), IN(6)
      COMMON /IO/ NINP, NOUT, IER, NPRINT
      J=1
   10 READ (NINP,100) (K(I),IN(I),IV(I),I=1,6)
      DO 60 I=1,6
      L=K(I)+1
      GO TO (20,30,50,70), L
C     NO MODIFICATION
   20 IARRAY(J)=IV(I)
      J=J+1
      GO TO 60
C     REPEAT
   30 L=IN(I)
      DO 40 M=1,L
      IARRAY(J)=IV(I)
   40 J=J+1
      GO TO 60
C     INTERPOLATE
   50 WRITE (NOUT,120) HOL1,HOL2
      IER=1
      RETURN
   60 CONTINUE
```

```
      GO TO 10
C     TERMINATE
   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 (/1X,2A6,I6/(6I12))
  120 FORMAT (44HOATTEMPTING TO INTERPOLATE BETWEEN INTEGERS ,2A6)
  130 FORMAT (33HOINCORRECT NUMBER OF INPUT ITEMS ,2A6)
      END
      SUBROUTINE DRAW (NANMAT,RBAR,DAPLOT,NTIM,NP,CONCEN,DUM33,NISO
     1 ,NCELLS,NCELL,NTIME,ISO,NISO1,CONBOU,DIST,NCELL1,NBCL,NBCR)
C     DRAW PLOTS OF RESULTS
      DIMENSION NANMAT(NISO1,2), RBAR(NCELLS), CONCEN(NP), DUM33(NP),
     1 DAPLOT(NISO,NCELL,NTIME), DIST(NCELL1), CONBOU(NISO,2)
      DIMENSION X(2), Y(2), LABELX(3), LABELY(4)
      CALL GPLOT (1HU,10HDASH PLOTS,10)
      CALL BGNPL (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.**XSCALE
      IF (NP.EQ.NCELLS) GO TO 40
      IF (XSCALE.GE.2.) GO TO 20
      DO 10 I=1,NP
   10 DUM33(I)=RBAR(I+1)
      NX=13
      ENCODE (NX,160,LABELX)
      GO TO 80
   20 X(2)=X(2)/XSCLDIV
      IXS=XSCALE
      NX=23
      ENCODE (NX,170,LABELX) IXS
      DO 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)
      NX=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
   80 DO 150 I=1,ISO
      Z=0.0
      DO 100 NT=1,NTIM
      DO 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 (NBCL.EQ.1) Y(2)=AMAX1(CONBOU(I,2),Z)
      IF (NBCR.EQ.1) Y(2)=AMAX1(CONBOU(I,1),Z)
      IF (NBCR.EQ.1.AND.NBCL.EQ.1) Y(2)=Z
      YSCALE=AINT(ALOG10(Y(2)))
      YSCLDIV=1.0
      IF (YSCALE.LT.2.) GO TO 110
      YSCLDIV=10.**YSCALE
      Y(2)=Y(2)/YSCLDIV
      IYS=YSCALE
      NY=33
```


358 182

```
      ENCODE (NY,180,LABELY) IYS
      GO TO 120
  110 NY=23
      ENCODE (NY,190,LABELY)
  120 CALL TITLE (NANMAT(I,1),7,LABELX,NX,LABELY,NY,5.5,8.)
      CALL GRAF (X(1),10HSCALE     ,X(2),Y(1),10HSCALE     ,Y(2))
      DO 140 NT=1,NTIM
      DO 130 J=1,NP
      CONCEN(J)=DAPLOT(I,J,NT)
  130 CONCEN(J)=CONCEN(J)/YSCLDIV
      CALL CURVE (DUM33,CONCEN,NP,0)
  140 CONTINUE
      CALL ENDPL (I)
  150 CONTINUE
      CALL DONEPL
      RETURN
C
  160 FORMAT (13HDISTANCE (CM))
  170 FORMAT (21HDISTANCE (CM) X 10**-,I2)
  180 FORMAT (31HCONCENTRATION (ATOM/CC) X 10**-,I2)
  190 FORMAT (23HCONCENTRATION (ATOM/CC))
      END
      SUBROUTINE SPL1D1 (N,X,F,W,IOP,IJ,A,B,C)
C     WHERE N= NUMBER OF POINTS IN THE INTERPOLATION
C           X= ORIGIN OF TABLE OF INDEPENDENT VARIABLE
C           F= ORIGIN OF TABLE OF DEPENDENT VARIABLE
C           W= AN ARRAY OF DIMENSION N WHICH CONTAINS THE CALCULATED
C              SECOND DERIVATIVES UPON RETURN
C           IOP= AN ARRAY OF DIMENSION 2 WHICH CONTAINS COMBINATIONS OF
C                THE INTEGERS 1 THRU 5 USED TO SPECIFY THE BOUNDARY
C                CONDITIONS
C           IJ= SPACING I" THE F AND W TABLES
C           A,B,C= ARRAYS OF DIMENSION N USED FOR TEMPORARY STORAGE
C
      DIMENSION IOP(2), X(4), F(2), W(2), A(2), B(2), C(2), COMM(6)
      DATA (COMM(J),J=1,6) /8HSPL1D1 N,8H LESS TH,8HAN 4. RE,8HSULTS IN,
     1 8HCORRECT.,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)*IJ+1
      J1=M+IJ
      J2=M-IJ
      CON=(X(I+1)-X(I-1))/3.
      DON=(X(I)-X(I-1))/6.
      B(I)=CON-(DON**2)/B(I-1)
      E=(F(J1)-F(M))/(X(I+1)-X(I))-(F(M)-F(J2))/(X(I)-X(I-1))
      W(M)=E-(DON*W(J2))/B(I-1)
   20 A(I)=-(DON*A(I-1))/B(I-1)
   30 K1=(N-2)*IJ+1
      C(N-1)=-((X(N)-X(N-1))/6.)/B(N-1)
      W(K1)=W(K1)/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-I
      CON=(X(J+1)-X(J))/6.
      A(J)=(A(J)-CON*A(J+1))/B(J)
      C(J)=-(CON*C(J+1))/B(J)
      K3=(J-1)*IJ+1
      M=K3+IJ
   50 W(K3)=(W(K3)-CON*W(M))/B(J)
   60 K4=(N-1)*IJ+1
      IF (IOP(1)-5) 70,90,70
   70 C1=W(1)
      IF (IOP(2)-5) 80,110,80
   80 C2=W(K4)
      GO TO 130
```


358 183

```
 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)=F(1)*(1./A1+1./A2+1./A3)-A2*A3*F(IJ+1)/(A1*A4*A5)+A1*A3*F(2
   1 *IJ+1)/(A2*A4*A6)-A1*A2*F(3*IJ+1)/(A3*A5*A6)
    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=L1-IJ
    L3=L2-IJ
    W(K4)=-B2*B3*F(L3)/(B6*B4*B1)+B1*B3*F(L2)/(B6*B5*B2)-B1*B2*F(L1)/
   1 (B4*B5*B3)+F(K4)*(1./B1+1./B2+1./B3)
    GO TO 80
130 DO 160 I=1,K
    M=(I-1)*IJ+1
    GO TO 170
140 IF (I-1) 150,160,150
150 W(1)=W(1)-BOB*W(M)
    W(K4)=W(K4)-BILL*W(M)
    A(1)=A(1)-BOB*A(I)
    A(N)=A(N)-BILL*A(I)
    C(1)=C(1)-BOB*C(I)
    C(N)=C(N)-BILL*C(I)
160 CONTINUE
    GO TO 550
170 MK=IOP(1)
    GO TO (180,210,260,310,260), MK
180 IF (I-1) 200,190,200
190 A(1)=-1.
    C(1)=0.
    GO TO 340
200 BOB=0.
    GO TO 340
210 IF (I-1) 230,220,230
220 A(1)=-1.
    C(1)=0.
    W(1)=0.
    GO TO 340
230 IF (I-2) 240,240,250
240 BOB=-C1
    GO TO 340
250 BOB=0.
    GO TO 340
260 IF (I-1) 280,270,280
270 A(1)=-(X(2)-X(1))/3.
    C(1)=0.
    W(1)=-C1+(F(IJ+1)-F(1))/(X(2)-X(1))
    GO TO 340
280 IF (I-2) 290,290,300
290 BOB=(X(2)-X(1))/6.
    GO TO 340
300 BOB=0.
    GO TO 340
310 IF (I-1) 330,320,330
320 A(1)=-1.
    C(1)=1.
    W(1)=0.
    GO TO 340
330 BOB=0.
340 ML=IOP(2)
    GO TO (350,380,430,480,430 , ML
```

358 184

