GULF GENERAL ATOMIC

Gulf-GA-A12021

FIPER-II, A NUMERICAL SOLUTION OF TWO-DIMENSIONAL DIFFUSION AND ADSORPTION PROBLEMS

by

J. Appel, B. Forutanpour, and B. Roos

Prepared under
Contract AT(04-3)-167
Project Agreement No. 17
for the
San Francisco Operations Office
U.S. Atomic Energy Commission

July 1972

GULF GENERAL ATOMIC COMPANY
P.O. BOX 608, SAN DIEGO, CALIFORNIA 92112

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

The FIPER-II computer code is described herein as it existed on April 1972. The code has been in continuous development for 2 years and in its presented form has been applied successfully by Gulf Oil Corporation to the kind of problems discussed later in this report. However, the development and improvement of the code are being continued, so that duplication of results (or even close agreement) between problems run with the code as published and the code as it existed either before or after this time is not necessarily to be expected.

Gulf has exercised due care in preparation but does not warrant the merchantability, accuracy, and completeness of the code or of its description contained herein. The complexity of this kind of program precludes any guarantee to that effect. Therefore, any user must make his own determination of the suitability of the code for any specific use, and of the validity of the information produced by use of the code.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

GULF GENERAL ATOMIC

Gulf-GA-A12021

FIPER-II, A NUMERICAL SOLUTION OF TWO-DIMENSIONAL DIFFUSION AND ADSORPTION PROBLEMS

by

J. Appel, B. Forutanpour, and B. Roos

Prepared under
Contract AT(04-3)-167
Project Agreement No. 17
for the
San Francisco Operations Office
U.S. Atomic Energy Commission

NOTICE

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

Gulf General Atomic Project 317

July 1972

GULF GENERAL ATOMIC COMPANY
P O BOX 608, SAN DIEGO, CALIFORNIA 92112

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED *leg*

100

CONTENTS

1.	PHYSICAL PROBLEM STATEMENT	1
2.	MATHEMATICAL PROBLEM STATEMENT	4
3.	DERIVATION OF THE FINITE DIFFERENCE EQUATIONS.	6
	3.1. Grid of Mesh Points and Domain of the Variables.	6
	3.2. Integration of the Diffusion Equations	8
	3.3. Boundary Condition	14
4.	THE SOLUTION OF THE FINITE DIFFERENCE EQUATIONS.	16
	4.1. Method of Solution	16
	4.2. Simplification of the Finite Difference Equations.	21
	4.3. Computation of Time Step	24
	4.3.1. Computation of Initial Time Step	24
	4.3.2. Successive Computation of Time Step and Release of Materials	27
5.	PROGRAM DESCRIPTION.	29
	5.1. General Description of FIPER-II.	29
	5.2. Description of Subprograms	42
	5.2.1. Group 1.	43
	5.2.2. Group 2.	43
	5.2.3. Group 3.	43
6.	INPUT DESCRIPTION.	57
	6.1. Input Data for a New Problem	57
	6.2. Input Preparation for Restarting a Problem	63
	6.3. Input Preparation for Plotting	64
	6.4. Information Input Cards.	65
7.	DESCRIPTION OF OUTPUT.	66
	7.1. Preliminary Output	66
	7.2. Basic Output	66
	7.3. Output for Plotting.	67
	7.4. Output for Restarting.	67

8.	COMPUTATION OF CONCENTRATIONS AND CURRENTS	69
8.1.	Subcase 1, $\phi = 1$	70
8.2.	Subcase 2, $\phi = 10$	71
8.2.	Subcase 3, $\phi = 0.1$	71
	REFERENCES.	96

FIGURES

1.	Representation of two-dimensional grid of mesh points over the region of existence of the boundary value problem.	7
2.	Representation of region R_{mn} over which Eq. 2.1 is integrated. . .	10
3.	Concentration profile $\log U$ versus r, z for $\phi = 1$ at time $t = 2.011 \times 10^3$ sec.	72
4.	Concentration profile $\log U$ versus r, z for $\phi = 1$ at time $t = 2.0086 \times 10^4$ sec	73
5.	Concentration profile $\log U$ versus r, z for $\phi = 1$ at time $t = 2.0027 \times 10^5$ sec	74
6.	Average concentration versus z for $\phi = 1$ at time $t = 2.011 \times 10^3$ sec.	75
7.	Average concentration versus z for $\phi = 1$ at time $t = 2.0086 \times 10^4$ sec.	76
8.	Average concentration versus z for $\phi = 1$ at time $t = 2.0027 \times 10^5$ sec.	77
9.	Release rate as a function of time for $\phi = 1$	78
10.	Release as a function of time for $\phi = 1$	79
11.	Concentration profile $\log U$ versus r, z for $\phi = 10$ at time $t = 2.011 \times 10^3$ sec.	80
12.	Concentration profile $\log U$ versus r, z for $\phi = 10$ at time $t = 2.0086 \times 10^4$ sec	81
13.	Concentration profile $\log U$ versus r, z for $\phi = 10$ at time $t = 2.0027 \times 10^5$ sec	82
14.	Average concentration versus z for $\phi = 10$ at time $t = 2.011 \times 10^3$ sec.	83
15.	Average concentration versus z for $\phi = 10$ at time $t = 2.0086 \times 10^4$ sec.	84
16.	Average concentration versus z for $\phi = 10$ at time $t = 2.0027 \times 10^5$ sec.	85
17.	Release rate as a function of time for $\phi = 10$	86
18.	Release as a function of time for $\phi = 10$	87
19.	Concentration profile $\log U$ versus r, z for $\phi = 0.1$ at time $t = 2.011 \times 10^3$ sec.	88

20.	Concentration profile $\log U$ versus r, z for $\phi = 0.1$ at time $t = 2.0086 \times 10^4$ sec	89
21.	Concentration profile $\log U$ versus r, z for $\phi = 0.1$ at time $t = 2.0027 \times 10^5$ sec	90
22.	Average concentration versus z for $\phi = 0.1$ at time $t = 2.011 \times 10^3$ sec.	91
23.	Average concentration versus z for $\phi = 0.1$ at time $t = 2.0086 \times 10^4$ sec.	92
24.	Average concentration versus z for $\phi = 0.1$ at time $t = 2.0027 \times 10^5$ sec.	93
25.	Release rate as a function of time for $\phi = 0.1$	94
26.	Release as a function of time for $\phi = 0.1$	95

TABLES

5.1.	List of Symbols.	30
5.2.	Listing of Special Routines.	54

Blank

1. PHYSICAL PROBLEM STATEMENT

The release of radioactive fission product metals (such as cesium and strontium) into the helium coolant of an HTGR system is of considerable interest from the standpoints of safety and maintenance. The HTGR fuel element is a hexagonal block of graphite containing vertical coolant and fuel holes. The fuel holes are filled with rods composed of coated fuel particles bonded together by a graphite matrix. The release mechanism for the metallic fission product nuclides consists of (1) release from the fuel particle source, (2) sorption and diffusion in the matrix material, (3) sorption and diffusion in the graphite, (4) evaporation through the boundary layer (between the helium coolant and the graphite surface), and (5) transport by helium to graphite and metal plateout surfaces. Prediction of the degree of release of metallic fission products requires accurate calculational methods and basic input information consisting of diffusion, adsorption, and fuel particle release data.

The materials into which the fission atoms penetrate have two physico-chemical properties which are of particular interest. The first is the diffusion coefficient; it is of the form

$$D = \ell^2 \omega \exp (-\langle E_d \rangle / kT) \quad , \quad (1.1)$$

where ℓ is the jump distance, ω is the vibrational frequency of the host material, and $\langle E_d \rangle$ is an average activation energy for diffusion. The materials should provide diffusion barriers; i.e., the activation energy E_d for an elementary jump process should be large (>2 eV). Such materials exist; for example, strontium diffusing in single crystalline graphite has an activation energy of the order of 10 eV. However, the real material, in this case reactor graphite, contains mechanical defects such as grain boundaries, along which a significant transfer of fission atoms can take place with a smaller activation energy.

The second property of interest is the adsorption behavior of fission atoms on internal surfaces such as grain boundaries and open or closed pores. Internal surfaces adsorb fission products to a degree that depends on the quantum mechanical interaction between a fission atom and the surface. The probability that a surface site with adsorption energy E_a is occupied is given by

$$\theta(E_a) = \frac{1}{1 + \left[e^{(\mu - E_a)/kT} / Z(E_a) \right]} \quad (1.2)$$

where μ is the chemical potential and $Z(E_a)$ is a partition function of the adatoms, taking into account their vibrational degrees of freedom. Each adatom can be thought of as an anharmonic oscillator bound in a potential well of depth E_a . The probability that a site is occupied with a fission atom increases with E_a (Eq. 1.2).

HTGR core materials such as graphite, silicon carbide, and actinide oxides have a broad spectrum of adsorption energies,

$$0 \leq E_a \leq E_a^{\max} \quad (1.3)$$

where the maximum adsorption energy $E_a^{\max}$ is of the order of several electron volts. Once the sites with large adsorption energies are occupied, there is a snowball effect in that the adatoms move between sites of small adsorption energies E_a with a rather large jump frequency as determined by E_d .

Since both phenomena, adsorption and diffusion, are thermally activated processes that are strongly concentration dependent, it is important to have accurate experimental data describing both phenomena and their interrelation. A qualitative theoretical understanding of these phenomena allows for their extrapolation to reactor conditions.

The knowledge of adsorption and diffusion parameters is a prerequisite for a numerical calculation of concentration profiles and release currents from the transport equation,

$$\frac{\partial U}{\partial t} = S - \lambda U - \text{div } \underset{\sim}{J} \quad . \quad (1.4)$$

The rate of change of the concentration U is determined by three terms:*

1. The source term S that gives the number of fission atoms created per unit volume and time
2. The radioactive decay term λ
3. The current term depending on the divergence of the diffusion current $\underset{\sim}{J}$

The quantities in Eq. 1.4 depend in general on three space coordinates $\underset{\sim}{r}(x,y,z)$ and on the time variable t . With the FORTRAN V program FIPER (see Ref. 1), Eq. 1.1 can be solved in one space dimension and as a function of time. In this report we present a FORTRAN V program for the solution of the transport equation in two space dimensions and as a function of time.

To make the diffusion problem a definitive one, we must state the interface and boundary conditions and the initial concentration profile. The interface conditions are given by the physical requirement that the chemical potential of the fission atoms and the corresponding current density is continuous across the interface. The boundary conditions can be of the form where either the concentration or its gradient vanishes at a boundary. A more general expression is

$$\alpha(t) \underset{\sim}{\nabla} U / \perp \text{ surface} + \beta(t) U = \gamma(t) \quad (1.5)$$

where the functions $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ are given. The initial condition, $t = 0$, is given by the concentration profile $U(\underset{\sim}{r}, t = 0)$.

*The concentration is denoted here as U , and not as c , because the FORTRAN language does not contain c .

2. MATHEMATICAL PROBLEM STATEMENT

In this report we present the solution of two-dimensional boundary value problems involving systems of linear diffusion equations of the form

$$\frac{\partial U(\vec{r}, t)}{\partial t} = S(\vec{r}, t) - \lambda U(\vec{r}, t) - \nabla \cdot \vec{J}_T(U, \vec{r}, t) \quad (2.1)$$

where the current $\vec{J}_T(U, \vec{r}, t)$ is

$$\vec{J}_T(U, \vec{r}, t) = -D(U, \vec{r}, t) \nabla U(\vec{r}, t) + \vec{V}(\vec{r}, t) U(\vec{r}, t) \quad (2.2)$$

and $U(\vec{r}, t)$ = volume concentration (10^{-6} g/cm³)
 $S(\vec{r}, t)$ = volume generated source (10^{-6} g/cm³/sec)
 $D(U, \vec{r}, t)$ = diffusion coefficient (cm²/sec)
 λ = reaction rate or decay constant (1/sec)
 $\vec{V}(\vec{r}, t)$ = transport coefficient (10^{-6} cm/sec).

Efforts were made to solve the problem for the r-z geometry. Therefore, Eqs. 2.1 and 2.2 for such a geometry can be written as follows:

$$\frac{\partial U(r, z, t)}{\partial t} = S(r, z, t) - \lambda U(r, z, t) - \frac{1}{r} \frac{\partial}{\partial r} [r J_r(r, z, t)] - \frac{\partial}{\partial z} J_z(r, z, t) \quad (2.3)$$

$$J_r(r, z, t) = -D(r, z, t) \frac{\partial}{\partial r} U(r, z, t) + V_r(r, z, t) U(r, z, t) \quad (2.4)$$

$$J_z(r, z, t) = -D(r, z, t) \frac{\partial}{\partial z} U(r, z, t) + V_z(r, z, t) U(r, z, t) \quad (2.5)$$

Furthermore, the solution of the system was made subject to the following conditions:

$$U^{--}(r, z, t) = \Phi^{--}(r, z) U^{++}(r, z, t) \quad (2.6)$$

$$U^{+-}(r, z, t) = \Phi^{+-}(r, z) U^{++}(r, z, t) \quad (2.7)$$

$$U^{-+}(r, z, t) = \Phi^{-+}(r, z) U^{++}(r, z, t) \quad (2.8)$$

$$J_r^{+-}(r, z, t) = J_r^{--}(r, z, t) \quad (2.9)$$

$$J_r^{-+}(r, z, t) = J_r^{++}(r, z, t) \quad (2.10)$$

$$J_z^{+-}(r, z, t) = J_z^{++}(r, z, t) \quad (2.11)$$

$$J_z^{-+}(r, z, t) = J_z^{--}(r, z, t) \quad (2.12)$$

The first superscript refers to the r and the second to the z variable defining $U^{+-}(r, z, t) \equiv U(r + \epsilon, z - \epsilon, t)$, where ϵ is an infinitesimal positive number.

Similar definitions are applied to the currents J_r and J_z . The foregoing conditions make the solution of such systems a unique task; i.e., the interfaces at points (r, z) are allowed provided the equilibrium conditions are preserved and the currents remain continuous at all points.

Finally, the boundary conditions are assumed to be of the form

$$\alpha(r, z, t) \frac{\partial U(r, z, t)}{\partial n} + \beta(r, z, t) U(r, z, t) = \gamma(r, z, t) \quad (2.13)$$

where n refers to the direction perpendicular to the boundary line.

