WASTE ISOLATION PILOT PLANT
Sandia National Laboratories

Software Cover Sheet

Form Number: 461-B **Effective: 9/12/96**

Procedure: QAP 19-1 **Page** 1 **of** 1

1. Document Name: User's Manual
2. Document Version Number: 1.02
3. WPO Number: 45734
4. Software Name: SECOTP2D
5. Version ID: 1.41
6. Software Classification[1]: SNL-SW
7. Summary of Contents COVER SHEET, DOCUMENT, REVIEW FORM, COMMENTS

   Total number of pages in body of document: 63

8. **Code Sponsor/Preparer(s) Signature(s)[2]:**

James L. Ramsey (Code Sponsor's Name *(print)*)	James L Ramsey (Signature)	8/15/97 (Date)
John Stikar (Preparer's Name *(print)*)	John A. Stikar (Signature)	8/15/97 (Date)
(Preparer's Name *(print)*)	(Signature)	(Date)

9. **Reviewer(s) Signature(s)[2]:**

Alex H. Treadway (Reviewer's Name *(print)*)	Alex H. Treadway (Signature)	8/15/97 (Date)
(Reviewer's Name *(print)*)	(Signature)	(Date)
(Reviewer's Name *(print)*)	(Signature)	(Date)

10. **Department Manager Approval:**

Hong-Nian Jow (Department Manager's Name *(print)*)	Hong N. Jow (Signature)	8/15/97 (Date)

11. **SCM Coordinator's Signature:**

John J. Loukota (SCM Coordinator's Name *(print)*)	John J. Loukota (Signature)	8/15/97 (Date)

(1) *Choose from classifications in Appendix A.*

(2) *If there are more than one preparer or reviewer, each person's responsibilities shall be delineated in the document. Each reviewer shall complete a reviewer's form which applies to the document reviewed, or document their review in a memorandum. The review documentation shall be included in the document. Attach additional Software Cover Sheets if there are more than three Code Sponsor/preparers or reviewers.*

WPO# 45734 Keywords: SECOTP2D : ST2D2 : UM :

SWCF File Code: SWCF- A (A or C WBS#) : 1.4.01.6.1 : QA : PA (Alpha Code (s)) :SFT

WIPP PA

User's Manual

for

SECOTP2D, Version 1.41

Document Version 1.02

WPO # 45734

June 9, 1997

Table of Contents

Figures

Tables

1.0 INTRODUCTION

This document serves as a User's Manual for SECOTP2D. As such, it describes the code's purpose and function, the user's interaction with the code, and the models and methods employed by the code. An example output file is included for the user's convenience.

SECOTP2D has been qualified for use under the quality assurance procedure 19-1 (QAP 19-1) of the Nuclear Waste Management Division of Sandia National Laboratories. Documentation of the testing process can be obtained from the Sandia WIPP Central Files (SWCF). The most pertinent documents are the Requirements Document & Verification and Validation Plan (RD/VVP, Version 1.02, WPO# 45732), and the Validation Document (VD, Version 1.02, WPO# 45735).

The SECOTP2D code, Version 1.41, performs multiple component transport in fractured porous media. The media are represented using discrete-fracture and dual-porosity models. In the discrete-fracture model, the system is assumed to have multiple planar fractures (slab geometry). The governing equations include the effect of advection, hydrodynamic dispersion, linear radioactive decay and generation, and an assumption of linear equilibrium isotherms. In the dual-porosity model, the system is comprised of numerous parallel fractures and a porous rock matrix. The matrix governing equations represent the effect of diffusion, decay, and generation. The flow field, which can be either steady or unsteady, can be obtained from SECOFL2D [1] or an equivalent single-phase flow code.

SECOTP2D is an implicit finite volume code that is second-order accurate in space and time. It uses operator splitting, an Approximate Factorization procedure, the delta formulation, and a finite volume staggered mesh. The advective terms are modeled using either Total Variation Diminishing (TVD), central differencing, or an upwind scheme. Time discretization is a generalized three-level scheme. SECOTP2D features manual and automatic boundary definition that can vary from cell to cell. The code has modern FORTRAN structure and its operator splitting scheme is efficient in execution time and memory usage. The code has an option of computing the history of integrated discharge around any number of rectilinear closed boundaries within the computational mesh.

This manual describes the governing equations, the development of numerical schemes, and detailed explanations of how to use Version 1.41 in the environment of the Compliance Assessment Methodology Controller (CAMCON).