```
  350 IF (I-1) 370,360,370
  360 A(N)=0.
      C(N)=-1.
      GO TO 140
  370 BILL=0.
      GO TO 140
  380 IF (I-1) 400,390,400
  390 A(N)=0.
      C(N)=-1.
      W(K4)=0.
      GO TO 140
  400 IF (I-K) 420,410,420
  410 BILL=-C2
      GO TO 140
  420 BILL=0.
      GO TO 140
  430 IF (I-1) 450,440,450
  440 A(N)=0.
      C(N)=(X(N-1)-X(N))/3.
      W(K4)=C2-(F(K4)-F(K1))/(X(N)-X(N-1))
      GO TO 140
  450 IF (I-K) 470,460,470
  460 BILL=(X(N)-X(N-1))/6.
      GO TO 140
  470 BILL=0.
      GO TO 140
  480 IF (I-1) 500,490,500
  490 A(N)=0.
      C(N)=(X(N-1)+X(1)-X(N)-X(2))/3.
      W(K4)=(F(IJ+1)-F(1))/(X(2)-X(1))-(F(K4)-F(K1))/(X(N)-X(N-1))
      GO TO 140
  500 IF (I-2) 520,510,520
  510 BILL=(X(2)-X(1))/6.
      GO TO 140
  520 IF (I-K) 540,530,540
  530 BILL=(X(N)-X(N-1))/6.
      GO TO 140
  540 BILL=0.
      GO TO 140
  550 CON=A(1)*C(N)-C(1)*A(N)
      D1=-W(1)
      D2=-W(K4)
      W(1)=(D1*C(N)-C(1)*D2)/CON
      W(K4)=(A(1)*D2-D1*A(N))/CON
      DO 560 I=2,K
      M=(I-1)*IJ+1
  560 W(M)=W(M)+A(I)*W(1)+C(I)*W(K4)
      GO TO 580
  570 CALL LABRT (1,COMM,1)
  580 RETURN
      END
      SUBROUTINE SPL1D2 (N,X,F.W,IJ,Y,TAB)
C     WHERE N= NUMBER OF POINTS IN THE INTERPOLATION
C           X= ORIGIN OF TABLE OF THE INDEPENDENT VARIABLE
C           F= ORIGIN OF TABLE OF THE DEPENDENT VARIABLE
C           W= ORIGIN OF TABLE OF SECOND DERIVATIVES AS CALCULATED BY
C              SPL1D1
C           IJ= SPACING IN THE TABLES F AND W
C           Y= THE POINT AT WHICH INTERPOLATION IS DESIRED
C           TAB= AN ARRAY OF DIMENSION 3 WHICH CONTAINS THE FUNCTION
C                VALUE, FIRST DERIVATIVE, AND SECOND DERIVATIVE AT Y
C
      DIMENSION X(3), F(3), W(3), TAB(3)
C
C     LOCATE Y IN THE X TABLE
C
      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
```


358 185

```
      GO TO 50
   40 CALL SEARCH (Y,X,N,I,MFLAG)
   50 MI=(I-1)*IJ+1
      K1=MI+IJ
      FLK=X(I+1)-X(I)
C
C     CALCULATE F(Y)
C
      A=(W(MI)*(X(I+1)-Y)**3+W(K1)*(Y-X(I))**3)/(6.*FLK)
      B=(F(K1)/FLK-W(K1)*FLK/6.)*(Y-X(I))
      C=(F(MI)/FLK-FLK*W(MI)/6.)*(X(I+1)-Y)
      TAB(1)=A+B+C
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
C     CALCULATE THE SECOND DERIVATIVE AT Y
C
      TAB(3)=(W(MI)*(X(I+1)-Y)+W(K1)*(Y-X(I)))/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 THA
*                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   SX6     B7
          EQ      DONE12
 DONE1    SX6     B6                 ILO
          SX7     B1                 1
          NG      B5,DONE11
 DONE12   SA6     A5                 SAVE ILO
          SA7     A0                 MFLAG = 1
          SA6     B4                 NDX = ILO
          IFEQ    *F,2          CALLED BY FTN
SEARCH1   SA1     TEMP          RESTORE A0
          SA0     X1
          ENDIF
```


358 186

```
*
*     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.
*
 SEARCH   DATA    0
          IFEQ    *F,2          CALLED BY FTN
          SX6     A0
          SA6     TEMP
          SB1     1             B1=1
          SA3     X1            X3=X
          SA1     A1+B1
          SB2     X1
          SB2     B2-B1         B2 NOW REFERENCES XT FOR INDEXES
          SA2     B2+B1         X1=XT(1)
          SA1     A1+B1
          SA4     X1            X4=N
          SA1     A1+B1
          SB4     X1            B4=ADDR(NDX)
          SA1     A1+B1
          SA0     X1            A0 =ADDR(MFLAG)
          SB5     B1
          SA5     ILO
          ELSE                  CALLED BY FUN OR RUN
          SA3     B1                    X3 CONTAINS X
          SA4     B3                    X4 CONTAINS N
          SA0     B5                    STORE ADDRESS OF MFLAG
          SB1     1                     KEEP VALUE OF 1
          SA5     ILO                   PICK UP BEGINNING VALUE OF ILO
          SB2     B2-B1                 B2 NOW REFERENCES XT FOR INDEXES
          SB5     B1                    ISTEP = 1
          SA2     B2+B1                 X1 = XT(1)
          ENDIF
          SB3     X4                    SET N
          SA4     B2+X4                 X4 = XT(N)
          SX7     B1                    1
          SB7     X5+B1                 IHI = ILO+1
          SB6     X5                    SET ILO
          SA7     A5                    STORE ILO = 1
          SA1     B2+B7                 X1 = XT(IHI)
          IX4     X4-X2                 XT(N) - XT(1)
          SA2     B2+B6                 X2 = XT(ILO)
*
*     IF XT IS MONOTONIC DECREASING BRANCH TO SWITCH VALUES
*
          NG      X4,SWIT               YES, XT(N) LT XT(1)  SWITCH
*
*     CHECK STARTING POINTS TO SEE IF X IS IN AN INTERVAL OF XT(ILO)
*     AND XT(IHI).  MUST BE TO BEGIN ACTUAL BINARY SECTION.
*
          GE      B7,B3,IHI2BIG         OPPS IHI IS GT N
*
 DECR     IX7     X3-X1                 X-XT(IHI)
          IX6     X3-X2                 X-XT(ILO)
          ZR      X7,DONE07
          PL      X7,BIGXW              X GT XT(IHI)  MUST BRING INTERVAL UP
          ZR      X6,DONE0              X = XT(ILO)
          PL      X6,DONE1              XT(ILO) LT X LT XT(ILO+1)  SO DONE
*
*     X IS LESS THAN STARTING VALUE OF ILO THEREFORE MUST BRING
*     INTERVAL DOWN TO DO BINARY SEARCH.  MOVE INTERVAL DOWN, DOUBLING
*     STEP SIZE EACH TIME UNTIL XT(ILO) LT X LT XT(IHI).
*
 SMLXW    SB6     B6-B5         DECREASE ILO TO GET THIS STEP
          LT      B6,B1,ILO2SML         ILO IS ZERO OR LESS NOT VALID
          SA2     A2-B5         GET NEXT VLAUE
          SB7     B6+B5                 IHI = ILO
          SB5     B5+B5                 ISTEP = 2*ISTEP
          GT      B6,B3,ILO2BIG
          SX4     B3                    X4 = N
          SX7     B6-B5                 NEXT DECREASING ILO
```


358 187