3. DERIVATION OF THE FINITE DIFFERENCE EQUATIONS

3.1. GRID OF MESH POINTS AND DOMAIN OF THE VARIABLES

To obtain the equivalent finite difference formulation of the boundary value problems delineated in the preceding section, we cover the space-time region $V = (0 \leq r_a < r < r_b, 0 \leq z_c < z < z_d, t_o < t < t_{\max})$ by rectangular net of lattice points (r_m, z_n, t_k) , which are the points of intersection of the lines

$$r_m = r_o + 2 \sum_{i=1}^{m-1} \Delta r_i \quad m = 1, 2, 3, \dots, M \quad (3.1)$$

$$z_n = z_o + 2 \sum_{j=1}^{n-1} \Delta z_j \quad n = 1, 2, 3, \dots, N \quad (3.2)$$

$$t_k = t_o + \sum_{\ell=1}^k \Delta t_\ell \quad \ell = 1, 2, 3, \dots, k \quad (3.3)$$

$$\text{where } r_o = r_a - \Delta r_1, \quad (3.4)$$

$$z_o = z_c - \Delta z_1. \quad (3.5)$$

We superimpose the two-dimensional grid of mesh points (r_m, z_n) for $m = 1, 2, \dots, M$ and $n = 1, 2, \dots, N$ over the region of existence of the boundary value problem. The letters r and z denote the radial and axial coordinates. The (m, n) mesh point corresponds to the coordinates (r_m, z_n) . The distance between the points (r_m, z_n) and (r_{m+1}, z_n) is denoted by $2\Delta r_m$, and the distance between the points (r_m, z_n) and (r_m, z_{n+1}) is denoted by $2\Delta z_n$ (see Fig. 1). In this report, the notation $m+x$ is interchangeable with r_{m+x} , $n+x$ with z_{n+x} , and $k+x$ with t_{k+x} . Furthermore, we assume that all interfaces coincide with the mesh points, and for the ordinary mesh points containing no interface, the equilibrium constants are assumed to be 1.

Fig. 1. Representation of two-dimensional grid of mesh points over the region of existence of the boundary value problem

3.2. INTEGRATION OF THE DIFFUSION EQUATIONS

To integrate Eq. 2.1 over the region R_{mn} , defined by $(r_{m-\frac{1}{2}} < r_m < r_{m+\frac{1}{2}}, z_{n-\frac{1}{2}} < z_n < z_{n+\frac{1}{2}})$, we subdivide R_{mn} into four subregions (see Fig. 1) and integrate over each subregion separately. For simplicity, we omit the factor 2π from all the integrals.

At a given time t_k the concentration integral will be

$$\iint_{R_{m,n}} U(r, z, t_k) r dr dz = H[m, n, \Phi(m, n)] U^{++}(m, n, k) \quad (3.6)$$

$$\text{where } H[m, n, \Phi(m, n)] = G(m) PH(m, n) + GG(m-1) ELH(m, n) \quad (3.7)$$

$$\text{and } G(m) = \Delta r_m (r_m + \frac{1}{2} \Delta r_m) \quad (3.8)$$

$$GG(m) = \Delta r_m (r_{m+1} - \frac{1}{2} \Delta r_m) \quad (3.9)$$

$$PH(m, n) = \Delta z_n + \Delta z_{n-1} \Phi^{+-}(m, n) \quad (3.10)$$

$$ELH(m, n) = \Delta z_n \Phi^{-+}(m, n) + \Delta z_{n-1} \Phi^{--}(m, n) \quad (3.11)$$

Similarly,

$$\iint_{R_{m,n}} \frac{\partial U(r, z, t_k)}{\partial t} r dr dz = H[m, n, \Phi(m, n)] \frac{\partial U^{++}(m, n, k)}{\partial t} \quad (3.12)$$

and

$$\iint_{R_{m,n}} S(r, z, t_k) r dr dz = H[m, n, \Phi(m, n) = 1] S(m, n, k) \quad (3.13)$$

The use of Green's formula gives

$$\begin{aligned}
& \iint_{R_{m,n}} \left\{ \frac{1}{r} \frac{\partial}{\partial r} \left[r J_r(r, z, t_k) \right] + \frac{\partial}{\partial z} J_z(r, z, t_k) \right\} r \, dr \, dz \\
&= \oint_C r J_r(r, z, t_k) \, dz - \oint_C r J_z(r, z, t_k) \, dr \\
&= \int_{z_{n+\frac{1}{2}}}^{z_n} r_{m-\frac{1}{2}} J_r(r_{m-\frac{1}{2}}, z, t_k) \, dz + \int_{z_n}^{z_{n-\frac{1}{2}}} r_{m-\frac{1}{2}} J_r(r_{m-\frac{1}{2}}, z, t_k) \, dz \\
&\quad - \int_{r_{m-\frac{1}{2}}}^{r_m} J_z(r, z_{n-\frac{1}{2}}, t_k) r \, dr - \int_r^{r_{m+\frac{1}{2}}} J_z(r, z_{n-\frac{1}{2}}, t_k) r \, dr \\
&\quad + \int_{z_{n-\frac{1}{2}}}^z r_{m+\frac{1}{2}} J_r(r_{m+\frac{1}{2}}, z, t_k) \, dz + \int_{z_n}^{z_{n+\frac{1}{2}}} r_{m+\frac{1}{2}} J_r(r_{m+\frac{1}{2}}, z, t_k) \, dz \\
&\quad - \int_{r_{m+\frac{1}{2}}}^{r_m} J_z(r, z_{n+\frac{1}{2}}, t_k) r \, dr - \int_r^{r_{m-\frac{1}{2}}} J_z(r, z_{n+\frac{1}{2}}, t_k) r \, dr \quad .
\end{aligned} \tag{3.14}$$

To integrate these integrals, we assume that the currents J_r or J_z are constant along the limits of integration and their values are computed at the points adjacent to the grid lines, as shown in Fig. 2.

Let us define the following geometrical parameters:

$$DZ(n) = \Delta z_n \tag{3.15}$$

$$RQ(m) = r_{m+\frac{1}{2}} \tag{3.16}$$

$$RP(m) = \frac{r_{m+\frac{1}{2}}}{2\Delta r_m} \tag{3.17}$$

Fig. 2. Representation of region R_{mn} over which Eq. 2.1 is integrated

$$HH(m) = G(m) + GG(m - 1) \quad (3.18)$$

$$VV(n) = DZ(n) + DZ(n - 1) \quad (3.19)$$

Note that from Eqs. 3.18 and 3.19 we obtain

$$H[m, n, \Phi(m, n) = 1] = HH(m) VV(n) \quad (3.20)$$

Using these notations, the numerical formulation equivalent to Eq. 3.14 is

$$\begin{aligned} & \int_C r J_r(r, z, t_k) dz - \int r J_z(r, z, t_k) dr \\ &= \sum_{i=1}^4 JDR(m, n, k)_i + \sum_{i=1}^4 JVR(m, n, k)_i + \sum_{i=1}^4 JDZ(m, n, k)_i + \sum_{i=1}^4 JVZ(m, n, k)_i \end{aligned} \quad (3.21)$$

$$\text{where } JDR(m, n, k)_1 = RP(m-1) DZ(n) D^{-+}(m-\frac{1}{2}, n, k) [\Phi^{-+}(m, n, k) U(m, n, k)$$

$$- U(m-1, n, k)] \quad , \quad (3.22)$$

$$JVR(m, n, k)_1 = -\frac{1}{2} RQ(m-1) DZ(n) V_r(m-\frac{1}{2}, n, k) [U(m-1, n, k)$$

$$+ \Phi^{-+}(m, n, k) U(m, n, k)] \quad , \quad (3.23)$$

$$JDR(m, n, k)_2 = RP(m-1) DZ(n-1) D^{--}(m-\frac{1}{2}, n, k) [\Phi^{--}(m, n, k) U(m, n, k)$$

$$- \Phi^{+-}(m-1, n, k) U(m-1, n, k)] \quad , \quad (3.24)$$

$$JVR(m, n, k)_2 = -\frac{1}{2} RQ(m-1) DZ(n-1) V_r(m-\frac{1}{2}, n, k) [\Phi^{+-}(m-1, n, k) U(m-1, n, k)$$

$$+ \Phi^{--}(m, n, k) U(m, n, k)] \quad , \quad (3.25)$$

$$\begin{aligned}
\text{JDZ}(m, n, k)_1 &= \frac{1}{2} \text{GG}(m-1) \text{D}^{--}(m, n-\frac{1}{2}, k) [\Phi^{--}(m, n, k) \text{U}(m, n, k) \\
&\quad - \Phi^{+-}(m, n-1, k) \text{U}(m, n-1, k)] / \text{DZ}(n-1)
\end{aligned} \tag{3.26}$$

$$\begin{aligned}
\text{JVZ}(m, n, k)_1 &= -\frac{1}{2} \text{GG}(m-1) \text{V}_z(m, n-\frac{1}{2}, k) [\Phi^{+-}(m, n-1, k) \text{U}(m, n-1, k) \\
&\quad + \Phi^{--}(m, n, k) \text{U}(m, n, k)]
\end{aligned} \tag{3.27}$$

$$\begin{aligned}
\text{JDZ}(m, n, k)_2 &= \\
&\frac{1}{2} \text{G}(m) \text{D}^{+-}(m, n-\frac{1}{2}, k) [\Phi^{+-}(m, n, k) \text{U}(m, n, k) - \text{U}(m, n-1, k)] / \text{DZ}(n-1)
\end{aligned} \tag{3.28}$$

$$\begin{aligned}
\text{JVZ}(m, n, k)_2 &= -\frac{1}{2} \text{G}(m) \text{V}_z(m, n-\frac{1}{2}, k) [\Phi^{+-}(m, n, k) \text{U}(m, n, k) \\
&\quad + \text{U}(m, n-1, k)]
\end{aligned} \tag{3.29}$$

$$\begin{aligned}
\text{JDR}(m, n, k)_3 &= -\text{RP}(m) \text{DZ}(n-1) \text{D}^{+-}(m+\frac{1}{2}, n, k) [\Phi^{--}(m+1, n, k) \text{U}(m+1, n, k) \\
&\quad - \Phi^{+-}(m, n, k) \text{U}(m, n, k)]
\end{aligned} \tag{3.30}$$

$$\begin{aligned}
\text{JVR}(m, n, k)_3 &= \frac{1}{2} \text{RQ}(m) \text{DZ}(n-1) \text{V}_r(m+\frac{1}{2}, n, k) [\Phi^{--}(m+1, n, k) \text{U}(m+1, n, k) \\
&\quad + \Phi^{+-}(m, n, k) \text{U}(m, n, k)]
\end{aligned} \tag{3.31}$$

$$\begin{aligned}
\text{JDR}(m, n, k)_4 &= -\text{RP}(m) \text{DZ}(n) \text{D}^{++}(m+\frac{1}{2}, n, k) [\Phi^{--}(m+1, n, k) \text{U}(m+1, n, k) \\
&\quad - \text{U}(m, n, k)]
\end{aligned} \tag{3.32}$$

$$\begin{aligned}
\text{JVR}(m, n, k)_4 &= \frac{1}{2} \text{RQ}(m) \text{DZ}(n) \text{V}_r(m+\frac{1}{2}, n, k) [\Phi^{--}(m+1, n, k) \text{U}(m+1, n, k) \\
&\quad + \text{U}(m, n, k)]
\end{aligned} \tag{3.33}$$

$$\begin{aligned} \text{JDZ}(m, n, k)_3 &= -\frac{1}{2} G(m) D^{++}(m, n+\frac{1}{2}, k) [\Phi^{+-}(m, n+1, k) U(m, n+1, k) \\ &\quad - U(m, n, k)] / \text{DZ}(n) \end{aligned} \quad (3.34)$$

$$\begin{aligned} \text{JVZ}(m, n, k)_3 &= \frac{1}{2} G(m) V_z(m, n+\frac{1}{2}, k) [\Phi^{+-}(m, n+1, k) U(m, n+1, k) \\ &\quad + U(m, n, k)] \end{aligned} \quad (3.35)$$

$$\begin{aligned} \text{JDZ}(m, n, k)_4 &= -\frac{1}{2} GG(m-1) D^{-+}(m, n+\frac{1}{2}, k) [\Phi^{--}(m, n+1, k) U(m, n+1, k) \\ &\quad - \Phi^{-+}(m, n, k) U(m, n, k)] / \text{DZ}(n) \end{aligned} \quad (3.36)$$

$$\begin{aligned} \text{JVZ}(m, n, k)_4 &= \frac{1}{2} GG(m-1) V_z(m, n+\frac{1}{2}, k) [\Phi^{--}(m, n+1, k) U(m, n+1, k) \\ &\quad + \Phi^{-+}(m, n, k) U(m, n, k)] \end{aligned} \quad (3.37)$$

Hence, the numerical representation of Eq. 2.1 will be

$$\begin{aligned} H[m, n, \Phi(m, n)] [U(m, n, t) - U(m, n, t-\Delta t)] / \Delta t &= HH(m) VV(n) S(m, n, t) \\ &- \lambda H[m, n, \Phi(m, n)] U(m, n, t) - \sum_{i=1}^4 \text{JDR}(m, n, t)_i \\ &- \sum_{i=1}^4 \text{JVR}(m, n, t)_i - \sum_{i=1}^4 \text{JDZ}(m, n, t)_i - \sum_{i=1}^4 \text{JVZ}(m, n, t)_i \end{aligned} \quad (3.38)$$

where $1 < m < M$ and $1 < n < N$.

Note that

$$\begin{aligned} 2\pi \sum_{i=1}^4 \text{JRD}(m, n, t)_i &= \text{net release rate in r-direction due to diffusion} \\ &\text{at time } t \end{aligned} \quad (3.39)$$

$$2\pi \sum_{i=1}^4 JDZ(m,n,t)_i = \text{net release rate in z-direction due to diffusion} \\ \text{at time t} \quad (3.40)$$

$$2\pi \sum_{i=1}^4 JVR(m,n,t)_i = \text{net release rate in r-direction due to} \\ \text{particle transportation at time t} \quad (3.41)$$

$$2\pi \sum_{i=1}^4 JVZ(m,n,t)_i = \text{net release rate in r-direction due to} \\ \text{particle transportation at time t} \quad (3.42)$$

3.3. BOUNDARY CONDITION

As mentioned previously, the boundary conditions are assumed to be of the general form expressed by Eq. 2.13. Without loss of generality, we assume that no vertical interface may exist at (r_1, z_n) or (r_M, z_n) and no horizontal interface at (r_m, z_1) and (r_m, z_N) . Therefore, the boundary conditions can be presented by the following finite difference equations:

$$-A^H(1,n,k) U(2,n,k) + B^H(1,n,k) U(1,n,k) \\ = D^H(1,n,k, k-\frac{1}{2}) \quad (3.43a)$$

$$B^H(M,n,k) U(M,n,k) - C^H(M,n,k) U(M-1,n,k) \\ = D^H(M,n,k, k-\frac{1}{2}) \quad (3.43b)$$

$$-A^V(m,1,k+\frac{1}{2}) U(m,2,k+\frac{1}{2}) + B^V(m,1,k+\frac{1}{2}) U(m,1,k+\frac{1}{2}) \\ = D^V(m,1,k, k+\frac{1}{2}) \quad (3.44)$$

$$\begin{aligned}
 & B^V(m, N, k + \frac{1}{2}) U(m, N, k + \frac{1}{2}) + C^V(m, N, k + \frac{1}{2}) U(m, N-1, k + \frac{1}{2}) \\
 & = D^V(m, N, k, k + \frac{1}{2}) \quad .
 \end{aligned}
 \tag{3.45}$$

4. THE SOLUTION OF THE FINITE DIFFERENCE EQUATIONS

4.1. METHOD OF SOLUTION

We use the alternating direction method, an implicit iterative technique devised by Peaceman and Rachford (Ref. 2), to solve the system of equations (3.38). Hence, at times

$$t_k = t_0 \sum_{i=1}^k \Delta t_i$$

we solve the sets of linear algebraic equations defined by Eq. 4.1, and those defined by Eq. 4.2 are solved at times $t_{k+\frac{1}{2}} = t_k + \frac{1}{2}\Delta t_k$. These sets are called horizontal sweep and vertical sweep equations, respectively.