The SECOTP2D code is a part of SECO suite of codes that includes flow, transport, and particle tracking in 2- and 3-dimensions. SECOTP2D has been referred to as SECO-TRANSPORT in the open literature [1, 2, 3, 4].

1.1 Software Identifier

Code Name: SECOTP2D (Sandia-ECOdynamics/TransPort in 2 Dimensions)
WIPP Prefix: ST2D2
Version Number: 1.41, June 9, 1997
Platform: FORTRAN 77 for OpenVMS AXP, ver 6.1, on DEC Alpha

1.2 Points of Contact

1.2.1 Code Sponsor

James Ramsey
Department 6849
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1328
Voice: (505) 848-0892
Fax: (505) 848-0705

1.2.2 Code Consultant

James Ramsey
Department 6849
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1328
Voice: (505) 848-0892
Fax: (505) 848-0705

2.0 FUNCTIONAL REQUIREMENTS

SECOTP2D Version 1.41 is required to model/employ the following functionality as part of the 1997 WIPP Performance Assessment Verification Test (PAVT).

R.1 - Two-dimensional advective/dispersive transport in a heterogeneous flow field
R.2 - Van Leer TVD limiter with Courant number restraints
R.3 - Matrix/fracture coupling and mass transfer
R.4 - Matrix Diffusion
R.5 - Matrix Retardation
R.6 - Radioisotope chain decay
R.7 - Point source term
R.8 - Dirichlet Boundary conditions
R.9 - Discharge reporting
R.10 - Mass balance reporting
R.11 - Variable time stepping
R.12 - Backward differencing in time
R.13 - Variable Grid Spacing

External Interface Requirements

R.14 - All of the input files required by this code are generated by PRESECOTP2D.
R.15 - The results from SECOTP2D are added to a CAMDAT database file by POSTSECOTP2D.
R.16 - This code must be linked to the libraries CAMCOM_LIB and CAMSUPES_LIB.

2.1 Additional Functionality

SECOTP2D Version 1.41 is also capable of modeling/employing the following functionality not required in the PAVT. Functionality not formally tested under QAP 19-1 is identified.

A.1 - One-dimensional centered differencing
A.2 - One-dimensional transport
A.3 - Single species decay
A.4 - Constant time step
A.5 - Multiple Point Source Terms
A.6 - Neuman boundary condition
A.7 - Setting of Initial Condition
A.8 - Upwind differencing of the advective term, limiter function 0
A.9 - Center differencing of the advective term, limiter function 1
A.10 - Weighted upstream differencing, limiter function 2 (**untested**)
A.11 - Minmod TVD limiters, limiter functions 3-6 (**untested**)

A.12 - Roe Superbee TVD limiter, limiter function 8 (**untested**)
A.13 - Automatic setting of boundary conditions (**untested**)
A.14 - Single porosity, or discrete fracture transport (**untested**)
A.15 - Matrix concentration reporting at multiple locations (**untested**)
A.16 - Second order three point backwards time differencing (**untested**)
A.17 - Second order central time differencing (**untested**)
A.19 - Reporting of maximum concentration (**untested**)
A.20 - Activity reporting (**untested**)

3.0 REQUIRED USER TRAINING AND/OR BACKGROUND

For the theoretical section of this manual to be useful, the user will need the following:

- An understanding of partial differential equations.
- Senior level undergraduate course in linear algebra.
- Graduate or senior level undergraduate course in numerical methods.
- First level undergraduate course in groundwater flow and transport.

To run the code and the associate pre- and postprocessors, an understanding of the CAMCOM environment is required.

To apply SECOTP2D effectively, the user should be aware of the code's capabilities and limitations. It is recommended that the user run one or more of the problems used to qualify the code. A description of the test problems can be found in the requirements document and verification and validation plan (RD/VVP) [5]. Test problem results are located in the validation document (VD) [6].