```
         GT     B5,B6,ILO2SML       NEXT INCREASING WILL BE OUT
         IX7    X4-X7               N - NEXT STEP(DECREASING)
         IX4    X3-X2               X-XT(ILO)
         NG     X7,ILO2BIG
         NG     X4,SMLXW            INTERVAL NOT FOUND  MUST REPEAT
         EQ     BSRCH               INTERVAL FORMED PERFORM BINARY.
*
*     XT WAS MONOTONIC DECREASING
*
 SWIT    SA1    A2                  SWITCH VALUES IN X2 AND X1
         SB7    B6                  IHI = ILO
         SB6    B6+B1               ILO = IHI
         SB5    -B5                 ISTEP = -1
         SA2    A2+B1               SET X2 = XT(ILO)
         LE     B6,B3,DECR          IF STILL IN RANGE RETURN TO NORMAL CO
         SB6    B1                  SET IHI = 1
*
*
 ILO2BIG PL     X4,BSRCH            INTERVAL FOUND TYY BINARY
         SA1    B2+B1               X1 = XT(1)
         SB7    B6
         SA2    B2+B3               X2 = XT(N)
         SB6    B3                  SET ILO TO N
*
         SX7    B1                  1
         SX6    B1+B1               2
         SA7    B4                  STORE NDX=1
         SA6    A0                  STORE MFLAG = 2
         SA7    A5                  STORE ILO = 1
         SX7    B3                  ILO = N
*
         IX6    X1-X3               XT(1)-X
         IX4    X3-X2               X-XT(N)
         IFEQ   *F,2        CALLED BY FTN
         NG     X6,SEARCH1
         ELSE
         NG     X6,SEARCH           X NOT IN TABLE   ALL DONE
         ENDIF
         PL     X4,BSRCH            X IS BETWEEN XT(ILO) AND XT(IHI)
*
         SA7    B4                  STORE NDX = N
         SA7    A5
         IFEQ   *F,2        CALLED BY FTN
         EQ     SEARCH1
         ELSE
         EQ     SEARCH              ALL DONE
         ENDIF
*
 ILO2SML IX4    X3-X2               XT(ILO)-X
         SA2    B2+B1               X1 = XT(1)
         SX7    B1                  1
         SX6    B1+B1               2
         PL     X4,BSRCH            FOUND AN INTERVAL TRY BINARY
         SB6    B1                  ILO = 1
*
         SA7    A5                  STORE ILO = 1
         SA6    A0                  STORE MFLAG = 2
*
         IX4    X3-X2               X-XT(1)
         SA7    B4                  STORE NDX = 1
         IFEQ   *F,2        CALLED BY FTN
         NG     X4,SEARCH1
         ELSE
         NG     X4,SEARCH           X IS LT XT(1)    ALL DONE
         ENDIF
         EQ     BSRCH               INTERVAL SET  TRY BINARY
*
*
 IHI2BIG SA1    B2+B3               X1 = X(N)
         SB6    B3-B1               ILO = N-1
         SA2    A1-B1               XI = XT(IHI-ISTEP)
*
```


358 188

```
         SX7     B1                  1
         SX6     B1+B1               2
         SA7     A5                  STORE ILO = 1
         SA6     A0                  STORE MFLAG = 2
         SA7     B4                  STO"S 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,IHIGTN           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 IHI  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)
         SB6     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,IHILT1            NEXT DECREASING VALUE IS OUTP
         NG      X7,IHILT1           NEXT DECREASING VALUE IS OUT
         IX7     X6-X7               N - NEXT STEP(INCREASING)
         IX4     X1-X3               X-XT(IHI)
         NG      X7,IHIGTN           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+Y5               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 BOUNDRY 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)
```


358 189

```
          IX0     X4-X3                   XT(MID)-X
          IX5     X1-X6                   IHI-ILO
          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,DONE00               FOUND RIGHT ANSWER EXACTLY
          SX7     B1                      1
          SA6     B4                      STORE ANSWER IN NDX
          SA7     A0                      STORE 1 IN MFLAG
          SA6     A5                      STORE ANSWER IN ILO
          IFEQ    *F,2          CALLED BY FTN
          EQ      SEARCH1
          ELSE
          EQ      SEARCH                  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 ILO
          IFEQ    *F,2          CALLED BY FTN
          EQ      SEARCH1
          ELSE
          EQ      SEARCH                  ALL DONE
          ENDIF
 DONE00   SX6     B5                      ANSWER IN B5
          MX7     0                       SET TO 0
          SA6     B4                      NDX = ANSWER
          SA7     A0                      MFLAG = 0
          SA6     A5                      ILO = ANSWER
          IFEQ    *F,2          CALLED BY FTN
          EQ      SEARCH1
          ELSE
          EQ      SEARCH                  ALL DONE
          ENDIF
*
*
 DONE07   SX6     B7
          MX7     0
          SA6     B4
          SA7     A0
          SA6     A5
          IFEQ    *F,2          CALLED BY FTN
          EQ      SEARCH1
          ELSE
          EQ      SEARCH
          ENDIF
 IHIGTN   SA1     B2+B3                   X1 = XT(N)
          ZR      X4,DONE07
          PL      X4,BSRCH                FOUND INTERVAL
          SB6     B3                      ILO = N
*
          SX6     B3                      N
          SX7     B1+B1                   2
          SA6     A5                      STORE ILO = 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
          ELSE
          NG      X4,SEARCH               X OUT OF RANGE  ALL DONE
          ENDIF
          LT      B5,B0,BSRCH             MUST KEEP POINTERS RIGHT
```