The horizontal sweep equation is

$$\begin{aligned} & -A^H(m, n, k) U(m+1, n, k) + B^H(m, n, k) U(m, n, k) \\ & - C^H(m, n, k) U(m-1, n, k) = D^H(m, n, k, k-\frac{1}{2}) \end{aligned} \quad (4.1)$$

and the vertical sweep equation is

$$\begin{aligned} & -A^V(m, n, k+\frac{1}{2}) U(m, n+1, k+\frac{1}{2}) + B^V(m, n, k+\frac{1}{2}) U(m, n, k+\frac{1}{2}) \\ & - C^V(m, n, k+\frac{1}{2}) U(m, n-1, k+\frac{1}{2}) = D^V(m, n, k+\frac{1}{2}, k) \end{aligned} \quad (4.2)$$

where $A^H(m, n, k) = RP(m) DZ(n-1) D^{+-}(m+\frac{1}{2}, n, k) \Phi^{--}(m+1, n)$

$$\begin{aligned} & - \frac{1}{2} RQ(m) DZ(n-1) V_r(m+\frac{1}{2}, n, k) \Phi^{--}(m+1, n) \\ & + RP(m) DZ(n) D^{++}(m+\frac{1}{2}, n, k) \Phi^{-+}(m+1, n) \\ & - \frac{1}{2} RQ(m) DZ(n) V_r(m+\frac{1}{2}, n, k) \Phi^{-+}(m+1, n) \end{aligned} \quad (4.3)$$

$$\begin{aligned}
C^H(m, n, k) = & RP(m-1) DZ(n) D^{-+}(m-\frac{1}{2}, n, k) \\
& + \frac{1}{2} RQ(m-1) DZ(n) V_r(m-\frac{1}{2}, n, k) \\
& + RP(m-1) DZ(n-1) D^{--}(m-\frac{1}{2}, n, k) \Phi^{+-}(m-1, n) \\
& + \frac{1}{2} RQ(m-1) DZ(n-1) V_r(m-\frac{1}{2}, n, k) \Phi^{+-}(m-1, n) \quad (4.4)
\end{aligned}$$

$$\begin{aligned}
B^H(m, n, k) = & \frac{1}{2} H[m, n, \Phi(m, n)] / \Delta t_k + \frac{1}{2} \lambda H[m, n, \Phi(m, n)] \\
& + RP(m-1) DZ(n) D^{-+}(m-\frac{1}{2}, n, k) \Phi^{-+}(m, n) \\
& - \frac{1}{2} RQ(m-1) DZ(n) V_r(m-\frac{1}{2}, n, k) \Phi^{-+}(m, n) \\
& + RP(m-1) DZ(n-1) D^{--}(m-\frac{1}{2}, n, k) \Phi^{--}(m, n) \\
& - \frac{1}{2} RQ(m-1) DZ(n-1) V_r(m-\frac{1}{2}, n, k) \Phi^{--}(m, n) \\
& + RP(m) DZ(n-1) D^{+-}(m+\frac{1}{2}, n, k) \Phi^{+-}(m, n) \\
& + \frac{1}{2} RQ(m) DZ(n-1) V_r(m+\frac{1}{2}, n, k) \Phi^{+-}(m, n) \\
& + RP(m) DZ(n) D^{++}(m+\frac{1}{2}, n, k) \\
& + \frac{1}{2} RQ(m) DZ(n) V_r(m+\frac{1}{2}, n, k) \quad (4.5)
\end{aligned}$$

$$\begin{aligned}
D^H(m, n, k, k-\frac{1}{2}) = & HH(m) VV(n) S(m, n, k) \\
& + \frac{1}{2} H[m, n, \Phi(m, n)] U(m, n, k-\frac{1}{2}) / \Delta t_k \\
& - \frac{1}{2} \lambda H[m, n, \Phi(m, n)] U(m, n, k-\frac{1}{2}) / \Delta t_k \\
& - \sum_{i=1}^4 JDZ(m, n, k-\frac{1}{2})_i - \sum_{i=1}^4 JVZ(m, n, k-\frac{1}{2})_i \quad (4.6)
\end{aligned}$$

$$\begin{aligned}
\text{and } A^V(m, n, k + \frac{1}{2}) &= \frac{1}{2} G(m) D^{++}(m, n + \frac{1}{2}, k + \frac{1}{2}) \Phi^{+-}(m, n+1) / DZ(n) \\
&- \frac{1}{2} G(m) V_z(m, n + \frac{1}{2}, k + \frac{1}{2}) \Phi^{+-}(m, n+1) \\
&+ \frac{1}{2} GG(m-1) D^{-+}(m, n + \frac{1}{2}, k + \frac{1}{2}) \Phi^{--}(m, n+1) / DZ(n) \\
&- \frac{1}{2} GG(m-1) V_z(m, n + \frac{1}{2}, k + \frac{1}{2}) \Phi^{--}(m, n+1)
\end{aligned} \tag{4.7}$$

$$\begin{aligned}
C^V(m, n, k + \frac{1}{2}) &= \frac{1}{2} GG(m-1) D^{--}(m, n - \frac{1}{2}, k + \frac{1}{2}) \Phi^{-+}(m, n-1) / DZ(n-1) \\
&+ \frac{1}{2} GG(m-1) V_z(m, n - \frac{1}{2}, k + \frac{1}{2}) \Phi^{-+}(m, n-1) \\
&+ \frac{1}{2} G(m) D^{+-}(m, n - \frac{1}{2}, k + \frac{1}{2}) / DZ(n-1) \\
&+ \frac{1}{2} G(m) V_z(m, n - \frac{1}{2}, k + \frac{1}{2})
\end{aligned} \tag{4.8}$$

$$\begin{aligned}
B^V(m, n, k + \frac{1}{2}) &= \frac{1}{2} H[m, n, \Phi(m, n)] / \Delta t_k + \frac{1}{2} \lambda H[m, n, \Phi(m, n)] \\
&+ \frac{1}{2} GG(m-1) D^{--}(m, n - \frac{1}{2}, k + \frac{1}{2}) \Phi^{--}(m, n) / DZ(n-1) \\
&- \frac{1}{2} GG(m-1) V_z(m, n - \frac{1}{2}, k + \frac{1}{2}) \Phi^{--}(m, n) \\
&+ \frac{1}{2} G(m) D^{+-}(m, n - \frac{1}{2}, k + \frac{1}{2}) \Phi^{+-}(m, n) / DZ(n-1) \\
&- \frac{1}{2} G(m) V_z(m, n - \frac{1}{2}, k + \frac{1}{2}) \Phi^{+-}(m, n) \\
&+ \frac{1}{2} G(m) D^{++}(m, n + \frac{1}{2}, k + \frac{1}{2}) / DZ(n) \\
&+ \frac{1}{2} G(m) V_z(m, n + \frac{1}{2}, k + \frac{1}{2}) \\
&+ \frac{1}{2} GG(m-1) D^{-+}(m, n + \frac{1}{2}, k + \frac{1}{2}) \Phi^{-+}(m, n) / DZ(n) \\
&+ \frac{1}{2} GG(m-1) V_z(m, n + \frac{1}{2}, k + \frac{1}{2}) \Phi^{-+}(m, n)
\end{aligned} \tag{4.9}$$

$$\begin{aligned}
D^V(m, n, k + \frac{1}{2}, k) &= HH(m) VV(n) S(m, n, k + \frac{1}{2}) \\
&+ \frac{1}{2} H[m, n, \Phi(m, n)] U(m, n, k) \\
&- \frac{1}{2} \lambda H[m, n, \Phi(m, n)] U(m, n, k) \\
&- \sum_{i=1}^4 JDR(m, n, k)_i - \sum_{i=1}^4 JVR(m, n, k)_i \quad . \quad (4.10)
\end{aligned}$$

The system of algebraic equations defined by Eqs. 4.1 and 4.2 can be expressed by the following matrix notation:

$$A^{(n)} \vec{X}_h^{(n)}(t_k) = \vec{b}_h^{(n)}(t_k, t_{k-\frac{1}{2}}) \quad n=2, 3, \dots, N-1 \quad (4.11)$$

$$B^{(m)} \vec{X}_v^{(m)}(t_{k+\frac{1}{2}}) = \vec{b}_v^{(m)}(t_{k+\frac{1}{2}}, t_k) \quad m=2, 3, 4, \dots, M-1 \quad (4.12)$$

where matrices $A^{(n)}$ and $B^{(m)}$ are tridiagonal matrices of size $(M \times M)$ and $(N \times N)$, respectively, and

$$\begin{aligned}
\vec{X}_h^{(n)}(t_k) &= [U(1, n, k), U(2, n, k), \dots, U(M, n, k)]^T \\
&n=1, 2, 3, \dots, N \quad (4.13)
\end{aligned}$$

$$\begin{aligned}
\vec{X}_v^{(m)}(t_{k+\frac{1}{2}}) &= [U(m, 1, k + \frac{1}{2}), U(m, 2, k + \frac{1}{2}), \dots, U(m, N, k + \frac{1}{2})]^T \\
&m=1, 2, \dots, M \quad (4.14)
\end{aligned}$$

$$\vec{b}_h^{(n)}(t_k, t_{k-\frac{1}{2}}) = D^H(m, n, k, k - \frac{1}{2}) \quad n=1, 2, \dots, N \quad (4.15)$$

$$\vec{b}_v^{(m)}(t_{k+\frac{1}{2}}, t_k) = D^V(m, n, k, k - \frac{1}{2}) \quad m=1, 2, \dots, M \quad . \quad (4.16)$$

The diffusion and boundary equations are presently assumed to be uncoupled and linear. This makes the elements of matrices $A^{(n)}$ and $B^{(m)}$ independent of concentrations. Therefore, the solution of the systems of linear algebraic equations (4.1 and 4.2) can be obtained by matrix inversion. An efficient method for solving these systems is the line inversion method, in which one starts by defining the following equations for each value of n , where $n = 2, 3, 4, \dots, N-1$:

$$E^{(n)}(1, k) = \frac{A^H(1, n, k)}{B^H(1, n, k)} \quad (4.17)$$

$$F^{(n)}(1, k) = \frac{D^H(1, n, k, k - \frac{1}{2})}{B^H(1, n, k)} \quad (4.18)$$

and

$$E^{(n)}(m, k) = \frac{A^H(m, n, k)}{B^H(m, n, k) - C^H(m, n, k) E^{(n)}(m-1, k)} \quad 2 \leq m \leq M - 1 \quad (4.19)$$

$$F^{(n)}(m, k) = \frac{D^H(m, n, k - \frac{1}{2}, k) + C^H(m, n, k) F^{(n)}(m-1, k)}{B^H(m, n, k) - C^H(m, n, k) E^{(n)}(m-1, k)} \quad 2 \leq m \leq M \quad (4.20)$$

Then

$$U(M, n, k) = F^{(n)}(M, k) \quad (4.21)$$

and

$$U(m, n, k) = E^{(n)}(m, k) U(m+1, n, k) + F^{(n)}(m, k) \quad 1 \leq m \leq M - 1 \quad (4.22)$$

Similar notations can be derived for the systems defined by Eq. 4.2.

4.2. SIMPLIFICATION OF THE FINITE DIFFERENCE EQUATIONS

From Section 4.1 it is evident that the actual application of the Peaceman-Rachford iterative method requires a good deal of arithmetic. In a real physical problem, however, we do not have the imposed interface conditions at every mesh point. Thus, the finite difference equations can be simplified for such points. Considering the finite difference equation corresponding to a node (m,n) , we notice that the node is "connected" to its four neighboring nodes $(m-1,n)$, $(m+1,n)$, $(m,n-1)$, and $(m,n+1)$. That is, $U(m,n)$ is a function of concentration, equilibrium constants, and other items corresponding to these nodes. Three equilibrium constants are assigned to a node. Calculation of $U(m,n)$ requires nine such constants, three of which correspond to the node (m,n) and the remainder are contributed from the connecting nodes. Therefore, a major portion of the arithmetic is due to these constants. Special effort has been made to avoid unnecessary operations for the cases where the constants are equal to 1. In such cases, the equations in Section 4.1 may be further factorized and a greater reduction in the number of operations will be achieved. In respect to the conditions of equilibrium constants, one of the following cases is applicable to a node (m,n) , and its finite difference equation may be simplified:

1. No Interface. In this case, all nine equilibrium constants are equal to 1.
2. Horizontal Interface. In this case, equilibrium constants may differ from 1 only for the nodes $(m-1,n)$, (m,n) , and $(m+1,n)$.
3. Vertical Interface. In this case, equilibrium constants may differ from 1 only for the nodes $(m,n-1)$, (m,n) , and $(m,n+1)$.
4. Mixed Equilibrium Constant. This is called the general case and is considered for the nodes that do not meet any of the above conditions.