4.0 DESCRIPTION OF THE MODELS AND METHODS

The equation for the transport of the *k*th radionuclide component in a porous media (*N* species) can be written as (for more detailed derivation of this equation see references [7, 8, 9, 10])

$$\phi R_k\left(\frac{\partial C_k}{\partial t}\right)-\nabla\cdot\left[\phi\mathbf{D}\nabla C_k-\mathbf{V}C_k\right]=-\phi R_k\lambda_k C_k+\phi R_{k-1}\lambda_{k-1}C_{k-1}+Q\tilde{C}_k+\Gamma_k \tag{1}$$

where $k = 1,...,N$ and each dependent variable C_k (M/L^3) is the concentration of the *k*th radionuclide. For k=1, the term involving C_{k-1} is omitted. Physical parameters include $\mathbf{D}$ (L^2/T), a 2 × 2 hydrodynamic dispersion tensor; $\mathbf{V}$ (L/T), the specific discharge; ϕ (dimensionless), the effective porosity defined as the ratio of fracture pore volume per unit volume of porous medium; R_k (dimensionless), the retardation coefficient; λ_k ($1/T$), the species decay constant; $\tilde{C}_k$ (M/L^3), the concentration of the *k*th injected radionuclide; Q ($1/T$), the well-specific injection rate defined as the volume of liquid injected per unit volume of porous medium; and Γ_k (M/TL^3), the mass transfer term between the fracture and the matrix. The hydrodynamic dispersion tensor [6] is defined as

$$\phi\mathbf{D}=\frac{1}{|\mathbf{V}|}\begin{pmatrix}u & -v\\ v & u\end{pmatrix}\begin{pmatrix}\alpha_L & 0\\ 0 & \alpha_T\end{pmatrix}\begin{pmatrix}u & v\\ -v & u\end{pmatrix}+\phi D_k^*\tau\,\mathbf{I} \tag{2}$$

where α_L and α_T are the longitudinal and transverse dispersivities (L), D_k^* *is* the free water molecular diffusion coefficient (L^2/T), τ is the tortuosity (dimensionless), u and v are the components of the Darcy velocity or specific discharge (L/T), and I is an identity matrix (dimensionless).

The N fracture equations are linear and sequentially-coupled. Three different boundary conditions, Dirichlet, Neumann, and flux, are available which can be used on different parts of the rectangular domain.

The flow field $\mathbf{V}$ is assumed to be independent of the solute concentration. In practice, the flow field is computed prior to running SECOTP2D and is obtained from the SECOFL2D code [1].

Because the dual-continuum model [8, 11] includes the exchange of mass between the matrix block and the fracture, it is necessary to solve a transport equation in the matrix block. Assuming there is no fluid flow (advection) in the matrix, the equation for the concentration of the *k*th species is given by [7].

$$\phi' R_k'\left(\frac{\partial C_k'}{\partial t}\right)-\frac{\partial}{\partial\chi}\left(\phi' D'\frac{\partial C_k'}{\partial\chi}\right)=-\phi' R_k'\lambda_k C_k'+\phi' R_{k-1}'\lambda_{k-1}C_{k-1}' \tag{3}$$

where χ is the coordinate originating from the symmetry line of the matrix block, and the prime denotes matrix properties. $D' = D_k^* \tau'$ is the coefficient of the molecular diffusion, where τ' is the matrix tortuosity. The remaining symbols have the same meaning as those in Eq. 1.

The equations for the fracture and the matrix block are coupled through a mass transfer term, Γ_k, which, for a slab block model (Figure 1), is given by

$$\Gamma_k = -\frac{2\phi}{b}\left(\phi' D' \frac{\partial C_k'}{\partial \chi}\bigg|_{\chi=\frac{b'}{2}}\right) \tag{4}$$

where b is the fracture aperture, b' is the matrix block length, and ϕ is the fracture porosity defined above. The term $2\phi/b$ in Eq. 4 represents the specific surface area of the coupled system (ratio of matrix surface area in contact with the fracture to the bulk volume).

The initial and boundary conditions of the matrix block are given by

$$C_k'(\chi,0) = 0 \tag{5}$$

$$\frac{\partial C_k'}{\partial \chi}(0,t) = 0 \tag{6}$$

$$C_k'\left(\frac{b'}{2},t\right) = C_k \; . \tag{7}$$

4.1 Numerical Discretization

4.1.1 Finite Volume Grid

SECOTP2D uses a finite volume staggered mesh as shown in Figure 2. The velocity components (u,v) are defined at cell faces and concentrations are evaluated at cell centers. The boundary conditions are applied using ghost cells. These additional cells are used to construct boundary conditions.