```
         ZR      X4,DONE0            FOUND RIGHT ANSWER
         SB6     B7                  OR WONT ZERO IN ON RIGHT
         SB7     B3                  SECTION
         EQ      BSRCH               FOUND INTERVAL  USE BINARY
*
*
 IHILT1  IX4     X1-X3               X - XT(IHI)
         SA2     B2+B1               X4 = XT(1)
         ZR      X4,DONE07
         PL      X4,BSRCH            FOUND INTERVAL
         SB6     1                   ILO = 1
*
         SX7     B1                  1
         SX6     B1+B1               2
         SA7     A5                  STORE ILO = 1
         SA6     A0                  STORE MFLAG = 2
*
         IX4     X2-X3               XT(1)-X
         SA7     B4                  STORE NDX = 1
         IFEQ    *F,2         CALLED BY FTN
         NG      X4,SEARCH1
         ELSE
         NG      X4,SEARCH           X OUT OF RANGE   ALL DONE
         ENDIF
         GT      B5,B0,BSRCH         MONTONIC DECREASING INTERVAL
         ZR      X4,DONE0            FOUND RIGHT ANSWER
         SB6     B7                  INTERVAL MUST BE BACKWARDS
         SB7     B1                  IHI = 1
         EQ      BSRCH               FOUND INTERVAL TRY BINARY
         IFEQ    *F,2         CALLED BY FTN
TEMP     BSS     1
         ENDIF
         END
C     THE FOLLOWING SUBROUTINE IS A REPLACEMENT FOR THE
C     ROUTINE SEARCH FOR USE WITH MACHINES THAT DO
C     NOT COMPILE COMPASS.
C     SUBROUTINE SEARCH(X,XT,N,NDX,MFLAG)
C     SEARCH FINDS A GIVEN VALUE X IN A MONOTONIC SERIES
C     DIMENSION XT(1)
C     IF (XT(N)-XT(1)) 140,10,20
C  10 PRINT 260
C     CALL EXIT
C  20 IF (XT(N)-X) 40,30,50
C  30 NDX=N
C     MFLAG=0
C     RETURN
C  40 NDX=N
C     MFLAG=2
C     RETURN
C  50 IF (XT(1)-X) 80,60,70
C  60 NDX=1
C     MFLAG=0
C     RETURN
C  70 NDX=1
C     MFLAG=2
C     RETURN
C  80 NDX=N/2.0+0.5
C     M0=1
C     M2=N
C  90 IF (XT(NDX)-X) 110,100,130
C 100 MFLAG=0
C     RETURN
C 110 M0=NDX
C     NDX=(M2-M0)/2.0+M0
C     IF (NDX-M0) 90,120,90
C 120 MFLAG=1
C     RETURN
C 130 M2=NDX
C     NDX (M2-M0)/2.0+M0
C     IF (NDX-M0) 90,120,90
C 140 IF (X-XT(1)) 170,150,160
```


358 191

```
C 150 NDX=1
C     MFLAG=0
C     RETURN
C 160 NDX=1
C     MFLAG=2
C     RETURN
C 170 IF (X-XT(N)) 190,180,200
C 180 NDX=N
C     MFLAG=0
C     RETURN
C 190 NDX=N
C     MFLAG=2
C     RETURN
C 200 NDX=N/2.0+0.5
C     M0=1
C     M2=N
C 210 IF (XT(NDX)-X) 230,220,250
C 220 MFLAG=0
C     RETURN
C 230 M2=NDX
C     NDX=(M2-M0)/2.0+M0
C     IF (NDX-M0) 210,240,210
C 240 MFLAG=1
C     RETURN
C 250 M0=NDX
C     NDX=(M2-M0)/2.0+M0
C     IF (NDX-M0) 210,240,210
C
C 260 FORMAT (5X,*YOUR ARRAY IS NOT MONOTONIC*)
C     END
      SUBROUTINE SGECO(A,LDA,N,IPVT,RCOND,Z)
      INTEGER LDA,N,IPVT(1)
      REAL A(LDA,1),Z(1)
      REAL RCOND
C
C     SGECO FACTORS A REAL MATRIX BY GAUSSIAN ELIMINATION
C     AND ESTIMATES THE CONDITION OF THE MATRIX.
C
C     IF  RCOND  IS NOT NEEDED, SGEFA IS SLIGHTLY FASTER.
C     TO SOLVE  A*X = B , FOLLOW SGECO BY SGESL.
C     TO COMPUTE  INVERSE(A)*C , FOLLOW SGECO BY SGESL.
C     TO COMPUTE  DETERMINANT(A) , FOLLOW SGECO BY SGEDI.
C     TO COMPUTE  INVERSE(A) , FOLLOW SGECO BY SGEDI.
C
C     ON ENTRY
C
C        A       REAL(LDA, N)
C                THE MATRIX TO BE FACTORED.
C
C        LDA     INTEGER
C                THE LEADING DIMENSION OF THE ARRAY  A .
C
C        N       INTEGER
C                THE ORDER OF THE MATRIX  A .
C
C     ON RETURN
C
C        A       AN UPPER TRIANGULAR MATRIX AND THE MULTIPLIERS
C                WHICH WERE USED TO OBTAIN IT.
C                THE FACTORIZATION CAN BE WRITTEN  A = L*U  WHERE
C                L  IS A PRODUCT OF PERMUTATION AND UNIT LOWER
C                TRIANGULAR MATRICES AND  U  IS UPPER TRIANGULAR.
C
C        IPVT    INTEGER(N)
C                AN INTEGER VECTOR OF PIVOT INDICES.
C
C        RCOND   REAL
C                AN ESTIMATE OF THE RECIPROCAL CONDITION OF  A .
C                FOR THE SYSTEM  A*X = B , RELATIVE PERTURBATIONS
C                IN  A  AND  B  OF SIZE  EPSILON  MAY CAUSE
C                RELATIVE PERTURBATIONS IN  X  OF SIZE  EPSILON/RCOND .
```


358 192

```
C                IF  RCOND  IS SO SMALL THAT THE LOGICAL EXPRESSION
C                           1.0 + RCOND .EQ. 1.0
C                IS TRUE, THEN  A  MAY BE SINGULAR TO WORKING
C                PRECISION.  IN PARTICULAR,  RCOND  IS ZERO  IF
C                EXACT SINGULARITY IS DETECTED OR THE ESTIMATE
C                UNDERFLOWS.
C
C        Z       REAL(N)
C                A WORK VECTOR WHOSE CONTENTS ARE USUALLY UNIMPORTANT.
C                IF  A  IS CLOSE TO A SINGULAR MATRIX, THEN  Z  IS
C                AN APPROXIMATE NULL VECTOR IN THE SENSE THAT
C                NORM(A*Z) = RCOND*NORM(A)*NORM(Z) .
C
C     LINPACK. THIS VERSION DATED 07/14/77 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     LINPACK SGEFA
C     BLAS SAXPY,SDOT,SSCAL,SASUM
C     FORTRAN ABS,AMAX1,SIGN
C
C     INTERNAL VARIABLES
C
      REAL SDOT,EK,T,WK,WKM
      REAL ANORM,S,SASUM,SM,YNORM
      INTEGER INFO,J,K,KB,KP1,L
C
      REAL SIGN
C
C     COMPUTE 1-NORM OF A
C
      ANORM = 0.0E0
      DO 10 J = 1, N
         ANORM = AMAX1(ANORM,SASUM(N,A(1,J),1))
   10 CONTINUE
C
C     FACTOR
C
      CALL SGEFA(A,LDA,N,IPVT,INFO)
C
C     RCOND = 1/(NORM(A)*(ESTIMATE OF NORM(INVERSE(A)))) .
C     ESTIMATE = NORM(Z)/NORM(Y) WHERE  A*Z = Y  AND  TRANS(A)*Y = E .
C     TRANS(A)  IS THE TRANSPOSE OF A .  THE COMPONENTS OF  E  ARE
C     CHOSEN TO CAUSE MAXIMUM LOCAL GROWTH IN THE ELEMENTS OF W  WHERE
C     TRANS(U)*W = E .  THE VECTORS ARE FREQUENTLY RESCALED TO AVOID
C     OVERFLOW.
C
C     SOLVE TRANS(U)*W = E
C
      EK = 1.0E0
      DO 20 J = 1, N
         Z(J) = 0.0E0
   20 CONTINUE
      DO 100 K = 1, N
         IF (Z(K) .NE. 0.0E0) 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 SSCAL(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. 0.0E0) GO TO 40
            WK = WK/A(K,K)
            WKM = WKM/A(K,K)
         GO TO 50
   40    CONTINUE
            WK = 1.0E0
            WKM = 1.0E0
```


358 193