For each of the above special cases, a simplified version of the equations in Section 4.1 has been derived and programmed. As an example, the finite difference equations for the first case are:

$$A^H(m, n, k) = VV(n) [RP(m) D(m+\frac{1}{2}, n, k) - \frac{1}{2} RQ(m) V_r(m+\frac{1}{2}, n, k)] \quad (4.23)$$

$$C^H(m, n, k) = A^H(m-1, n, k) + VV(n) RQ(m-1) V_r(m-\frac{1}{2}, n, k) \quad (4.24)$$

$$\begin{aligned} B^H(m, n, k) = & A^H(m, n, k) + C^H(m, n, k) + (\frac{1}{2} \lambda + \frac{1}{\Delta t}) H(m, n, \Phi=1) \\ & + VV(n) [RQ(m) V_r(m+\frac{1}{2}, n, k) - RQ(m-\frac{1}{2}) V_r(m-\frac{1}{2}, n, k)] \end{aligned} \quad (4.25)$$

$$D^H(m, n, k) = DXH(m, n, k) + DUH(m, n, k) - DVH(m, n, k) \quad (4.26)$$

$$A^V(m, n, k+\frac{1}{2}) = \frac{1}{2} HH(m) [D(m, n+\frac{1}{2})/DZ(n) - V_z(m, n+\frac{1}{2}, k+\frac{1}{2})] \quad (4.27)$$

$$C^V(m, n, k+\frac{1}{2}) = A^V(m, n-1, k+\frac{1}{2}) + HH(m) V_z(m, n-\frac{1}{2}, k+\frac{1}{2}) \quad (4.28)$$

$$\begin{aligned} B^V(m, n, k+\frac{1}{2}) = & A^V(m, n, k+\frac{1}{2}) + C^V(m, n, k+\frac{1}{2}) + HH(m) VV(n) (\frac{1}{2} \lambda + \frac{1}{\Delta t_k}) \\ & + HH(m) [V_z(m, n+\frac{1}{2}, k+\frac{1}{2}) - V_z(m, n-\frac{1}{2}, k+\frac{1}{2})] \end{aligned} \quad (4.29)$$

$$D^V(m, n, k+\frac{1}{2}) = DXV(m, n, k+\frac{1}{2}) + DUV(m, n, k+\frac{1}{2}) - DVV(m, n, k+\frac{1}{2}) \quad (4.30)$$

where

$$DXH(m, n, k) = HH(m) VV(n) [S(m, n, k) + (-\frac{1}{2} \lambda + \frac{1}{\Delta t}) U(m, n, k)] \quad (4.31)$$

$$\begin{aligned} DUH(m, n, k) = & HH(m) \left\{ D(m, n+\frac{1}{2}, k) [U(m, n+1, k) - U(m, n, k)] / [2 DZ(n)] \right. \\ & \left. - D(m, n-\frac{1}{2}, k) [U(m, n, k) - U(m, n-1, k)] / [2 DZ(n-1)] \right\} \end{aligned} \quad (4.32)$$

$$\begin{aligned} DVH(m, n, k) = & \frac{1}{2} HH(m) \{ [U(m, n, k) + U(m, n+\frac{1}{2}, k)] V_z(m, n+\frac{1}{2}, k) \\ & - [U(m, n, k) + U(m, n-1, k)] V_z(m, n-\frac{1}{2}, k) \} \end{aligned} \quad (4.33)$$

$$DXV(m, n, k+\frac{1}{2}) = VV(n) HH(m) [S(m, n, k+\frac{1}{2}) + (-\frac{1}{2} \lambda + \frac{1}{\Delta t}) U(m, n, k)] \quad (4.34)$$

$$\begin{aligned} DUV(m, n, k+\frac{1}{2}) = & VV(n) \{ RP(m) D(m+\frac{1}{2}, n, k+\frac{1}{2}) [U(m+1, n, k+\frac{1}{2}) - U(m, n, k+\frac{1}{2})] \\ & - RP(m-1) D(m-\frac{1}{2}, n, k+\frac{1}{2}) [U(m, n, k+\frac{1}{2}) - U(m-1, n, k+\frac{1}{2})] \} \end{aligned} \quad (4.35)$$

$$\begin{aligned} DVV(m, n, k+\frac{1}{2}) = & \frac{1}{2} VV(n) \{ RQ(m) V_r(m+\frac{1}{2}, n, k+\frac{1}{2}) [U(m+1, n, k+\frac{1}{2}) \\ & + U(m, n, k+\frac{1}{2})] - RQ(m-1) V_r(m-\frac{1}{2}, n, k+\frac{1}{2}) \\ & [U(m, n, k+\frac{1}{2}) + U(m-1, n, k+\frac{1}{2})] \} \end{aligned} \quad (4.36)$$

Note that if the functions V_r and V_z are zero, then

$$A^H(m, n, k) = C^H(m+1, n, k) \quad (4.37)$$

$$A^V(m, n, k+\frac{1}{2}) = C^V(m+1, n, k+\frac{1}{2}) \quad (4.38)$$

$$B^H(m, n, k) > A^H(m, n, k) \quad (4.39)$$

$$B^V(m, n, k+\frac{1}{2}) > A^V(m, n, k+\frac{1}{2}) \quad (4.40)$$

Therefore, matrices $A^{(n)}$ and $B^{(m)}$ will be symmetric and diagonally dominant when the transport function V is zero and the equilibrium constants are equal to 1.

In the computer program written for our problem, the simplified versions of the equations in Section 4.1 have been programmed. The program tests the

conditions for each node. For each node, if it is possible, an appropriate simplified version of the finite difference equations is used.

4.3. COMPUTATION OF TIME STEP

The Peaceman-Rachford method has proved to be a stable method for the cases where the source function and the diffusion coefficients are constant and the coefficient matrices representing the system are symmetric. In our case, none of the above assumptions exist. As we have shown in Section 4.1, coefficient matrices $A^{(n)}$ and $B^{(m)}$ are symmetric when the normal equilibrium condition exists and the transport function is zero. Therefore, we have to devise a method for the selection of the time step. In practice, the selection of the proper time step and mesh size is of utmost importance for obtaining fairly accurate results within a reasonable computation time. Unfortunately, there is no easy way of selecting these parameters. For example, selection of small mesh size does not guarantee accurate results, and, even if it did, the problem might require the choice of such small time steps that the problem would become economically unfeasible to solve. In the next sections we will show a relationship between the time step and the mesh size.

4.3.1 Computation of Initial Time Step

Assume that the source function and the diffusion coefficients are constant. Equation 3.38 may be explicitly written in the form

$$U_i^{(k+1)} = U_i^{(k)} + \frac{\Delta t_i}{H_i} \left[S_i + \sum_{j=1}^5 B_{ij} U_i^{(k)} - \sum_{j=1}^5 A_{ij} U_i^{(k)} \right] \quad (4.41)$$

where the subscript i refers to a node (m,n) ; $j=1,2,3,4$ refers to the four neighboring nodes $(m-1,n)$, $(m+1,n)$, $(m,n-1)$, and $(m,n+1)$; and the last subscript refers to a fictitious node which takes care of the decay term. Also, the superscript k refers to the time t_k . We consider two cases:

Case 1: Concentration at node i is nondecreasing. Let U_i^* and U_j^* represent the steady-state concentration at nodes i and j , respectively. At the steady-state time we have

$$S_i + \sum B_{ij} U_i^* - \sum A_{ij} U_i^* = 0$$

or

$$U_i^* = \left(S_i + \sum B_{ij} U_j^* \right) / \sum A_{ij} .$$

However, $U_i^{(k+1)}$ is less than U_i^* . Therefore,

$$U_i^{(k)} + \frac{\Delta t_i}{H_i} \left[S_i + \sum B_{ij} U_j^{(k)} - \sum A_{ij} U_i^{(k)} \right] \leq \left[S_i + \sum B_{ij}^* U_j^* \right] / \sum A_{ij} . \quad (4.42)$$

By simplifying this equation and considering that the value of the second term in Eq. 4.42 is greater than zero, we will get

$$\Delta t \leq \frac{H_i}{A_{ij}} [1 + f_i(t)] \quad (4.43)$$

$$\text{where } f_i(t_k) = \sum B_{ij} (U_j^* - U_j^{(k)}) / \left[S_i + \sum B_{ij} U_j^{(k)} - \sum A_{ij} U_i^{(k)} \right] \quad (4.44)$$

$$H_i > 0$$

$$\sum A_{ij} > 0 .$$

Note that the denominator of Eq. 4.44 is always positive during the transient time. It is possible to choose the mesh size small enough so that U_j^* are also greater than $U_j^{(k)}$. In that case $f_i(t)$ is a positive-valued function.

Thus,

$$\Delta t_i \leq \frac{H_i}{\sum A_{ij}} . \quad (4.45)$$

Case 2: Concentration at node i is nonincreasing. In this case, U_i^* is less than $U_i^{(k)}$ and the second term in the right-hand side of Eq. 4.42 is negative during the transient time. Therefore, we will obtain Eq. 4.43 again. In this case, it is possible to choose small mesh sizes so that U_i^* is less than $U_i^{(k)}$ during the transient time. In that case, $f_i(t)$ becomes a positive-valued function and once again we obtain

$$\Delta t_i \leq \frac{H_i}{4 \sum_{j=1} A_{ij}} \quad . \quad (4.46)$$

The time step Δt_i refers to the node i . Choosing a time step not greater than Δt_i assures that the node i is stable during the integration process. Therefore, we compute the term $H_i / \sum A_{ij}$ for each node and choose Δt^* to be

$$\Delta t^* = \min_i \Delta t_i \quad . \quad (4.47)$$

Consider the case where we have the following conditions:

Constant source S
 Constant diffusion coefficient D
 Zero mass transport function
 Normal equilibrium condition
 Constant mesh size

In such a case we have:

$$H_i = H(m,n) = (2\Delta r)(2\Delta z) r_m$$

$$A_{i1} = (D \Delta z / \Delta r) r_{m-1/2}$$

$$A_{i2} = (D \Delta z / \Delta r) r_{m+1/2}$$

$$A_{i3} = (D \Delta r / \Delta z) r_m$$

$$A_{i4} = (D \Delta r / \Delta z) r_m$$

$$A_{i5} = 0 \quad .$$

Therefore,

$$\begin{aligned} \sum_{j=1}^5 A_{ij} &= 2 D \Delta z / \Delta r r_m + 2 D \Delta r / \Delta z r_m \\ &= [2 D (\Delta r^2 + \Delta z^2) / (\Delta r \Delta z)] r_m \end{aligned}$$

and

$$\Delta t_i \leq \frac{2 \Delta r^2 \Delta z^2}{D (\Delta r^2 + \Delta z^2)} . \quad (4.48)$$

If $\Delta r = \ell/2$ and $\Delta z = \ell/2$, then

$$\Delta t^* \leq \frac{\ell^2}{4 D} . \quad (4.49)$$

Interestingly enough, the steady-state time for a cylinder of length ℓ is in the order of $\ell^2/(4D)$.

In the computer program, Eq. 4.43 is used to compute the minimum time step Δt^* . Then a fraction of Δt^* (for example, $10^{-3} \Delta t^*$) is used as the initial time step.

4.3.2. Successive Computation of Time Step and Release of Materials

At the completion of the first iteration we have the value of each term in Eq. 4.41. We may now write this equation for the entire system:

$$H_b \frac{U_b^{(k+1)} - U_b^{(k)}}{\Delta t} \approx S_b^{(k)} + \sum B_{bj} U_j^{(k)} - \sum A_{bi} U_b^{(k)} . \quad (4.50)$$

Each side of the equation is computed separately. If the percentage of difference between the sides is less than a prescribed value $E_{\min}$ we double the time step. Similarly, if this difference is greater than a prescribed limit we use half the present time and recalculate the terms. Otherwise, the same time step is the next iteration cycle. Note that the calculation of the right-hand side of Eq. 4.50 contains the expression for the release of materials from the system. That is, the release rate $R(t)$ at time t_k is

$$R(t_k) = \sum_{j=1}^4 A_{bj} U_j^{(k)} - \sum_{j=1}^4 B_{bj} U_i^{(k)} . \quad (4.51)$$

Computation of this term is one of our main objectives. Thus, the procedure described in this section serves two purposes, computation of time step and release rate of particles.

5. PROGRAM DESCRIPTION

5.1. GENERAL DESCRIPTION OF FIPER-II

A FORTRAN V computer program was developed to solve the two-dimensional diffusion problems described in the preceding sections. The program presently solves diffusion problems in the r-z geometry and is designed so that it can easily be modified to solve the diffusion problems in other geometries.

FIPER-II is a modular program for solving the broadest possible range of two-dimensional diffusion problems. We have separated those parts of the program that pertain to the actual solution of the partial differential equations, the geometry, the mesh size, the time step, etc., and the parts of the program that contain the parameters such as the diffusion coefficients, the sources, the decay constants, and the boundary functions. The program can therefore be used to solve a wide range of diffusion problems for which the mathematical model describing the transport phenomena fits within the framework of the boundary value problems delineated in the preceding sections. Thus, heat transfer problems, fission product diffusion problems, and a host of other particle diffusion problems can be solved. To achieve the necessary flexibility, the program has been separated into two distinct parts, the MASTER code and the AUXILIARY code. The MASTER code contains the routines that are problem independent; i.e., they never need to be modified by the user. The AUXILIARY code is composed of the routines that are strongly dependent on the physical problem statement such as the diffusion coefficient subroutine, the source subroutine, the boundary function subroutine, etc.

There are two driver subprograms, MAIN1 and MAIN2, in FIPER-II. In a given run, only one of the two will be executed. MAIN1 is in charge of calling necessary subprograms to solve the problem. MAIN2 is executed only when a diffusion problem has been solved in one run and its output, such as concentration distribution or release of fission products, is to be plotted in the present run.

Some of the important technical features of FIPER-II are:

1. Computation of time step
2. Three-dimensional plotting
3. Restarting the problem capability
4. Plotting the results of a problem in a separate run
5. Modularity of the program

A list of symbols is presented in Table 5-1.