4.1.2 Coordinate Transformation

The governing equations are transformed from Cartesian coordinates (physical space) to stretched Cartesian coordinates (computational space). The transformations are chosen so that the grid spacing in the computational space is uniform and of unit length. This produces a computational space that is a rectangular domain with a uniform mesh. The standard unweighted differencing scheme can now be applied in the numerical formulation. The metrics of the transformation are space dependent. To transform the governing equation from physical to computational space, chain rule expansions are used to represent the Cartesian derivatives in the computational space. The transformed equations, through some algebraic manipulation, are

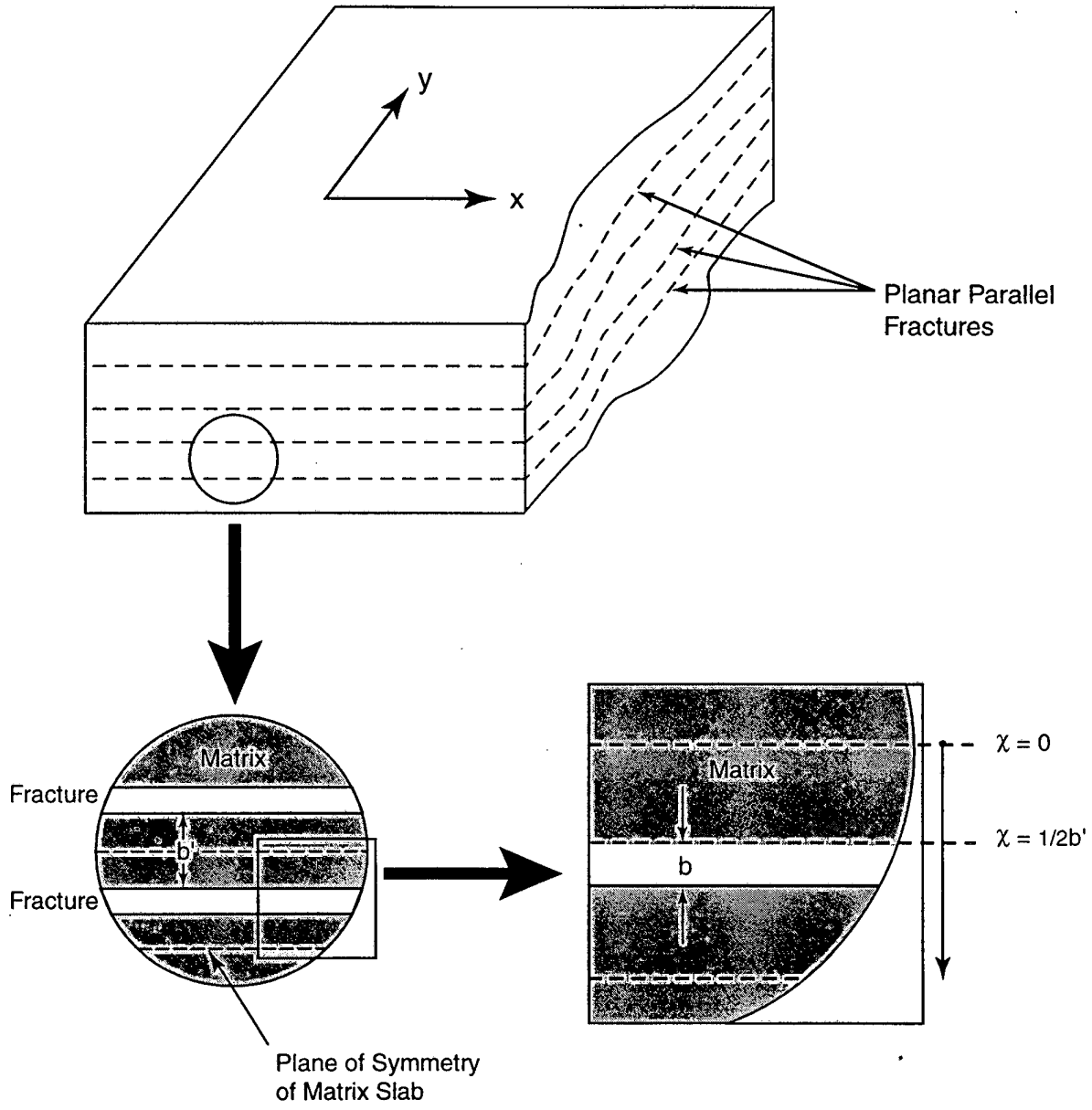


Figure 1. Schematic of dual-porosity model.