```
   50    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) + WK*A(K,J)
               S = S + ABS(Z(J))
   60       CONTINUE
            IF (S .GE. SM) GO TO 80
               T = WKM - WK
               WK = WKM
               DO 70 J = KP1, N
                  Z(J) = Z(J) + T*A(K,J)
   70          CONTINUE
   80       CONTINUE
   90    CONTINUE
         Z(K) = WK
  100 CONTINUE
      S = 1.0E0/SASUM(N,Z,1)
      CALL SSCAL(N,S,Z,1)
C
C     SOLVE TRANS(L)*Y = V
C
      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
  120 CONTINUE
      S = 1.0E0/SASUM(N,Z,1)
      CALL SSCAL(N,S,Z,1)
C
      YNORM = 1.0E0
C
C     SOLVE L*V = Y
C
      DO 140 K = 1, N
         L = IPVT(K)
         T   Z(L)
         [illegible] = Z(K)
         [illegible] = T
         [illegible] (K .LT. N) CALL SAXPY(N-K,T,A(K+1,K),1,Z(K+1),1)
         [illegible] (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)
      YNORM = S*YNORM
C
C     SOLVE  U*Z = V
C
      DO 160 KB = 1, N
         K = N + 1 - KB
         IF (ABS(Z(K)) .LE. ABS(A(K,K))) 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
```


358 194

```
C     MAKE ZNORM = 1.0
      S = 1.0E0/SASUM(N,Z,1)
      CALL SSCAL(N,S,Z,1)
      YNORM = S*YNORM
C
      IF (ANORM .NE. 0.0E0) RCOND = YNORM/ANORM
      IF (ANORM .EQ. 0.0E0) RCOND = 0.0E0
      RETURN
      END
      SUBROUTINE SGEFA(A,LDA,N,IPVT,INFO)
      INTEGER LDA,N,IPVT(1),INFO
      REAL A(LDA,1)
C
C     SGEFA FACTORS A REAL MATRIX BY GAUSSIAN ELIMINATION.
C
C     SGEFA IS USUALLY CALLED BY SGECO, BUT IT CAN BE CALLED
C     DIRECTLY WITH A SAVING IN TIME IF  RCOND  IS NOT NEEDED.
C     (TIME FOR SGECO) = (1 + 9/N)*(TIME FOR SGEFA) .
C
C     ON ENTRY
C
C        A       REAL(LDA, N)
C                THE MATRIX TO BE FACTORED.
C
C        LDA     INTEGER
C                THE LEADING DIMENSION OF THE ARRAY  A .
C
C        N       INTEGER
C                THE ORDER OF THE MATRIX  A .
C
C     ON RETURN
C
C        A       AN UPPER TRIANGULAR MATRIX AND THE MULTIPLIERS
C                WHICH WERE USED TO OBTAIN IT.
C                THE FACTORIZATION CAN BE WRITTEN  A = L*U  WHERE
C                L  IS A PRODUCT OF PERMUTATION AND UNIT LOWER
C                TRIANGULAR MATRICES AND  U  IS UPPER TRIANGULAR.
C
C        IPVT    INTEGER(N)
C                AN INTEGER VECTOR OF PIVOT INDICES.
C
C        INFO    INTEGER
C                = 0  NORMAL VALUE.
C                = K  IF  U(K,K) .EQ. 0.0 .  THIS IS NOT AN ERROR
C                     CONDITION FOR THIS SUBROUTINE, BUT IT DOES
C                     INDICATE THAT SGESL OR SGEDI WILL DIVIDE BY ZERO
C                     IF CALLED.  USE  RCOND  IN SGECO FOR A RELIABLE
C                     INDICATION OF SINGULARITY.
C
C     LINPACK. THIS VERSION DATED 07/14/77 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     BLAS SAXPY,SSCAL,ISAMAX
C
C     INTERNAL VARIABLES
C
      REAL T
      INTEGER ISAMAX,J,K,KP1,L,NM1
C
C
C     GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
      INFO = 0
      NM1 = N - 1
      IF (NM1 .LT. 1) GO TO 70
      DO 60 K = 1, NM1
         KP1 = K + 1
C
C        FIND L = PIVOT INDEX
```

358 195

```
C
         L = ISAMAX(N-K+1,A(K,K),1) + K - 1
         IPVT(K) = L
C
C        ZERO PIVOT IMPLIES THIS COLUMN ALREADY TRIANGULARIZED
C
         IF (A(L,K) .EQ. 0.0E0) GO TO 40
C
C           INTERCHANGE IF NECESSARY
C
            IF (L .EQ. K) GO TO 10
               T = A(L,K)
               A(L,K) = A(K,K)
               A(K,K) = T
   10       CONTINUE
C
C           COMPUTE MULTIPLIERS
C
            T = -1.0E0/A(K,K)
            CALL SSCAL(N-K,T,A(K+1,K),1)
C
C           ROW ELIMINATION WITH COLUMN INDEXING
C
            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
   20          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
   50    CONTINUE
   60 CONTINUE
   70 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(1),JOB
      REAL A(LDA,1),DET(1),WORK(1)
C
C     SGEDI COMPUTES THE DETERMINANT AND INVERSE OF A MATRIX
C     USING THE FACTORS COMPUTED BY SGECO OR SGEFA.
C
C     ON ENTRY
C
C        A       REAL(LDA, N)
C                THE OUTPUT FROM SGECO OR SGEFA.
C
C        LDA     INTEGER
C                THE LEADING DIMENSION OF THE ARRAY  A .
C
C        N       INTEGER
C                THE ORDER OF THE MATRIX  A .
C
C        IPVT    INTEGER(N)
C                THE PIVOT VECTOR FROM SGECO OR SGEFA.
C
C        WORK    REAL(N)
C                WORK VECTOR.  CONTENTS DESTROYED.
C
C        JOB     INTEGER
C                = 11   BOTH DETERMINANT AND INVERSE.
C                = 01   INVERSE ONLY.
C                = 10   DETERMINANT ONLY.
C
```

```
C     ON RETURN
C
C        A       INVERSE OF ORIGINAL MATRIX IF REQUESTED.
C                OTHERWISE UNCHANGED.
C
C        DET     REAL(2)
C                DETERMINANT OF ORIGINAL MATRIX IF REQUESTED.
C                OTHERWISE NOT REFERENCED.
C                DETERMINANT = DET(1) * 10.0**DET(2)
C                WITH  1.0 .LE. ABS(DET(1)) .LT. 10.0
C                OR  DET(1) .EQ. 0.0 .
C
C     ERROR CONDITION
C
C        A DIVISION BY ZERO WILL OCCUR IF THE INPUT FACTOR CONTAINS
C        A ZERO ON THE DIAGONAL AND THE INVERSE IS REQUESTED.
C        IT WILL NOT OCCUR IF THE SUBROUTINES ARE CALLED CORRECTLY
C        AND IF SGECO HAS SET RCOND .GT. 0.0 OR SGEFA HAS SET
C        INFO .EQ. 0 .
C
C     LINPACK. THIS VERSION DATED 07/14/77 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     BLAS SAXPY,SSCAL,SSWAP
C     FORTRAN ABS,MOD
C
C     INTERNAL VARIABLES
C
      REAL T
      REAL TEN
      INTEGER I,J,K,KB,KP1,L,NM1
C
C
C     COMPUTE DETERMINANT
C
      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)
C        ...EXIT
            IF (DET(1) .EQ. 0.0E0) GO TO 60
   10       IF (ABS(DET(1)) .GE. 1.0E0) GO TO 20
               DET(1) = TEN*DET(1)
               DET(2) = DET(2) - 1.0E0
            GO TO 10
   20       CONTINUE
   30       IF (ABS(DET(1)) .LT. TEN) GO TO 40
               DET(1) = DET(1)/TEN
               DET(2) = DET(2) + 1.0E0
            GO TO 30
   40       CONTINUE
   50    CONTINUE
   60    CONTINUE
   70 CONTINUE
C
C     COMPUTE INVERSE(U)
C
      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
               T = A(K,J)
```