TABLE 5-1
LIST OF SYMBOLS

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
AFERMI	a	Fermi constant in $FX = 1 / \left[\exp\left(\frac{x - a}{b}\right) + 1 \right]$
ALPHA(ID12)	α_n	Array to store the coefficients α_{o_i}
ALPHAO(4)	α_{o_i}	Coefficient of $\frac{\partial u}{\partial n}$ in the boundary condition equations
BETA(ID12)	β_n	Array to store coefficients β_{o_i}
BETAO(4)	β_{o_i}	Coefficient of U in the boundary condition equations
BFERMI	b	The constant b in function FX
CAV(ID12)		Array used to save the average concentration in the r- or the z-direction
CRDNT		Indicates whether mesh points should be read in or computed in MAIN2
D(ID1, ID2)	D(m,n,t)	Diffusion coefficient corresponding to the point $(r_{m+\frac{1}{2}}, z_{n+\frac{1}{2}})$ at time t
DECAY	λ	$DECAY = \omega_\ell \lambda$
DECAYC		$DECAYC = \omega_c \lambda$
DELTAP(ID8)	Δt_p	Print intervals

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
DELTAT(ID7)	Δt_n	Array of different time steps to be used
DENUMH(ID1, ID5)		Denominator of Eq. 4.19 for the case of horizontal sweep
DENUMV(ID6, ID2)		Denominator of Eq. 4.19 for the case of vertical sweep
DETAIL		For a description of this input variable, see page 57
DEVICE		Device unit from which plotting data is to be read
DFLAG		Flag used to indicate under what conditions DIFUSN subroutine should be recalled for recalculation of diffusion coefficients D
DNAME(12)		Diffusion coefficients heading array
DOP		Option switch used to indicate which set of equations should be used for calculating diffusion coefficients D
DT	Δt_k	Time step at the k^{th} iteration
DTALL		$DTALL = \Delta t_k + \Delta t_{k+1}$
DTHALF	$\Delta t_k / 2$	Half of time step Δt_k
DTMAX	Δt_{max}	Maximum prescribed time step
DTMAXO		Factor used in computation of DTMAX
DTMIN	Δt_{min}	Minimum prescribed time step
DTMINO		Factor used in computation of DTMIN
DTNEXT	Δt_{k+1}	Time step to be used in the $k+1^{th}$ cycle
DTOLD	Δt_{k-1}	Time step used in the previous cycle $k-1$
DTOV	$2/\Delta t_k$	Inverse of half of time step
DTP	Δt_p	Current print interval
DTPLOT(ID13)		Array of different plot intervals to be used

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
DTPVAR		Factor used in decision-making of next Δt_p
DTRATO		Ratio of Δt_{k+1} to Δt_k
DTSTAR	Δt^*	Maximum time step prescribed by Section 4.3.1
DTVAR		Factor used in decision-making of next Δt_k
DZ(ID2)	Δz_n	Half the distance between z_n and z_{n+1}
DZOV(ID2)	Δz_n^{-1}	Inverse of Δz_n
EDOWN(ID1)	$EV(m, 1, k)$	The right-hand side of Eq. 4.17 for the case of vertical sweep
EH(ID1, ID5)	$EH(m, n, k)$	Contents of Eq. 4.19
ELEFT(ID2)	$EH(1, n, k)$	Contents of Eq. 4.17
EPSDBL		Error criterion for doubling time step
EPSHAP		Error criterion for reducing time step
ERROR		Percentage difference between left-hand and right-hand side of Eq. 3.38
EV(ID6, ID2)	$EV(m, n, k + \frac{1}{2})$	Contents of Eq. 4.19 for vertical sweep
FB(ID1, ID9)		Array used to save horizontal interface conditions
FDOWN(ID1)	$FV(m, 1, k + \frac{1}{2})$	Contents of Eq. 4.18 for vertical sweep
FEFLAG		Flag used to indicate under what conditions FEESUB routine should be recalled for recalculation of Φ function
FEOP		Option flag used to indicate which set of expressions should be used for calculating $\Phi(m, n)$ function
FL(ID10, ID2)		Array used to save vertical interface conditions
FLEFT(ID2)	$FH(1, n, k)$	The right-hand side of Eq. 4.18

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
FMM(ID10, ID9)	$\Phi^{--}(m, n)$	Matrix to save all $\Phi^{--}(m, n)$ having values different from 1
FMMM	$\Phi^{--}(m, n)$	Temporary location to save $\Phi^{--}(m, n)$
FMMMNP	$\Phi^{--}(m, n+1)$	Temporary location to save $\Phi^{--}(m, n+1)$
FMMMPN	$\Phi^{--}(m+1, n)$	Temporary location to save $\Phi^{--}(m+1, n)$
FMP(ID10, ID9)	$\Phi^{-+}(m, n)$	Matrix to save all $\Phi^{-+}(m, n)$ having values different from 1
FMPM	$\Phi^{-+}(m, n)$	Temporary location to save $\Phi^{-+}(m, n)$
FMPMN	$\Phi^{-+}(m, n-1)$	Temporary location to save $\Phi^{-+}(m, n-1)$
FMPMPN	$\Phi^{-+}(m+1, n)$	Temporary location to save $\Phi^{-+}(m+1, n)$
FNAME(12)		Heading array for equilibrium matrices
FPM(ID10, ID9)	$\Phi^{+-}(m, n)$	Matrix to save all $\Phi^{+-}(m, n)$ having values different from 1
FPMMN	$\Phi^{+-}(m-1, n)$	Temporary location to save $\Phi^{+-}(m-1, n)$
FPMN	$\Phi^{+-}(m, n)$	Temporary location to save $\Phi^{+-}(m, n)$
FPMNP	$\Phi^{+-}(m, n+1)$	Temporary location to save $\Phi^{+-}(m, n+1)$
FR(ID1)		Array containing results of Eq. 4.20 for horizontal sweep
FRATIO(8)		Temporary storage area for computing A_{ij} coefficients in Eq. 4.42
FZ(ID2)		Array containing results of Eq. 4.20 for vertical sweep
G(ID1)	$G(m)$	$VV(n) = \Delta r_m (r_m + \frac{1}{2} \Delta r_m)$
GAMMA(ID12)	γ_n	Array to store β_{oi}
GAMMAO(4)	γ_{oi}	Right-hand side of boundary condition equations
GG(ID1)	$GG(m)$	$GG(m) = \Delta r_m (r_{m+1} + \frac{1}{2} \Delta r_m)$
H(ID1)	$HH(m)$	$HH(m) = G(m) + GG(m-1)$
HBUFF(ID1, 2)		Temporary storage area for $U(m, n, k)$ where n is fixed

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
HFACE(ID2)		NS = HFACE(n) indicates that the NS th horizontal interface is at mesh point (m,n)
IENDX		Location containing symbol "\$\$\$\$\$"
IHFACE(ID2)		Array containing vertical mesh numbers at which horizontal interfaces exist
IID1		Variable equivalent to dimension parameter ID1; number of mesh points in r-direction MMAX must be less than ID1
IID2		Variable equivalent to dimension parameter ID2; number of mesh points in z-direction NMAX must be less than ID2
IID3		Variable equivalent to dimension parameter ID3; NUMVR must be less than ID3
IID4		Variable equivalent to dimension parameter ID4; NUMVZ must be less than ID4
IID5		Variable equivalent to dimension parameter ID5; presently, ID5 must be equal to ID2
IID6		Variable equivalent to dimension parameter ID6; presently, ID6 must be equal to ID1
IID7		Variable equivalent to dimension parameter ID7; NUMDT may not exceed ID7
IID8		Variable equivalent to dimension parameter ID8; NUMDTP may not exceed ID8
IID9		Variable equivalent to dimension parameter ID9; NFACE must be less than ID9
IID10		Variable equivalent to dimension parameter ID10; MFACE must be less than ID10
IID11		Variable equivalent to dimension parameter ID11; NCOUNT may not exceed ID11
IID12		Variable equivalent to dimension parameter ID12; ID12 = MAX0(ID1,ID2)
IID13		Variable equivalent to dimension parameter ID13; NUMPLT may not exceed ID13

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
INSIDE		Switch indicating locations of matrices computed in subroutine SETUP
IPLOT		Plot switch indicating when and how plotting should be done
IPRINT		Printing switch indicating format of output to be used to print concentrations, releases, etc.
IVFACE(ID1)		Array containing radial mesh numbers at which vertical interfaces exist
JLEFT		Current of particles at left boundary
JLOWER		Current of particles at lower boundary
JRIGHT		Current of particles at right boundary
JUPPER		Current of particles at upper boundary
KPRINT		Switch indicating in what form the volume integrals should be printed
LABEL(12,6)		Title cards; there may be up to 5 title cards to be read and printed
LABEL _i (12) i=1,2,...,9 A,B,C,D,X		Arrays used to plot or print labels or headings in various occasions
LAMDA	λ	Decay constant
LEFTFG		Flag to indicate under what conditions LEFTBC subroutine should be recalled for calculation of left boundary
LEFTOP		Option switch to indicate which set of equations in LEFTBC should be used for calculating left boundary
LOGITO		Variable to indicate the scaling option used for plotting concentration
LOWRFG		Flag to indicate under what condition LOWERB subroutine should be recalled for calculation of lower boundary
LOWROP		Option switch to indicate which set of equations in LOWERB should be used to calculate lower boundary

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
MFACE		Number of vertical interfaces
MMAX	M	Maximum number of mesh points in r-direction
MMAXM1	M-1	M-1
MORE		Option switch to indicate whether any information card is to be read by the subroutine INFORM
MVR(ID2)		$\ell = \text{MVR}(n)$ indicates that $V_r(m, \ell)$ is nonzero
MVZ(ID1)		$\ell = \text{MVZ}(m)$ indicates that $V_z(\ell, n)$ is nonzero
NAMEP(12)		Heading array used in plotting concentrations
NAMES(12)		Heading array used in MAIN2 for printing the release integrals
NBLANK		Location containing blank Hallerith
NCHANG		Number of different time steps used in MAIN1
NCOUNT		Number of data points saved per release integral
NDATE(4)		Array containing the exact date at which FIPER-II begins its computations
NDT		Indicates that NDT^{th} value of Δt_p is being used
NDTP		Indicates that NDTP^{th} value of Δt_p is being used
NDUMPT		Input variable indicating how program should be terminated
NFACE		Number of horizontal interfaces
NFSW		Switch indicating whether Φ functions should be printed or not, and in what format
NITER	k	Contains number of iterations performed

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
NLABEL		Number of title cards read in subroutine INOUT
NMAX	N	Number of mesh points in z-direction
NMAXM1	N-1	N-1
NOCHANG		Number of times that Δt could have been doubled but it was avoided due to $\Delta t_{\max}$ restriction
NOFACE		Switch indicating whether problem has any interface condition or not
NOMIT		If set to a value greater than zero, then release rates will not be printed by subroutine OUTPUT
NONLIN		Switch indicating whether problem is linear or not
NOTES(12)		Array used to read information cards in subroutine INFORM
NOVR		Switch indicating whether $V_r(m,n,k)$ is zero or not
NOVZ		Switch indicating whether $V_z(m,n,k)$ is zero or not
NPLOT		Frequency of saving release integral values to be plotted
NTAPE		Logical unit on which output of plottings is to be stored
NTIMER		Switch used by MAIN1 to call subroutine TIMER
NUMDT		Number of different input time steps loaded
NUMDTP		Number of different input print intervals loaded
NUMPLT		Number of plots of concentration versus r-z plane
NUMVR		Number of radial grid lines at which radial transport coefficient is different from zero

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
NUMVZ		Number of axial grid lines at which axial transport coefficient is different from zero
NWARN		At NWARN seconds before termination of a problem, FIPER-II saves necessary data for restarting the problem at a later time
NZONE		Indicates which set of equations in sub-routine LOCATE is used
OMEGAL	ω_{ℓ}	ω_{ℓ} is presently set to 1/2
OMEGAR(ID1)	$\omega_r(m)$	$\omega_r(m) = (r_m - r_{m-1}) / (r_{m+1} - r_{m-1})$
OMEGAT	ω_t	Relaxation factor; it is set to 1/2
OMEGAZ(ID2)	$\omega_z(n)$	$\omega_z(n) = (z_n - z_{n-1}) / (z_{n+1} - z_{n-1})$
PI	π	3.14159265
PI2	2π	Twice the value of π
PLOTME		Logical switch to indicate whether to plot results or not
POOL(ID12)		An array used as buffer area in FIPER-II for various purposes
PRINTT		Input variable in MAIN2 to indicate whether data should be printed or not
PTABLE(ID13)		Time table for plotting concentration versus r-z plane
PTIME		Indicates next printing time at each cycle
PTIMEX		Contains maximum time value up to which current Δt_p should be used
PTMAX(ID8)		Array containing maximum time values at which corresponding Δt_p should be used
Q		$Q = \text{SVINTG} - \text{LAMDA} * \text{UVINTG}$ at time t
QPJ		$QPJ = Q + \text{SUMJ}$ at time t_k
QPJOLD		QPJ at time t - Δt_k

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
R(ID1)	r_m	r-coordinate of mesh point (m,n)
RELZL(ID11)		Release rate at top of inner cylinder versus time
RELZR(ID11)		Release rate at top of outer cylinder versus time
RESTR		Switch indicating whether job is restart run or not
RH(ID1)		RH(m) = 0, no vertical interface at (m,n) = 3, vertical interface at (m,n)
RIGTFG		Flag to indicate under what conditions RIGHTB routine should be called
RIGTOP		Option switch to indicate which set of equations in RIGHTB should be used for calculating right-hand side boundary condition
RP(ID1)	RP(m)	$RP(m) = r_{m+\frac{1}{2}} / (2 \Delta r_m)$
RQ(ID1)	RQ(m)	$RQ(m) = r_{m+\frac{1}{2}}$
RR(ID1)	$\bar{r}_m$	Normalized r-coordinate of mesh point (m,n) for plotting purposes
RS		Scaling factor used to scale mesh sizes for plotting purposes
RV(ID1)		RV(m) = 1, no vertical interface at (m,n) = 2, vertical interface at (m+1,n) = 3, vertical interface at (m,n)
S(ID1,ID2)	S(m,n,k)	Source function at point (m,n) and time t_k
SFLAG		Flag to indicate under what conditions SOURCE routine should be called
SNAME(12)		Heading array used in printing matrix S(m,n)
SOP		Option switch used to indicate set of equations to be used in subroutine SOURCE
STAR(115)		Array containing symbols for printing purposes

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
SUMJ		Net release of particles
SVINTG		Source volume integral
TFINAL		Maximum problem time
TFLAG		Flag to indicate under what condition subroutine TEMPER should be called
TIME	t_k	Current time
TIMET(ID11)		Time table for which release integrals are saved and plotted
TIMEX		Maximum time up to which present time step is to be used
TIMEO	t_o	Initial time in diffusion problem
TMAX(ID7)		Time table for using different time steps
TNEXT	t_K	Time at which subroutine TEMPER should be called
TOP		Option switch to indicate set of equations to be used in TEMPER routine
TOTALC(ID11)		Total release of particles from top of cylinder versus time
TOTALL(ID11)		Total release of particles from inner cylinder versus time
TOTALR(ID11)		Total release of particles from outer cylinder versus time
U(ID1, ID2)	$U(m,n,k)$	Value of concentration at mesh point (m,n) and at time t_k
UB(10)		Temporary storage area for values of concentration at boundaries
UC(10)		Array to save coefficients of equation
UEPS		Factor used in decision-making of new time step
UNAME(12)		Heading array for concentration matrix