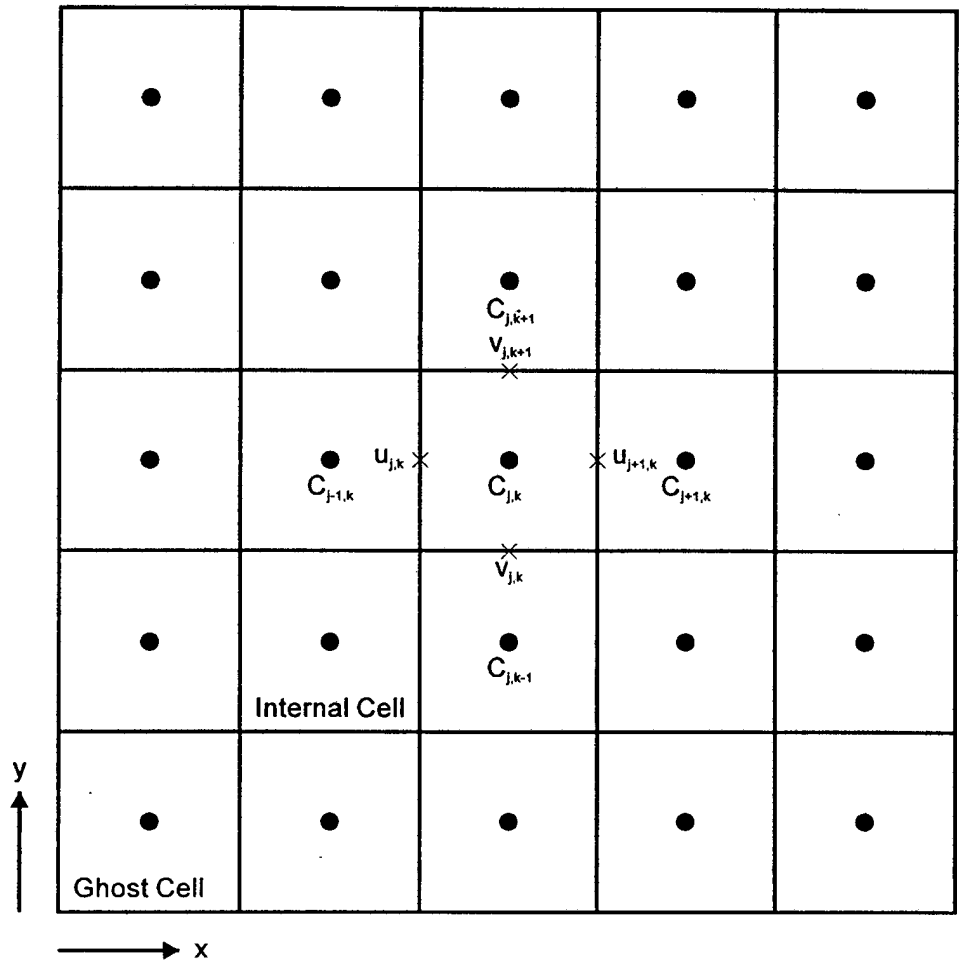


Figure 2. Schematic of finite volume staggered mesh showing internal and ghost cells. The concentrations are defined at cell centers and velocities at cell faces.

written in strong conservation form. For additional information on the coordinate transformation used here, see Ref. [12].

4.1.3 Invariants of the Transformation

In the process of transforming the governing equations, additional terms containing the derivatives of the transformation are generated. These terms are collected at the end of the transformation and are known as the invariant of transformation. With the use of metric definitions, it can be shown that the invariants are analytically zero. There is an important problem associated with these invariants when equations are evaluated for uniform properties. If we require that the transformed governing equations satisfy the uniform conditions, the discretized form of the invariants must sum to zero. In two-dimensional, standard central differencing does satisfy this requirement.

4.1.4 Fracture Equation

Eq. 1 has been transformed from physical into computational space using the transformation

$$x = x(\xi) \tag{8}$$

$$y = y(\eta) \tag{9}$$

where the transformation metrics are $\xi_x = Jy_\eta$, $\eta_y = Jx_\xi$, and $J = \xi_x\eta_y$, where the subscripts denote partial derivatives. The transformed equation, with further algebraic manipulations, is put into a strong conservation form to ensure mass conservation. The resultant transport equation in computational space equation is given by