```
               A(K,J) = 0.0E0
               CALL SAXPY(K,T,A(1,K),1,A(1,J),1)
   80       CONTINUE
   90       CONTINUE
  100    CONTINUE
C
C        FORM INVERSE(U)*INVERSE(L)
C
         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
  110       CONTINUE
            DO 120 J = KP1, N
               T = WORK(J)
               CALL SAXPY(N,T,A(1,J),1,A(1,K),1)
  120       CONTINUE
            L = IPVT(K)
            IF (L .NE. K) CALL SSWAP(N,A(1,K),1,A(1,L),1)
  130    CONTINUE
  140    CONTINUE
  150 CONTINUE
      RETURN
      END
      INTEGER FUNCTION ISAMAX (N,SX,INCX)
C
C     FINDS THE INDEX OF ELEMENT HAVING MAX. ABSOLUTE VALUE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1), SMAX
      INTEGER I, INCX, IX, N
C
      ISAMAX=0
      IF (N.LT.1) RETURN
      ISAMAX=1
      IF (N.EQ.1) RETURN
      IF (INCX.EQ.1) GO TO 30
C
C        CODE FOR INCREMENT NOT EQUAL TO 1
C
      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))
   10 IX=IX+INCX
   20 CONTINUE
      RETURN
C
C        CODE FOR INCREMENT EQUAL TO 1
C
   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)
C
C     TAKES THE SUM OF THE ABSOLUTE VALUES.
C     USES UNROLLED LOOPS FOR INCREMENT EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
```

```
      REAL SX(1), STEMP
      INTEGER I, INCX, M, MP1, N, NINCX
C
      SASUM=0.0E0
      STEMP=0.0E0
      IF (N.LE.0) RETURN
      IF (INCX.EQ.1) GO TO 20
C
C        CODE FOR INCREMENT NOT EQUAL TO 1
C
      NINCX=N*INCX
      DO 10 I=1,NINCX,INCX
      STEMP=STEMP+ABS(SX(I))
   10 CONTINUE
      SASUM=STEMP
      RETURN
C
C        CODE FOR INCREMENT EQUAL TO 1
C
C        CLEAN-UP LOOP
C
   20 M=MOD(N,6)
      IF (M.EQ.0) GO TO 40
      DO 30 I=1,M
      STEMP=STEMP+ABS(SX(I))
   30 CONTINUE
      IF (N.LT.6) GO TO 60
   40 MP1=M+1
      DO 50 I=MP1,N,6
      STEMP=STEMP+ABS(SX(I))+ABS(SX(I+1))+ABS(SX(I+2))+ABS(SX(I+3))+ABS
     1 (SX(I+4))+ABS(SX(I+5))
   50 CONTINUE
   60 SASUM=STEMP
      RETURN
      END
      SUBROUTINE SAXPY (N,SA,SX,INCX,SY,INCY)
C
C     CONSTANT TIMES A VECTOR PLUS A VECTOR.
C     USES UNROLLED LOOP FOR INCREMENTS EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1), SY(1), SA
      INTEGER I, INCX, INCY, IX, IY, M, MP1, N
C
      IF (N.LE.0) RETURN
      IF (SA.EQ.0.0) RETURN
      IF (INCX.EQ.1.AND.INCY.EQ.1) GO TO 20
C
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
C          NOT EQUAL TO 1
C
      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
   10 CONTINUE
      RETURN
C
C        CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C        CLEAN-UP LOOP
C
   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
```


358 199

```
      IF (N.LT.4) RETURN
   40 MP1=M+1
      DO 50 I=MP1,N,4
      SY(I)=SY(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)
   50 CONTINUE
      RETURN
      END
      REAL FUNCTION SDOT (N,SX,INCX,SY,INCY)
C
C     FORMS THE DOT PRODUCT OF TWO VECTORS.
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1), SY(1), STEMP
      INTEGER I, INCX, INCY, IX, IY, M, MP1, N
C
      STEMP=0.0E0
      SDOT=0.0E0
      IF (N.LE.0) RETURN
      IF (INCX.EQ.1.AND.INCY.EQ.1) GO TO 20
C
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
C          NOT EQUAL TO 1
C
      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=STEMP+SX(IX)*SY(IY)
      IX=IX+INCX
      IY=IY+INCY
   10 CONTINUE
      SDOT=STEMP
      RETURN
C
C        CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C        CLEAN-UP LOOP
C
   20 M=MOD(N,5)
      IF (M.EQ.0) GO TO 40
      DO 30 I=1,M
      STEMP=STEMP+SX(I)*SY(I)
   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)+SX(I+1)*SY(I+1)+SX(I+2)*SY(I+2)+SX(I+3)*SY
     1 (I+3)+SX(I+4)*SY(I+4)
   50 CONTINUE
   60 SDOT=STEMP
      RETURN
      END
      SUBROUTINE SSCAL (N,SA,SX,INCX)
C
C     SCALES A VECTOR BY A CONSTANT.
C     USES UNROLLED LOOPS FOR INCREMENT EQUAL TO 1.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SA, SX(1)
      INTEGER I, INCX, M, MP1, N, NINCX
C
      IF (N.LE.0) RETURN
      IF (INCX.EQ.1) GO TO 20
C
C        CODE FOR INCREMENT NOT EQUAL TO 1
C
```

```
      NINCX=N*INCX
      DO 10 I=1,NINCX,INCX
      SX(I)=SA*SX(I)
   10 CONTINUE
      RETURN
C
C        CODE FOR INCREMENT EQUAL TO 1
C
C        CLEAN-UP LOOP
C
   20 M=MOD(N,5)
      IF (M.EQ.0) GO TO 40
      DO 30 I=1,M
      SX(I)=SA*SX(I)
   30 CONTINUE
      IF (N.LT.5) RETURN
   40 MP1=M+1
      DO 50 I=MP1,N,5
      SX(I)=SA*SX(I)
      SX(I+1)=SA*SX(I+1)
      SX(I+2)=SA*SX(I+2)
      SX(I+3)=SA*SX(I+3)
      SX(I+4)=SA*SX(I+4)
   50 CONTINUE
      RETURN
      END
      SUBROUTINE SSWAP (N,SX,INCX,SY,INCY)
C
C     INTERCHANGES TWO VECTORS.
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO 1.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      REAL SX(1), SY(1), STEMP
      INTEGER I, INCX, INCY, IX, IY, M, MP1, N
C
      IF (N.LE.0) RETURN
      IF (INCX.EQ.1.AND.INCY.EQ.1) GO TO 20
C
C       CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS NOT EQUAL
C         TO 1
C
      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
C
C       CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C       CLEAN-UP LOOP
C
   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
```

358 201