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
UOP		Option switch to indicate how initial concentrations should be computed or loaded
UPERFG		Flag to indicate under what conditions subroutine UPPERB should be called
UPEROP		Option switch to indicate set of equations to be used for calculating upper boundary conditions
UPFLAG		Flag to indicate under what conditions subroutine SETUPX should be called
US		Scaling factor used in scaling and plotting concentration values
USERS(ID12)		Array reserved and to be used by users
UVINTG		Volume concentration integral of system at time t
UVIOLD		Volume concentration integral of system at time $t - \Delta t$
V(ID2)	VV(n)	$V(n) = DZ(n) + DZ(n - 1)$
VARP		Factor used in decision-making of next print interval Δt_p
VART		Factor used in decision-making of next time step Δt_k
VBUFF(2, ID2)		Temporary storage area for $U(m, n, k + \frac{1}{2})$ where m is fixed
VECTOR(3)		Array indicating angle at which three-dimensional plotting should be done
VFACE(ID1)		MS = VFACE(m) indicates that MS th vertical interface is at mesh point (m, n)
VFLAG		Flag to indicate under what conditions subroutine VLOCTY should be called
VOP		Option switch to indicate set of equations to be used in subroutine VLOCTY to compute $V_r(m, n)$ and $V_z(m, n)$

<u>Variable Name</u>	<u>Symbol</u>	<u>Description</u>
VR(ID1, ID3)	$V_r(m, n)$	Radial mass transport function or coefficient
VRIGHT(ID2)	$B^H(M, n, k)$	Coefficient of $U(M, n, k)$ in Eq. 3.43b
VUPPER(ID1)	$B^V(m, N, k + \frac{1}{2})$	Coefficient of $U(m, N, k + \frac{1}{2})$ in Eq. 3.45
VZ(ID4, ID2)	$V_z(m, n)$	Axial mass transport function or coefficient
WH(ID1, ID5)	$C^H(m, n, k)$	Coefficient of $U(m-1, n, k)$ in Eq. 4.1
WRIGHT(ID2)	$C^H(M, n, k)$	Coefficient of $U(M-1, n, k)$ in Eq. 3.43b
WUPPER(ID1)	$C^V(m, N, k + \frac{1}{2})$	Coefficient of $U(m, N-1, k)$ in Eq. 3.45
WV(ID6, ID2)	$C^V(m, n, k + \frac{1}{2})$	Coefficient of $U(m, n-1, k)$ in Eq. 4.2
XRIGHT(ID2)	$D^H(M, n, k)$	Right-hand side of right boundary condition in Eq. 3.43b
XUPPER(ID1)	$D^V(m, N, k + \frac{1}{2})$	Right-hand side of upper boundary condition in Eq. 3.45
Z(ID2)	z_n	z-coordinate of mesh point (m, n)
ZH(ID2)		ZH(n) = 1, no horizontal interface at (m, n) = 2, vertical interface at (m, n+1) = 3, vertical interface at (m, n)
ZS		Scaling factor to scale axial mesh sizes for plotting purposes
ZV(ID2)		ZV(m) = 0, no vertical interface at (m, n) = 3, vertical interface at (m, n)
ZZ(ID2)	$\bar{z}_n$	Normalized z-coordinate of mesh point (m, n) for plotting purposes

5.2. DESCRIPTION OF SUBPROGRAMS

The FIPER-II program contains 61 subprograms, which may be classified into three groups, depending on the nature of their tasks.

5.2.1. Group 1

Group 1 consists of the master routines. The function of these subprograms is solving the finite difference equations and printing output. They are:

AVEREG	BEGINR	CONTRL
CURRENT	DATES	DRAW
DUMPER	DUMPIT	GEOMTY
INOUT	LOCATE	MAIN1
OUTPUT	PRINT	PRINTF
PRINTR	PUNCHU	RESIST
RESUME	SAVER	SETUPX
TEST	TIMER	

5.2.2. Group 2

Group 2 consists of the auxiliary routines. Some of the subprograms in the group may require modification or extra programming for solving diffusion problems. The 17 subprograms in this group are:

AUXIN	CALCDP	CALCDT
CORNER	DIFUSN	EXTRA
FEESUB	GRADNT	INFORM
LEFTBC	LOWERB	PAGLIM
RIGHTB	SOURCE	TEMPER
UPERB	VLOCTY	

5.2.3. Group 3

The subprograms used in plotting concentration values versus r-z plane or release integrals versus time are considered to form the group. They may be entirely deleted if one should decide not to plot the output results. The following 21 subroutines are in this group:

DLINV	DLOGV	GRIDER
GRIDV	LAB	LABELG
LABELN	MAIN2	PLOTERR
PLOTP	PLOTR	PLOTU
PLOTV	PLOT2D	PLOT3D
PLOT	RANGER	SCALED
STEREO	SURFAC	USCALE

A brief description of the subprograms is given alphabetically below. For more details, see the listing of the FIPER-II program.

AUXIN: Presently, this is a null routine. It has been set up so the user can read and print the input pertaining to the auxiliary routines, i.e., the physically dependent input, and also for initialization of variables used in such routines. A sample problem is solved in Section 8 and the auxiliary inputs are read in by the subprogram.

Called by: FIPER-II

AVEREG: Computes axial or radial average concentrations.

Called by: OUTPUT

PLOTERR

Calls: LOCATE

BEGINR: An entry name in the DUMPER subroutine. It reads the data saved by DUMPER in a previous run and restarts the problem.

Called by: FIPER-II

Calls: OUTPUT

CALCDP: To be utilized in the case where printing intervals are to be computed in some way beyond the present ability of FIPER-II.

Called by: MAIN1

RESUME

CALCDT: To be utilized in the case where time steps are to be computed in some way beyond the present ability of FIPER-II.

Called by: MAIN1

RESUME

CONTROL: A group of subroutines in FIPER-II, called special routines, compute the boundary conditions and the functions utilized in the diffusion equation. Since some of these functions may be constant during some time intervals, their computation at each iteration cycle should be avoided. A special flag has been assigned to each of these subroutines to indicate the conditions under which the subroutine corresponding to them should be referred to. For example, SFLAG is the flag name corresponding to the subroutine SOURCE. It assumes one of the following values:

SFLAG = 0, SOURCE is time independent.

= 1, SOURCE is temperature dependent.

= 2, SOURCE is concentration dependent.

Similar values and their interpretation are expected for the other flags.

Called by: MAIN1

RESUME

Calls:	TEMPER	LEFTBC
	DIFUSN	RIGHTB
	SOURCE	LOWERB
	FEESUB	UPPERB
	VLOCTY	SETUPX

CORNER: Computes the elements of diffusion coefficient matrix $D(m,n)$.

Called by: DIFUSN

CURRENT: Computes the release of particles, the source volume, and volume of the model.

Called by: MAIN1

Calls: RESIST

DATES: Saves the exact date and time at which it is called. The information is used for the problem identification and is printed on the top of the output pages.

Called by: FIPER-II

Calls: DATGET

GTIME

DIFUSN: Computes diffusion coefficients. Refer to the description of Special Routines in this section for more details.

Called by: CONTRL

Calls: CORNER

DRAW: As mentioned in Section 4.2, simplified versions of equations in 4.1 have been programmed. There are five versions of such equations associated with each $D^H(m,n,k)$ or $D^V(m,n,k+\frac{1}{2})$. For each of these matrices, such as $D^H(m,n,k)$, DRAW prints a matrix. The (m,n) element of the matrix indicates which of the versions will be used to compute $D^H(m,n,k)$.

Called by: GEOMTY

DUMPER: Prints and punches, or saves on peripheral unit, values of all variables required for restarting the job in a later run.

Called by: MAIN1

Calls: OUTPUT

PUNCHU

PRINTR

DUMPIT: Prints the contents of all geometry-dependent variables and arrays. This routine is called when the input variable DETAIL is set to a number greater than 3 (see page 57).

Called by: MAIN1

RESUME

Calls: PRINT

EXTRA: A specially designed routine for the problems whose model is made of two concentric cylinders. Computes the release of materials from the top of the cylinders.

Called by: MAIN1

FEESUB: Computes equilibrium functions. For more information refer to the description of Special Routines.

Called by: CONTRL

Calls: PRINTF

FIPER-II: The main program

Calls:	MAIN1	AUXIN
	MAIN2	RESUME
	INOUT	DATES
	BEGINR	

GEOMTY: Computes geometry-dependent arrays and variables; also decides which simplified versions of finite difference equations, if any, are to be used for mesh points.

Called by: INOUT

MAIN2

Calls: DRAW

GRADNT: Computes initial concentration for the sample problem in Section 8.

Called by: AUXIN

GRIDER: Translates and rotates the coordinates of mesh points. The subroutine is called when plotting of data requires some scaling to be done.

Called by: USCALE

PLOT2D

INFORM: Prints the input data in detail. If variable MORE in AUXIN is set to 2, then INFORM reads cards supplied by user to furnish more information regarding the problem. Such cards are to be the last batch of input cards to be read for a problem and should be followed by a card having \$\$\$\$\$\$ punched in its first 6 columns.

Called by: MAIN1

RESUME

INOUT: Reads and prints input cards related to the master routines.

Called by: FIPER-II

Calls: GEOMTY

LEFTBC: Computes the left boundary conditions. For more details refer to the description of Special Routines.

Called by: CONTRL

LOCATE: Computes the value of equilibrium functions utilized in the computation of concentration at node (m,n).

Called by: SETUPX

PRINTF

RESIST

AVEREG

LOWERB: Computes boundary conditions at the bottom of cylinder. For more details refer to the description of Special Routines.

Called by: CONTRL

MAIN1: Main program of FIPER-II for solution of a diffusion problem.

Called by: FIPER-II

Calls:	TEST	PAGLIM
	CONTRL	WARN
	DUMPIT	SAVER
	TIMER	OUTPUT
	CALCDT	DUMPER
	CALCDP	EXTRA
	INFORM	CURRENT

MAIN2: Main program in FIPER-II for the cases where the results of a problem previously solved are to be plotted in the current run.

Called by: FIPER-II

Calls: SETUP PLOTR
 GEOMTY PRINTR
 PRINT FINISH
 PLOTTER

OUTPUT: Prints the concentration at each mesh point; also prints release integrals and other terms related to the diffusion equations.

Called by: MAIN1 BEGINR
 DUMPER RESUME
 SAVER

Calls: PLOT
 AVEREG

PAGLIM: Presently, this is a dummy subroutine. It has been set up to let MAIN1 know how many pages of paper are left to be used for printing the output results. When the number is five or less, the restart data are saved and the execution is terminated.

Called by: MAIN1

PLOTTER: Initiates plotting of release integrals and concentration profiles.

Called by: MAIN2
 SAVER

Calls: AVEREG
 PLOT2D
 PLOTU
 PLOTV
 RANGER

PLOTP: Plots a function $f(t)$ versus t . Subprograms called by this routine are a part of GGA plot package.

Called by: PLOT2D

Calls: PLOT
DVR

PLOTR: Computes minimum and maximum values of release integrals during a time interval and calls PLOT2D routine for plotting the releases.

Called by: MAIN2

SAVER

Calls: PLOT2D

PLOTU: Plots concentration matrix U versus r - z plane.

Called by: PLOTU

Calls: USCALE
SCALED
PLOT3D

PLOTV: Initiates plotting of a function $f(t)$ versus t . Subprograms called by this routine are a part of GGA plot package.

Called by: PLOTU

Calls: ADF
FRAME
GRIDV
LAB
PLOTU

PLOT2D: Initiates plotting of an array of numbers versus another array. The elements of the arrays are scaled and plotted. Except for subroutine GRIDER, the subprograms are a part of GGA plot package.

Called by: PLOTU

PLOTR

Calls: GRIDER GRIDV
ADF PLOTU
FRAME LAB

PLOT3D: Initiates three-dimensional plotting of concentrations versus r-z plane. Subprograms called by this routine are a part of GGA plot package.
 Called by: PLOTU

PLOT: Plots the release of materials versus time on paper.
 Called by: OUTPUT

PRINT: This is a general-purpose printing subroutine, designed to print elements of a given matrix. The coordinates of the mesh points are also printed along the edges of the matrix.
 Called by: MAIN2
 PRINTF
 DUMPIT

PRINTF: Prints the equilibrium functions.
 Called by: FEESUB
 Calls: LOCATE
 PRINT

PRINTR: Prints the release integrals versus time.
 Called by: MAIN2
 SAVER
 DUMPER

PUNCHU: Punches the elements of a given matrix, column by column.
 Called by: DUMPER

RANGER: Translates, rotates, and scales discrete values of function $f(t)$ and the corresponding t .
 Called by: PLOTTER

RESIST: Computes coefficients A_{ij} , B_{ij} , S_i , and H_i expressed in Eq. 4.42.
 Called by: TIMER
 CURENT
 Calls: LOCATE

RESUME: An entry name in subroutine MAIN1. When a restart problem is to be solved, FIPER-II calls RESUME.

Called by: FIPER-II

Calls:	CONTRL	TIMER
	CALCDP	INFORM
	CALCDT	SAVER
	DUMPIT	OUTPUT

RIGHTB: Computes the right-hand side boundary conditions. For more details refer to the description of Special Routines.

Called by: CONTRL

SAVER: Punches, or saves on a peripheral unit, the last value of concentrations, release integrals, and other necessary parameters. Also plots these items when input variable IPLOT is set appropriately. Subroutines SETUP and FINISH called by this routine are a part of GGA plot package.

Called by: MAIN1

RESUME

Calls: OUTPUT

SETUP

PLOTR

PLOTERR

FINISH

PRINTR

SCALED: Scales the dimensions of a model problem for plotting.

Called by: PLOTU

SETUPX: Calculates the elements of coefficient matrices $A^{(n)}$ and $B^{(m)}$ in Section 4.1. Also computes the matrices $E^H_{(m,n,k)}$ and $E^V_{(m,n,k+\frac{1}{2})}$, for which expressions are also given in that section. Note that the computation of these matrices depends on the diffusion coefficient, the mass transfer function, and the time step. Therefore, CONTRL refers to this routine whenever one of the above variables assumes a new value.

Called by: CONTRL

Calls: LOCATE

SOURCE: Computes the source function S. For more details refer to the description of Special Routines.

Called by: CONTRL

Special

Routines: Table 5-2 is a listing of subprograms in this group. These routines have the same basic structure. For example, they all have two special variables called XOP and XFLAG. The subprograms are only called by the CONTRL subroutine and are part of the auxiliary code. They are designed to compute the boundary conditions or the functions used in the diffusion equation. Some of the functions or boundary conditions most often used in diffusion problems have been programmed in this routine. In some cases, however, the user has to incorporate his special functions. To use one of the several available functions in a subprogram, one has to assign a proper value to its XOP variable. For better understanding of these routines, consider the general structure of subroutine SOURCE:

```
SUBROUTINE SOURCE
.
.
.
GO TO (100,200,300),SOP

100 CONTINUE
SFLAG=...
.
.
.
S(m,n) = f1(m,n)
GO TO 1000
200 CONTINUE
SFLAG=...
.
.
.
S(m,n) = f2(m,n)
GO TO 1000
```

```

300  CONTINUE
      IF TIME . LT. 100) SFLAG=...
      IF TIME . GE. 100) SFLAG=...
      S(m,n) = f3(m,n)
1000 CONTINUE } calculation of
      .         } variables and other
      .         } parameters that are
      .         } a function of source
      RETURN
      END

```

When the switch SOP is set to i, the source function $f_i(m,n)$ will be used, where $i=1,2,3$. Note that variable XOP is an input variable, while XFLAG is not. The latter must be set to one of the values shown on page 45. Note also that SFLAG in the above example is set to a value depending on the properties of source functions. Similar techniques are expected to be practiced by the user.