$$\begin{aligned}\phi R_k \frac{\partial}{\partial t}\left(\hat{C}_k\right)+\frac{\partial}{\partial \xi}\left(\hat{E}\right)+\frac{\partial}{\partial \eta}\left(\hat{F}\right)=&\frac{\partial}{\partial \xi}\left(\hat{E}_{v1}\right)+\frac{\partial}{\partial \xi}\left(\hat{E}_{v2}\right)\\ &+\frac{\partial}{\partial \eta}\left(\hat{F}_{v1}\right)+\frac{\partial}{\partial \eta}\left(\hat{F}_{v2}\right)\\ &-\phi R_k\lambda_k\hat{C}_k+\phi R_{k-1}\lambda_{k-1}\hat{C}_{k-1}\\ &+\hat{Q}+\hat{\Gamma}\end{aligned} \tag{10}$$

where

$$\hat{C}_k = \frac{C_k}{J} \tag{11}$$

$$\hat{E} = \xi_x u\hat{C}_k \tag{12}$$

$$\hat{F} = \eta_y v\hat{C}_k \tag{13}$$

$$\hat{E}_{v1} = \frac{\xi_x^2 \phi D_{11}}{J} \frac{\partial \hat{D}_k}{\partial \xi} \tag{14}$$

$$\hat{E}_{v2} = \frac{\xi_x \eta_y \phi D_{12}}{J} \frac{\partial \hat{C}_k}{\partial \eta} \tag{15}$$

$$\hat{F}_{v1} = \frac{\xi_x \eta_y \phi D_{21}}{J} \frac{\partial \hat{C}_k}{\partial \xi} \tag{16}$$

$$\hat{F}_{v2} = \frac{\eta_y^2 \phi D_{22}}{J} \frac{\partial \hat{C}_k}{\partial \eta} \tag{17}$$

$$\hat{Q} = \frac{Q \tilde{C}_k}{J} \tag{18}$$

$$\hat{\Gamma} = \frac{\Gamma}{J} \quad . \tag{19}$$

Eq. 10 is solved using an implicit Approximate Factorization procedure [12]. The advective terms are modeled by TVD [13] and the remaining terms by central differencing. A generalized three-level implicit finite volume scheme, in delta form [12], can be written as

$$\phi R_k \Delta \hat{C}_k^n = \frac{\theta \Delta t}{1+\varphi} \left(\phi R_k \Delta \hat{C}_k^n \right)_t + \frac{\Delta t}{1+\varphi} \left(\phi R_k \Delta \hat{C}_k^n \right)_t + \frac{\varphi}{1+\varphi} \left(\phi R_k \Delta \hat{C}_k^{n-1} \right) \tag{20}$$

where

$$\Delta \hat{C}_k^n = \Delta \hat{C}_k^{n+1} - \hat{C}_k^n \quad . \tag{21}$$

The $\Delta \hat{C}_k^n$ can be thought of as a correction to advance the solution to a new time-level (n+l). Eq. 20, with appropriate choice of the parameters θ and φ, produces two and three-level implicit schemes as shown in Table 1.

Table 1. List of Temporal Discretization Schemes

θ	φ	Schemes	Truncation error
1	0	Euler, implicit	O (Δt)
½	0	Trapezoidal, implicit	O (Δt^2)
1	½	3-point-backward, implicit	O (Δt^2)

Applying Eq. 20 to Eq. 10, we have

$$\begin{aligned}\phi R_k \Delta\hat{C}_k^n = \frac{\theta\Delta t}{1+\varphi}\Big[&-\left(\Delta\hat{E}^n\right)_\xi - \left(\Delta\hat{F}^n\right)_\eta + \left(\Delta\hat{E}_{v1}^n\right)_\xi + \left(\Delta\hat{F}_{v2}^n\right)_\eta - \phi R_k\lambda_k\Delta\hat{C}_k^n \\ &+\phi R_{k-1}\lambda_{k-1}\Delta\hat{C}_{k-1}^n + \Delta\hat{\Gamma}_k^n + \Delta\hat{Q}_k^n\Big] \\ &+\frac{\theta\Delta t}{1+\varphi}\left[\left(\Delta\hat{E}_{v2}^{n-1}\right)_\xi + \left(\Delta\hat{F}_{v1}^{n-1}\right)_\eta\right] \\ &+\frac{\Delta t}{1+\varphi}\Big[-\hat{E}_\xi^n - \hat{F}_\eta^n + \left(\hat{E}_{v1}^n\right)_\xi + \left(\hat{E}_{v2}^n\right)_\xi + \left(\hat{F}_{v1}^n\right)_\eta + \left(\hat{F}_{v2}^n\right)_\eta \\ &\quad -\phi R_k\lambda_k\hat{C}_k^n + \phi R_{k-1}\lambda_{k-1}\hat{C}_{k-1}^n + \hat{Q}_k^n + \hat{\Gamma}_k^n\Big] \\ &+\frac{\varphi}{1+\varphi}\left[\phi R_k\Delta\hat{C}_k^{n-1}\right]\end{aligned} \tag{22}$$