```
      STEMP=SX(I+1)
      SX(I+1)=SY(I+1)
      SY(I+1)=STEMP
      STEMP=SX(I+2)
      SX(I+2)=SY(I+2)
      SY(I+2)=STEMP
   50 CONTINUE
      RETURN
      END
```

APPENDIX C

DASH TEST PROBLEM (with Output)

Problem input:

```
    1    0
    A         1    0    0    0    0    0    0    0  8.0225E-07
    B         2    1    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
 108          1108             2108            33
    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-063
 104 0.0        3
    1.266E-05       0.0              0.0              6.330E-063
 104 0.0        3
    1.808E-06       0.0              0.0              9.042E-073
 104        0.03
    1.000E+10       5.000E+093
 102 0.0        3
 148 0.0        3
 148 0.0        3
```

358 202

Problem output:

DECAY CHAINS AND NUCLIDE RELATED DATA

NUCLIDE	ID	DECAY PARENT 1	DECAY PARENT 2	CAPTURE PARENT 1	CAPTURE PARENT 2	N-2N	N-ALPHA	N-P	DECAY CONSTANT
A	1	0	0	0	0	0	0	0	8.02250E-07
B	2	1	0	0	0	0	0	0	1.60450E-06

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

RADII 25

0.	.20833E+00	.41667E+00	.62500E+00	.83333E+00	.10417E+01
.12500E+01	.14583E+01	.16667E+01	.18750E+01	.20833E+01	.22917E+01
.25000E+01	.27083E+01	.29167E+01	.31250E+01	.33333E+01	.35417E+01
.37500E+01	.39583E+01	.41667E+01	.43750E+01	.45833E+01	.47917E+01
.50000E+01					

MATERIALS 24

1	1	1	1	1	1
1	1	2	2	2	2
2	2	2	2	3	3
3	3	3	3	3	3

TEMPERATURES 4

.10000E+04 .10000E+04 .10000E+04 .10000E+04

TEMP RADII 4

0. .20000E+01 .30000E+01 .50000E+01

DIJ-0 4

.54260E-05 0. 0. .27130E-05

AIJ 4

0. 0. 0. 0.

DIJ-0 4

.12660E-04 0. 0. .63300E-05

AIJ 4

0. 0. 0. 0.

DIJ-0 4

.18060E-05 0. 0. .90420E-06

AIJ 4

0. 0. 0. 0.

358 203

```
LEFT CONCEN        2
  .10000E+11    .50000E+10

RIGHT CONCEN       2
 0.            0.

INITIAL CONC      48
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.

SOURCE INPUT      48
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.
 0.            0.            0.            0.            0.            0.

    2*****************************************************************
    2        *        *        *        *        *        *        *
    2   1    *   1    *   1    *   1    *   1    *   1    *   1    *   1
    2        *        *        *        *        *        *        *
    2*****************************************************************
    0.000    .208     .417     .625     .833    1.042    1.250    1.458

    *****************************************************************
    *        *        *        *        *        *        *        *
    *   2    *   2    *   2    *   2    *   2    *   2    *   2    *   2
    *        *        *        *        *        *        *        *
    *****************************************************************
    1.667   1.875    2.083    2.292    2.500    2.708    2.917    3.125

    *****************************************************************2
    *        *        *        *        *        *        *        *   2
    *   3    *   3    *   3    *   3    *   3    *   3    *   3    *   3   2
    *        *        *        *        *        *        *        *   2
    *****************************************************************2
    3.333   3.542    3.750    3.958    4.167    4.375    4.583    4.792   5.000

  RBAR

   .10417E+00    .31250E+00    .52083E+00    .72917E+00    .93750E+00    .11458E+01
   .13542E+01    .15625E+01    .17708E+01    .19792E+01    .21875E+01    .23958E+01
   .26042E+01    .28125E+01    .30208E+01    .32292E+01    .34375E+01    .36458E+01
   .38542E+01    .40625E+01    .42708E+01    .44792E+01    .46875E+01    .48958E+01