TABLE 5-2
LISTING OF SPECIAL ROUTINES

<u>Subroutine</u>	<u>Name of XOP Variable</u>	<u>Name of XFLAG Variable</u>	<u>Computation Performed</u>
DIFUSN	DOP	DFLAG	Diffusion coefficients D
FEESUB	FEEOP	FEFLAG	Equilibrium coefficient ϕ
LEFTBC	LEFTOP	LEFTFG	Left boundary condition
LOWERB	LOWROP	LOWRFG	Lower boundary condition
RIGHTB	RIGTOP	RIGTFG	Right boundary condition
SOURCE	SOP	SFLAG	Source function S
TEMPER	TOP	TFLAG	Temperature T
UPPERB	UPEROP	UPERFG	Upper boundary condition
VLOCTY	VOP	VFLAG	Transport coefficients V_r and V_z

STEREO: Plots concentration values versus r-z plane. The routine is a modified version of the STEREO subprogram in GGA plot package. Subroutines called by STEREO are in that package.

Called by: PLOT3D

Calls: PLOTV

DVR

FRAME

SURFAC: Initiates plotting of a function $f(x,y)$ versus its independent variables x and y . Subprograms called by this routine are a part of GGA plot package.

TEMPER: Presently, this is a dummy routine. It is set up for the case where a function such as source or diffusion coefficients is temperature dependent. In that case, this subprogram computes temperature function and temperature-dependent variables. For more details refer to the description of Special Routines.

Called by: CONTRL

TEST: The required size of arrays for solving a problem is checked versus its actual size. If the number of reserved locations is not sufficient, an error message is given and the execution is terminated.

Called by: MAIN

TIMER: Computes the value of $\Delta t_{\min}$ discussed in Section 4.3.

Called by: MAIN1

RESUME

Calls: RESIST

UPERB: Computes the upper boundary condition. For more detail refer to the description of Special Routines.

Called by: CONTRL

USCALE: Scales concentration values for plotting.

Called by: PLOTU

Calls: GRIDER

VLOCTY: Computes the mass transport functions V_r and V_z . For more information refer to the description of Special Routines.

Called by: CONTRL

6. INPUT DESCRIPTION

As stated earlier, FIPER-II terminates and restarts computations on a particular problem and is also designed to plot the results of a problem in a separate run. Therefore, the input data for FIPER-II depends mainly on the nature of the task.

6.1. INPUT DATA FOR A NEW PROBLEM

The input data required by the master routines are listed below. All other data must be provided through the AUXIN subroutine and should be placed after the former set.

<u>Card</u>	<u>Symbol</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
0	RESTART	01-06	I6	RESTART = 0 New problem
				= 5 Restart problem. Read restart data via input cards
				= n Restart problem. Tape unit n supplies restart data
	NDUMPT	07-12	I6	NDUMPT = 5 Punch restart data
				= n Save restart data on unit n
	DEVICE	13-18	I6	DEVICE = 0 Solve diffusion problem
				= 5 Read punched cards and plot
				= m Read data from tape unit m and plot data
	NTAPE	19-24	I6	NTAPE = ℓ Store output of plottings on unit ℓ
	DETAIL	25-30	I6	DETAIL = 0 Do not use the method of computing $\Delta t_{\min}$ described in Section 4.3.1

<u>Card</u>	<u>Symbol</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>	
0				= 1	Compute $\Delta t_{\min}$ but do not print Δt_i and A_{ij}
				= 2	Compute $\Delta t_{\min}$ and print Δt_i and A_{ij}
				= 3	Print all geometrical arrays for information purpose
				= 4	Compute Δt_i and print all geometrical arrays
				= 5	Compute Δt_i and print Δt_i , A_{ij} , and all geometrical arrays
	NWARN	26-36	I6	NWARN = n	Proceed saving restarting data at n seconds prior to problem termination time
1	LABEL	01-72	12A6	Title Cards -	maximum of five cards followed by a blank card is allowed
2	FEOP	01-06	I6	FEOP = 1	No interface exists
				> 1	Open
	TOP	07-12	I6	TOP = 1	Problem is independent of temperature parameter
				= n	Compute temperature T using the n^{th} option provided in TEMPER subroutine
	SOP	13-18	I6	SOP = 1	Use zero source
				= n	Compute source S using the n^{th} option provided in SOURCE subroutine
	DOP	19-24	I6	DOP = 1	Use $D(m,n) = 1$ for diffusion coefficients
				= n	Compute diffusion coefficients $D(m,n)$ using the n^{th} option provided in DIFUSN subroutine
	LEFTOP	25-30	I6	LEFTOP = 1	Left boundary condition is
					$\text{ALPHAO}(1) \frac{\partial U_b}{\partial r} + \text{BETAO}(1) U_b = \text{GAMMAO}(1)$

<u>Card</u>	<u>Symbol</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>	
				= n	Compute left boundary condition using n th option provided in LEFTBC subroutine
2	RIGTOP	31-36	I6	RIGTOP = 1	Use right boundary condition $\text{ALPHO}(2) \frac{\partial U_b}{\partial r} + \text{BETAO}(2) U_b = \text{GAMMAO}(2)$
				= n	Compute right boundary condition using the n th option in RIGTBC subroutine
	LOWROP	37-42	I6	LOWROP = 1	Use lower boundary condition $\text{ALPHAO}(3) \frac{\partial U_b}{\partial z} + \text{BETAO}(3) U_b = \text{GAMMAO}(3)$
				= n	Compute lower boundary condition using the n th option in LOWERB subroutine
	UPEROP	43-48	I6	UPEROP = 1	Use lower boundary condition $\text{ALPHAO}(4) \frac{\partial U_b}{\partial z} + \text{BETAO}(4) U_b = \text{GAMMAO}(4)$
				= n	Compute upper boundary condition using the n th option provided in UPERBC subroutine
	VOP	49-54	I6	VOP = 0	Use zero mass transport coefficient
				= n	Compute mass transport coefficients V_r and V_z using the n th option provided in VLOCTY subroutine
	UOP	55-60	I6	UOP = 0	Zero initial concentration
				= 1	Read initial concentration, column by column
	IPRINT	61-66	I6	IPRINT = 1	Print concentration matrix U versus r-z plane
				= 2	Print boundary concentration and matrix U versus r-z plane
				= 3	Print short form of output

<u>Card</u>	<u>Symbol</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>	
2	IPLOT	67-72	I6	IPLOT = 0	Do not do plotting
				= 1	Plot the results in this run
				= 5	Punch and plot the output
				= -5	Punch the output but do not plot
				= n	Save the output on unit n and plot
				= -n	Save the output on unit n but do not plot
3	TMAX	01-12	E12.6		Maximum time $t_{\max}$ for which the problem is to be solved
	LAMDA	13-24	E12.6		Decay constant
	EPSHAF	25-36	E12.6		Maximum value of permitted error. If the percentage difference between the right- and left-hand sides of diffusion equation exceeds EPSHAF, reduce the time step to half of its present value
	EPSDBL	37-48	E12.6		Minimum value of permitted error. If the percentage difference between the right- and left-hand sides of diffusion equation becomes less than EPSDBL, double the time step
	DTMINO	49-60	E12.6	DTMINO < 0	Do not use time step less than $ DTMINO * \Delta t^*$
				DTMINO > 0	Do not use time step less than DTMINO
	DTMAXO	61-72	E12.6	DTMAXO < 0	Do not use time step greater than $ DTMAXO * \Delta t^*$
				DTMAXO > 0	Do not use time step greater than DTMAXO

Note: If the last four fields above were left blank, no computation of time step will be done and the time table described below will be used. Otherwise, the following actions will be taken:

1. If the time table below is null, initial time step of 10^{-6} will be used to compute the boundary conditions Δt^* and the functions used in the diffusion equation.

<u>Card</u>	<u>Symbol</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
				2. If the time table is not null, the table will be used as long as no computation of the time step becomes necessary. The computed time step will be used afterward.
4	DELTAT(I) I=1,2,...,NUMDT	01-12	E12.6	I th time step to be used for the next XITER iterations or in the interval 1 $TMAX(I-1) < t \leq TMAX(I)$
	TMAX(I)	13-24	E12.6	DELTA(I) is to be used in interval $TMAX(I-1) < t < TMAX(I)$
	XITERT(I)	25-36	E12.6	Number of iterations that DELTAT(I) is to be used
Note: Provide as many of the above and the following cards as are necessary, and follow them by a blank card. Also, provide data for either the second or third field.				
5	DELTAP(I) I=1,2,...,NUMDT	01-12	E12.6	I th print interval to be used for the next XITERP iterations or in interval $PTMAX(I-1) < t \leq PTMAX(I)$
	PTMAX(I)	13-24	E12.6	DELTAP(I) is to be used for the next XITERP iterations or in the interval $PTMAX(I-1) < t \leq PTMAX(I)$
	XITERP(I)	25-36	E12.6	Number of times that DELTAP(I) is to be used
Note: Provide as many of the following cards as are necessary, and follow them by a blank card. Also, provide data for either the second or the third field.				
6	DTPLOT(I) I=1,2,...,NUMPLT	01-12	E12.6	I th plot interval to be used for the next PITER iterations or in the interval $PTABLE(I-1) < t < PTABLE(I)$
	PTABLE(I)	13-24	E12.6	DPLOT(I) is to be used for the next PITER iterations or in the interval mentioned above
	PITER(I)	25-36	E12.6	Number of times that DTPLOT(I) is to be used
Note: Use as many of the following cards as are necessary to define the geometry of the system in the r-direction, and follow them by a blank card.				

<u>Card</u>	<u>Symbol</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
7	RNUM(I)	01-12	E12.6	Number of times the I^{th} mesh size is to be used in the r-direction
	DELTAR(I)	13-24	E12.6	I^{th} mesh size in the r-direction
Note: Use as many of the following cards as are necessary to define the geometry of the system in the z-direction, and follow them by a blank card.				
8	ZNUM(I)	01-12	E12.6	Number of times the J^{th} mesh size is to be used in the z-direction
	DELTAZ(I)	13-24	E12.6	J^{th} mesh size in the z-direction
9	ALPHA0(1)	01-12	E12.6	Coefficient α in Eq. 2.13 for defining the left boundary condition
	BETA0(1)	13-24	E12.6	Coefficient β in Eq. 2.13 for the left boundary condition
	GAMMA0(1)	25-36	E12.6	Right-hand term in Eq. 2.13 for defining the left boundary condition
	ALPHA0(2)	37-48	E12.6	Coefficient α in Eq. 2.13 for defining the right boundary condition
	BETA0(2)	49-60	E12.6	Coefficient β in Eq. 2.13 for the right boundary condition
	GAMMA0(2)	61-72	E12.6	Right-hand term in Eq. 2.13 for defining the right boundary condition
10	ALPHA0(3)	01-12	E12.6	Coefficient α in Eq. 2.13 for defining the lower boundary condition
	BETA0(3)	13-24	E12.6	Coefficient β in Eq. 2.13 for defining the lower boundary condition
	GAMMA0(3)	25-36	E12.6	Right-hand term in Eq. 2.13 for defining the lower boundary condition
	ALPHA0(4)	37-48	E12.6	Coefficient α in Eq. 2.13 for defining the upper boundary condition
	BETA0(4)	49-60	E12.6	Coefficient β in Eq. 2.13 for defining the upper boundary condition
	GAMMA0(4)	61-72	E12.6	Right-hand term in Eq. 2.13 for defining the upper boundary condition
11	IVFACE(M) M=1,2,...,MFACE	01-12	E12.6	m = IVFACE(M) indicates that the M^{th} vertical interface is at the radial mesh point m. Provide as many of these cards as are necessary, and follow them by a blank card

<u>Card</u>	<u>Symbol</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
12	IHFACE(N) N=1,2,...,NFACE	01-12	E12.6	n = IHFACE(N) indicates that the N th horizontal interface is at the axial mesh point number n. Provide as many of these cards as are necessary, and follow them by a blank card
Note: Use as many of the following cards as are necessary to define axial mass transport coefficient V_z , and follow them by a blank card.				
13	VZM(I)	01-12	E12.6	Radial mesh number at which axial transport coefficient V_z is nonzero
	VZV(I)	13-24	E12.6	mo = VZV(I) indicates that the value of axial transport function at (m,n) is $V_z(mo,n)$
Note: Use as many of the following cards as are necessary to define radial mass transport coefficient V_r , and follow them by a blank card.				
14	VRN(J)	01-12	E12.6	Axial mesh number at which radial transport coefficient V_r is nonzero
	VRV(J)	13-24	E12.6	no = VRV(J) indicates that the value of radial transport function at (m,n) is $V_r(m,no)$
15	U(m,n)	01-72	6E12.6	For UOP = 1 format (6E12.6) is used to read the initial concentration matrix U, column by column

6.2. INPUT PREPARATION FOR RESTARTING A PROBLEM

Only one new card is to be prepared for this case; this is the first card described in the previous section. The input data supplied by variables RESTART and NDUMPT via this card should be updated. The remaining data for restarting a problem are the input cards prepared for the original problem and the cards that were punched or saved on a tape in that run. These inputs are read in the following order:

1. Input cards related to master routine
2. Input cards, if any, prepared for auxiliary routines
3. Punched cards, if any, made by its previous run

6.3. INPUT PREPARATION FOR PLOTTING

Using FIPER-II, one may solve a problem in one run and plot the results in another run. This procedure is strongly recommended for both economy and flexibility. Experience has shown that the method of scaling concentration profiles versus r-z geometry to obtain descriptive three-dimensional plots is not fully known. Also, the position from which the concentration surface should be viewed and plotted plays an important factor. It is left to the user to decide how the surface is to be viewed. Furthermore, one may desire to plot the data using different angles of orientation or different scaling factors. On some occasions, it is desirable to plot a selection of data and omit the rest. Using the recommended procedure, all the tasks mentioned above can be easily accomplished.

Only two input cards are to be prepared for this case. In the case where plotting data were punched by a previous run, the two data cards should precede the remaining inputs.

<u>Card</u>	<u>Symbol</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
1	DEVICE	13-18	I6	DEVICE = 5 Read punched cards and plot = n Read data from tape unit n and plot
2	MMAX	01-12	E12.6	Number of mesh points in r-direction
	NMAX	13-24	E12.6	Number of mesh points in z-direction
	VECTOR(1)	25-36	E12.6	r-coordinate of the point from which the concentration surface is to be viewed [suggest VECTOR(1) = 1]
	VECTOR(2)	37-48	E12.6	z-coordinate of the point from which the concentration surface is to be viewed [suggest VECTOR(2) = 1]
	VECTOR(3)	49-60	E12.6	U-coordinate of the point from which the concentration surface is to be viewed [suggest VECTOR(3) = 1]
	PRINTT	61-72	E12.6	PRINTT = 0 Do not print the plot data = 1 Print the plot data

Note: If DEVICE = 5, enter the punched plot data here.

6.4. INFORMATION INPUT CARDS

This set of input cards is optional. If the user sets variable MORE equal to 2, then subroutine INFORM, which is called by MAIN1, expects to read and print a set of input cards. This option is provided to enable the user to document a complete description of his problems. The cards are read using format (12A6), and they should be followed by a card whose first 6 columns are \$\$\$\$\$\$. Note that this option may be used when a problem is to be solved, that is, when MAIN1 is executed. Also, note that this is the last batch of input cards for a problem.

7. DESCRIPTION OF OUTPUT

7.1. PRELIMINARY OUTPUT

The initial pages of a problem output contain a description of the problem. The input cards are also printed immediately after they have been read. In some cases, the input data and other arrays or variables computed as a result of such inputs are printed side by side.

When the input processing is complete, the diffusion coefficients, source function, equilibrium conditions, and other physical, dependent functions are computed, and their values corresponding to each mesh point in the problem are stored and printed in matrix form. Depending on the value of input variable DETAIL, other arrays and matrices containing the geometry-dependent parameters may be printed.

7.2. BASIC OUTPUT

The basic output is matrices of the concentrations at all points, with each matrix corresponding to a given point in time. The coordinates of points are also printed along the edges of matrices. Printed with each matrix are the problem time t , number of iterations performed, and the last value of time step. If input variable IPRINT is set to 2, then $U(m,n)$ where $m=2,3,\dots,M_{MAXM1}$, and $n=2,3,\dots,N_{MAXM1}$ plus concentrations at the boundaries are printed. For the case where IPRINT is set to 1, actual concentrations at all mesh points are printed. In addition to concentration matrices, other important quantities such as net release of particles across each boundary line and the values of volume integrals are printed.

For the case where input variable IPRINT is set to 3, a short form of the above-mentioned items, which excludes concentration matrices and tabulates the remaining items, is printed.

7.3. OUTPUT FOR PLOTTING

The output data for plotting are punched or saved on a tape, depending on the value assigned to the input variable IPLOT. In either case, they are saved in the following order:

1. Radial coordinates $r(m)$ of mesh points
2. Axial coordinates $z(n)$ of mesh points
3. Iteration number NITER and exact date and time NDATE at which a concentration matrix is saved
4. Concentration matrix $U(m,n)$; the matrix is saved column by column.

The data in items 3 and 4 are saved for each plot interval. Then the following sets of data are saved:

5. A blank card, if data are to be punched
6. Iteration number NITER and exact time and date NDATE at which the arrays below are punched
7. Release rate RELZL(k) from top of inner cylinder at time TIMET(k)
8. Release rate RELZR(k) from top of outer cylinder at time TIMET(k)
9. Total release TOTALL(k) from top of inner cylinder at time TIMET(k)
10. Total release TOTALR(k) from top of outer cylinder at time TIMET(k)
11. Total release TOTALC(k) from top of cylinder at time TIMET(k)
12. Time TIMET(k) at which above releases have been computed, where $k=1,2,\dots,NCOUNT$.
13. A blank card to separate these cards from the restart.

7.4. OUTPUT FOR RESTARTING

The last batch of data saved by the program is the output used to restart the problem at a later time. In the case where such data are to be punched, format (6E12.6) is used. Otherwise, the data are stored on unit NDUMPT in free format. The data are saved in the following order:

1. TIME, NITER, NDATE(I) where I = 1,2,3,4
2. TIME, PTIME, TIMEX,PTIMEX,DT,DTP
3. DTPVAR,DTVAR,DNEXT,DTHALF,DTOV,NDT
4. NDTP,RELZ1,RELZ11,RELZ12,OLDZ1,OLDZ11
5. OLDZ12,TOTZ1,TOTZ11,TOTZ12,NITER,NCOUNT
6. TFINAL,NCHANG,NOCHNG,DTPLOTS,PLOTTM,PLOTMX
7. NPLOTS
8. U(m,n) (column by column)
9. RELZL(k), k=1,2,...,NCOUNT
10. RELZR(k), k=1,2,...,NCOUNT
11. TOTALL(k), k=1,2,...,NCOUNT
12. TOTALR(k), k=1,2,...,NCOUNT
13. TOTALC(k), k=1,2,...,NCOUNT
14. TIMET(k), k=1,2,...,NCOUNT
15. Three blank cards

8. COMPUTATION OF CONCENTRATIONS AND CURRENTS

We have applied the FIPER-II code to a cylindrical diffusion problem characterized by simple boundary conditions. The two-region geometry has the following form:

The length of the cylinder is 0.932 cm, and its outer radius is 0.466 cm. The interface is at $z^* = 0.166$ cm and at $r^* = 0.083$ cm. The boundary conditions are given by:

$$U(r) = 1, \quad z = 0$$

$$U(r) = 0, \quad z = z_1$$

$$\frac{\partial U(z)}{\partial r} = 0, \quad r = r_1 \quad .$$

These conditions are independent of time t . Regions 1 and 2 are characterized by different diffusion coefficients. We have

$$D = \begin{cases} D_1 = 10^{-5} \text{ cm}^2/\text{sec} & \text{in region 1} \\ D_2 = 2 \times 10^{-8} \text{ cm}^2/\text{sec} & \text{in region 2} \end{cases} .$$

The decay constant λ is that of Sr-90, $\lambda = 7.85 \times 10^{-10} \text{ sec}^{-1}$. As for the interface, it first of all separates regions with diffusion coefficients (for example, a pore with a large diffusion coefficient D) from a material such as graphite which has a low diffusion coefficient D_2 . Furthermore, the condition that at the interface the chemical potentials of regions 1 and 2 are equal can lead to a ratio ϕ between the concentrations in regions 1 and 2. We shall study three subcases corresponding to three different values of ϕ .

8.1. SUBCASE 1, $\phi = 1$

In this case the time development of the concentration profiles is entirely determined by the diffusion coefficients in regions 1 and 2. At three consecutive times the concentration profiles U versus r, z are shown in Figs. 3 through 5. As indicated in Fig. 3, the concentration is always plotted in the vertical direction. The maximum and minimum values of the logarithmic concentration plot are given on the bottom as 1 and 1.4435×10^{-19} , respectively. One clearly sees from Figs. 3 through 5 that the low diffusion constant in region 2 gives rise to large concentration gradients until we reach the steady state (Fig. 3) after a time $(r-r_*)^2/(4 D_2) \sim 10^6 \text{ sec}$. The average concentration

$$\bar{U}(z) = \frac{\int_0^{r_1} U(r, z) r \, dr}{r_1^2/2}$$

is plotted as a function of z in Figs. 6 through 8. In Figs. 9 and 10 the release rate and the release are plotted as a function of time t . The release rate initially increases slowly with t and finally saturates at a time $t \sim (z-z^*)^2/(4 D_2)$.

8.2. SUBCASE 2, $\phi = 10$

In this case we assume that region 2 has a small adsorptivity for the diffusing species. This manifests itself by a sharp drop of the concentration at the interface. Figures 11 through 18 show that the release from region 2 is much smaller than in subcase 1 (Fig. 10).

8.3. SUBCASE 3, $\phi = 0.1$

From the physical point of view, this is the most interesting subcase. Region 1 corresponds to a fast diffusion path. It is surrounded by region 2 where the diffusion is low but the adsorptivity is large, as in reactor graphite. Hence, region 2 initially provides a sink for the diffusing species (Figs. 19 through 24). Once region 2 is saturated, it releases more atoms than region 1 (Figs. 25 and 26).

CONCENTRATION VS. R-Z PLANE.

24 FEB 72 (14.40.57)

TIME= 2.0110+03 SS(1)= 1.0000+10 SS(2)= 0.0000 ZMIN= 1.4435-19 ZMAX= 1

Fig. 3. Concentration profile $\log U$ versus r, z for $\phi = 1$ at time $t = 2.011 \times 10^3$ sec

CONCENTRATION VS. R-Z PLANE.

24 FEB 72 (14.40.57)

TIME= 2.0086+04 SS(1)= 1.0000+13 SS(2)= 0.0000 ZMIN= 1.4435-19 ZMAX= 1

Fig. 4. Concentration profile $\log U$ versus r, z for $\phi = 1$ at time $t = 2.0086 \times 10^4$ sec

CONCENTRATION VS. R-Z PLANE.

24 FEB 72 (14.40.57)

U

TIME= 2.0027+05 SS(1)= 1.0000+04 SS(2)= 0.0000 ZMIN= 1.4435-19 ZMAX= 1

Fig. 5. Concentration profile $\log U$ versus r, z for $\phi = 1$ at time $t = 2.0027 \times 10^5$ sec

TIME = 2.0110+03

24 FEB 72 (14:40:57)

AXIAL AVERAGE CONCENTRATION (MICRO MOLES)

SCALE=(1.00+07, 0.00)

AXIAL DISTANCE (CM.)

SCALE=(1.00+01, 0.00)

Fig. 6. Average concentration versus z for $\phi = 1$ at time $t = 2.011 \times 10^3$ sec

Fig. 7. Average concentration versus z for $\phi = 1$ at time $t = 2.0086 \times 10^4$ sec

Fig. 8. Average concentration versus z for $\phi = 1$ at time $t = 2.0027 \times 10^5$ sec

Fig. 9. Release rate as a function of time for $\phi = 1$

SCALE=(1.00+11, 0.00)

TOTAL RELEASE PER REGION (MICROMOLES)

TIME = 4.6397+05

24 FEB 72 (14:40:57)

TIME (SEC.)

Fig. 10. Release as a function of time for $\phi = 1$

CONCENTRATION VS. R-Z PLANE.

25 FEB 72 (10.01.52)

TIME= 2.0110+03 SS(1)= 1.0000+10 SS(2)= 0.0000 ZMIN= 1.4435-19 ZMAX= 1

Fig. 11. Concentration profile $\log U$ versus r, z for $\phi = 10$ at time $t = 2.011 \times 10^3$ sec

CONCENTRATION VS. R-Z PLANE

25 FEB 72 (10.01.52)

TIME= 2.0086+04 SS(1)= 1.0000+13 SS(2)= 0.0000 ZMIN= 1.4433-19 ZMAX= 1

Fig. 12. Concentration profile $\log U$ versus r, z for $\phi = 10$ at time $t = 2.0086 \times 10^4$ sec

CONCENTRATION VS. R-Z PLANE.

25 FEB 72 (10.01.92)

TIME= 2.0027+05 SS(1)= 1.0000+05 SS(2)= 0.0000 ZMIN= 1.4433-19 ZMAX= 1

Fig. 13. Concentration profile $\log U$ versus r, z for $\phi = 10$ at time $t = 2.0027 \times 10^5$ sec

Fig. 14. Average concentration versus z for $\phi = 10$ at time $t = 2.011 \times 10^3$ sec

AXIAL AVERAGE CONCENTRATION (MICRO MOLES)

TIME = 2.0086+04

25 FEB 72 (10.01.52)

Fig. 15. Average concentration versus z for $\phi = 10$ at time $t = 2.0086 \times 10^4$ sec

Fig. 16. Average concentration versus z for $\phi = 10$ at time $t = 2.0027 \times 10^5$ sec

Fig. 17. Release rate as a function of time for $\phi = 10$

)

TIME = 2.4247+05

25 FEB 72 (10.01.52)

SCALE=(1.00+13, 0.00)

TOTAL RELEASE PER REGION (MICROMOLES)

Fig. 18. Release as a function of time for $\phi = 10$

Fig. 19. Concentration profile $\log U$ versus r, z for $\phi = 0.1$ at time $t = 2.011 \times 10^3$ sec

CONCENTRATION VS. R-Z PLANE.

25 FEB 72 (12.29.55)

Fig. 20. Concentration profile $\log U$ versus r, z for $\phi = 0.1$ at time $t = 2.0086 \times 10^4$ sec

CONCENTRATION VS. R-Z PLANE.

25 FEB 72 (12.29.55)

TIME= 2.0027+05 SS(1)= 1.0000+04 SS(2)= 0.0000 ZMIN= 1.4435-19 ZMAX= 1

Fig. 21. Concentration profile $\log U$ versus r, z for $\phi = 0.1$ at time $t = 2.0027 \times 10^5$ sec

Fig. 22. Average concentration versus z for $\phi = 0.1$ at time $t = 2.011 \times 10^3$ sec

TIME = 2.0086+04

25 FEB 72 (12.29.55)

AXIAL AVERAGE CONCENTRATION (MICRO MOLES)

SCALE=(1.00+03, 0.00)

AXIAL DISTANCE (CM.)

SCALE=(1.00+01, 0.00)

Fig. 23. Average concentration versus z for $\phi = 0.1$ at time $t = 2.0086 \times 10^4$ sec

TIME = 2.0027+05

25 FEB 72 (12.29.55)

AXIAL AVERAGE CONCENTRATION (MICRO MOLES)

SCALE=(1.00+01, 0.00)

AXIAL DISTANCE (CM.)

SCALE=(1.00+01, 0.00)

Fig. 24. Average concentration versus z for $\phi = 0.1$ at time $t = 2.0027 \times 10^5$ sec

RELEASE RATE PER REGION (MICRO-MOLES/SEC.)

SCALE=(1.00+16, 0.00)

TIME = 2.4247+05

25 FEB 72 (12.29.55)

TIME (SEC.)

Fig. 25. Release rate as a function of time for $\phi = 0.1$

TIME = 2.4247+05

25 FEB 72 (12:29:55)

SCALE=(1.00+14, 0.00)

TOTAL RELEASE PER REGION (MICROMOLES)

TIME (SEC.)

Fig. 26. Release as a function of time for $\phi = 0.1$

REFERENCES

1. Forutanpour, B., and B. W. Roos, "FIPER X, A FORTRAN V Program for the Solution of One-Dimensional Linear and Non-Linear Diffusion Problems," USAEC Report GA-9904, Gulf General Atomic, September 1969.
2. Peaceman, D. W., and H. H. Rachford, "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. Soc. Ind. Appl. Math. 3, No. 1, 28-41 (1955).