  MESH TEMPERATURES

   .10000E+04    .10000E+04    .10000E+04    .10000E+04    .10000E+04    .10000E+04
   .10000E+04    .10000E+04    .10000E+04    .10000E+04    .10000E+04    .10000E+04
   .10000E+04    .10000E+04    .10000E+04    .10000E+04    .10000E+04    .10000E+04
   .10000E+04    .10000E+04    .10000E+04    .10000E+04    .10000E+04    .10000E+04
```


358 204

CELL CONCENTRATIONS AT 0. DAYS
(CONCENTRATIONS IN ATOMS/CC)

ISOTOPE	LEFT BOUNDARY	CELL 1	CELL 2	CELL 3	CELL 4
A	1.00000E+10	0.	0.	0.	0.
B	5.00000E+09	0.	0.	0.	0.

ISOTOPE	CELL 5	CELL 6	CELL 7	CELL 8	CELL 9
A	0.	0.	0.	0.	0.
B	0.	0.	0.	0.	0.

ISOTOPE	CELL 10	CELL 11	CELL 12	CELL 13	CELL 14
A	0.	0.	0.	0.	0.
B	0.	0.	0.	0.	0.

ISOTOPE	CELL 15	CELL 16	CELL 17	CELL 18	CELL 19
A	0.	0.	0.	0.	0.
B	0.	0.	0.	0.	0.

ISOTOPE	CELL 20	CELL 21	CELL 22	CELL 23	CELL 24
A	0.	0.	0.	0.	0.
B	0.	0.	0.	0.	0.

ISOTOPE	RIGHT BOUNDARY
A	0.
B	0.

NO. OF ATOMS = 0.

CELL CONCENTRATIONS AT 2.00000E+00 DAYS
(CONCENTRATIONS IN ATOMS/CC)

ISOTOPE	LEFT BOUNDARY	CELL 1	CELL 2	CELL 3	CELL 4
A	1.00000E+10	9.29795E+09	7.96461E+09	6.71483E+09	5.55947E+09
B	5.00000E+09	4.58334E+09	3.76847E+09	3.00574E+09	2.32140E+09

ISOTOPE	CELL 5	CELL 6	CELL 7	CELL 8	CELL 9
A	4.50618E+09	3.55868E+09	2.71648E+09	1.97513E+09	1.51218E+09
B	1.73153E+09	1.24114E+09	8.45226E+08	5.31159E+08	3.53010E+08

ISOTOPE	CELL 10	CELL 11	CELL 12	CELL 13	CELL 14
A	1.27135E+09	1.06544E+09	8.92274E+08	7.49647E+08	6.35396E+08
B	2.68117E+08	2.01869E+08	1.51143E+08	1.13164E+08	8.55697E+07

ISOTOPE	CELL 15	CELL 16	CELL 17	CELL 18	CELL 19
A	5.47490E+08	4.84093E+08	3.22124E+08	1.62196E+08	7.73686E+07
B	6.64433E+07	5.43220E+07	2.97671E+07	1.18012E+07	4.56331E+06

ISOTOPE	CELL 20	CELL 21	CELL 22	CELL 23	CELL 24
A	3.49964E+07	1.50226E+07	6.10502E+06	2.28116E+06	5.71744E+05
B	1.72444E+06	6.35629E+05	2.27180E+05	7.65217E+04	1.79805E+04

ISOTOPE	RIGHT BOUNDARY
A	0.
B	0.

NO. OF ATOMS = 1.446715010794452E+10

CELL CONCENTRATIONS AT 4.00000E+00 DAYS
(CONCENTRATIONS IN ATOMS/CC)

ISOTOPE	LEFT BOUNDARY	CELL 1	CELL 2	CELL 3	CELL 4
A	1.00000E+10	9.42778E+09	8.34731E+09	7.33075E+09	6.37816E+09
B	5.00000E+09	4.70421E+09	4.11867E+09	3.55015E+09	3.00805E+09

ISOTOPE	CELL 5	CELL 6	CELL 7	CELL 8	CELL 9
A	5.48958E+09	4.66485E+09	3.90357E+09	3.20501E+09	2.75003E+09
B	2.49978E+09	2.03055E+09	1.60328E+09	1.21885E+09	9.74220E+08

ISOTOPE	CELL 10	CELL 11	CELL 12	CELL 13	CELL 14
A	2.50296E+09	2.28157E+09	2.08539E+09	1.91402E+09	1.76697E+09
B	8.44765E+08	7.32089E+08	6.35525E+08	5.54337E+08	4.87757E+08

ISOTOPE	CELL 15	CELL 16	CELL 17	CELL 18	CELL 19
A	1.64372E+09	1.54371E+09	1.23427E+09	8.34885E+08	5.49025E+08
B	4.35023E+08	3.95409E+08	2.86757E+08	1.67269E+08	9.49803E+07

ISOTOPE	CELL 20	CELL 21	CELL 22	CELL 23	CELL 24
A	3.50278E+08	2.15566E+08	1.25588E+08	6.46189E+07	1.98144E+07
B	5.26285E+07	2.84312E+07	1.47910E+07	6.97871E+06	2.03949E+06

ISOTOPE	RIGHT BOUNDARY
A	0.
B	0.

NO. OF ATOMS = 2.022416554881128E+10

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

ISOTOPE	LEFT BOUNDARY	CELL 1	CELL 2	CELL 3	CELL 4
A	1.00000E+10	9.48391E+09	8.51455E+09	7.60570E+09	6.75534E+09
B	5.00000E+09	4.75291E+09	4.26232E+09	3.78171E+09	3.31659E+09

ISOTOPE	CELL 5	CELL 6	CELL 7	CELL 8	CELL 9
A	5.96166E+09	5.22303E+09	4.53792E+09	3.90483E+09	3.48873E+09
B	2.87162E+09	2.45057E+09	2.05634E+09	1.69104E+09	1.45176E+09

ISOTOPE	CELL 10	CELL 11	CELL 12	CELL 13	CELL 14
A	3.26026E+09	3.05272E+09	2.86580E+09	2.69920E+09	2.55260E+09
B	1.32[illegible]41E+09	1.20432E+09	1.10046E+09	1.00972E+09	9.31905E+08

ISOTOPE	CELL 15	CELL 16	CELL 17	CELL 18	CELL 19
A	2.42566E+09	2.31805E+09	1.96336E+09	1.46521E+09	1.07141E+09
B	8.66799E+08	8.14129E+08	6.51964E+08	4.43919E+08	2.95522E+08

ISOTOPE	CELL 20	CELL 21	CELL 22	CELL 23	CELL 24
A	7.63947E+08	5.25345E+08	3.38961E+08	1.89046E+08	6.06463E+07
B	1.92098E+08	1.21154E+08	7.25530E+07	3.82733E+07	1.19153E+07

ISOTOPE	RIGHT BOUNDARY
A	0.
B	0.

NO. OF ATOMS = 2.432060341188196E+10

358 206

CELL CONCENTRATIONS AT 8.00000E+00 DAYS
(CONCENTRATIONS IN ATOMS/CC)

ISOTOPE	LEFT BOUNDARY	CELL 1	CELL 2	CELL 3	CELL 4
A	1.00000E+10	9.51607E+09	8.61655E+09	7.77410E+09	6.98778E+09
B	5.00000E+09	4.78213E+09	4.34920E+09	3.92396E+09	3.51057E+09

ISOTOPE	CELL 5	CELL 6	CELL 7	CELL 8	CELL 9
A	6.25491E+09	5.57308E+09	4.94004E+09	4.35369E+09	3.96681E+09
B	3.11256E+09	2.73290E+09	2.37399E+09	2.03777E+09	1.81489E+09

ISOTOPE	CELL 10	CELL 11	CELL 12	CELL 13	CELL 14
A	3.75326E+09	3.55795E+09	3.38056E+09	3.22076E+09	3.07825E+09
B	1.69186E+09	1.57968E+09	1.47843E+09	1.38814E+09	1.30877E+09

ISOTOPE	CELL 15	CELL 16	CELL 17	CELL 18	CELL 19
A	2.95271E+09	2.84382E+09	2.47363E+09	1.93189E+09	1.48164E+09
B	1.24026E+09	1.18248E+09	9.93692E+08	7.32038E+08	5.28611E+08

ISOTOPE	CELL 20	CELL 21	CELL 22	CELL 23	CELL 24
A	1.10836E+09	7.97676E+08	5.35402E+08	3.07533E+08	1.00231E+08
B	3.72667E+08	2.53864E+08	1.62631E+08	9.02763E+07	2.88938E+07

ISOTOPE	RIGHT BOUNDARY
A	0.
B	0.

NO. OF ATOMS = 2.732894842037402E+10

CELL CONCENTRATIONS AT 1.00000E+01 DAYS
(CONCENTRATIONS IN ATOMS/CC)

ISOTOPE	LEFT BOUNDARY	CELL 1	CELL 2	CELL 3	CELL 4
A	1.00000E+10	9.54047E+09	8.68348E+09	7.88473E+09	7.14074E+09
B	5.00000E+09	4.80266E+09	4.41044E+09	4.02483E+09	3.64930E+09

ISOTOPE	CELL 5	CELL 6	CELL 7	CELL 8	CELL 9
A	6.44833E+09	5.80458E+09	5.20681E+09	4.65250E+09	4.28595E+09
B	3.28681E+09	2.93978E+09	2.61023E+09	2.29977E+09	2.09257E+09

ISOTOPE	CELL 10	CELL 11	CELL 12	CELL 13	CELL 14
A	4.08296E+09	3.89654E+09	3.72633E+09	3.57197E+09	3.43314E+09
B	1.97730E+09	1.87121E+09	1.77439E+09	1.68690E+09	1.60874E+09

ISOTOPE	CELL 15	CELL 16	CELL 17	CELL 18	CELL 19
A	3.30950E+09	3.20077E+09	2.82418E+09	2.25954E+09	1.77606E+09
B	1.53988E+09	1.48025E+09	1.27831E+09	9.85283E+08	7.45147E+08

ISOTOPE	CELL 20	CELL 21	CELL 22	CELL 23	CELL 24
A	1.36081E+09	1.00111E+09	6.84443E+08	3.98429E+08	1.30772E+08
B	5.49730E+08	3.90608E+08	2.59383E+08	1.47840E+08	4.79880E+07

ISOTOPE	RIGHT BOUNDARY
A	0.
B	0.

NO. OF ATOMS = 2.953406154721277E+10

358 207

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

358 208

☆U.S. Government Printing Office: 1979 — 677-013/76

Available from
US Nuclear Regulatory Commission
Washington, DC 20555

Available from
National Technical Information Service
Springfield, VA 22161

Microfiche	$ 3.00	126-150	7.25	251-275	10.75	376-400	13.00	501-525	15.25
001-025	4.00	151-175	8.00	276-300	11.00	401-425	13.25	526-550	15.50
026-050	4.50	176-200	9.00	301-325	11.75	426-450	14.00	551-575	16.25
051-075	5.25	201-225	9.25	326-350	12.00	451-475	14.50	576-600	16.50
076-100	6.00	226-250	9.50	351-375	12.50	476-500	15.00	601-up	—[1]
101-125	6.50								

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

358 209