or

$$\begin{aligned}&\phi R_k\Delta\hat{C}_k^n + \frac{\theta\Delta t}{1+\varphi}\left[\left(\Delta\hat{E}^n\right)_\xi + \left(\Delta\hat{F}^n\right)_\eta - \left(\Delta\hat{E}_{v1}^n\right)_\xi - \left(\Delta\hat{F}_{v2}^n\right)_\eta + \phi R_k\lambda_k\Delta\hat{C}_k^n - \Delta\hat{\Gamma}_k^n - \Delta\hat{Q}_k^n\right] \\ &\quad = \frac{\theta\Delta t}{1+\varphi}\left[\left(\Delta\hat{E}_{v2}^{n-1}\right)_\xi + \left(\Delta\hat{F}_{v1}^{n-1}\right)_\eta + \phi R_{k-1}\lambda_{k-1}\Delta\hat{C}_{k-1}^n\right] \\ &\quad +\frac{\Delta t}{1+\varphi}\Big[-\hat{E}_\xi^n - \hat{F}_\eta^n + \left(\hat{E}_{v1}^n\right)_\xi + \left(\hat{E}_{v2}^n\right)_\xi + \left(\hat{F}_{v1}^n\right)_\eta + \left(\hat{F}_{v2}^n\right)_\eta \\ &\quad -\phi R_k\lambda_k\hat{C}_k^n + \phi R_{k-1}\lambda_{k-1}\hat{C}_{k-1}^n + \hat{Q}^n + \hat{\Gamma}^n\Big] \\ &\quad +\frac{\varphi}{1+\varphi}\left[\phi R_k\Delta\hat{C}_k^{n-1}\right] \\ &\quad = \text{RHS}\end{aligned} \tag{23}$$

Note that the delta cross derivative terms, $(\Delta E_{v2})_\xi$ and $(\Delta F_{v1})_\eta$, are time-lagged from the n to n-1 time level in both Eqs. 22 and 23. The explicit treatment of the cross derivative terms is necessary to facilitate the operator slitting/factorization scheme used to solve Eq. 23. A discussion of the operator splitting solver is presented later in this section.

4.1.5 Flux Limiter Functions and TVD

The advective terms, $\hat{E}$ and $\hat{F}$ in Eq. 10, are approximated using a TVD flux scheme developed [3] for staggered meshes. TVD implements a combination of centered and locally upstream (or "upwind") weighted schemes. The basic concept of flux limiters (TVD and other

algorithms) is to apply upstream weighting locally and with only enough weighting to eliminate nonphysical oscillations near sharp concentration gradients. As the mesh is refined, the upstream weighting decreases, and the method approaches second-order accuracy.

In physical space, the flux limited form of uC at the $j-\frac{1}{2},k$ interface is written

$$uC\Big|_{j-1/2,k} = \frac{1}{2}\left(1-\Phi_{j-1/2,k}\right)\left[\left(C^n_{j,k}+C^n_{j-1,k}\right)u^n_{j-1/2,k}-\left(C^n_{j,k}-C^n_{j-1,k}\right)\left|u^n_{j-1/2,k}\right|\right] + \frac{1}{2}\Phi_{j-1/2,k}\left(C^n_{j,k}+C^n_{j-1,k}\right)u^n_{j-1/2,k} \tag{24}$$

In computational space, the equivalent expression is

$$\hat{E}^n_{j-1/2,k} = \frac{1}{2}a^n_{j-1/2,k}\left\{\left(1-\Phi_{j-1/2,k}\right)\left[\left(\hat{C}^n_{j,k}+\hat{C}^n_{j-1,k}\right)-\sigma\left(\hat{C}^n_{j,k}-\hat{C}^n_{j-1,k}\right)\right] + \Phi_{j-1/2,k}\left(\hat{C}^n_{j,k}+\hat{C}^n_{j-1,k}\right)\right\} \tag{25}$$

where $a = \xi_x u$ and $\sigma = \text{sign}\ (a)$.

The function Φ is called a limiter function [13]. SECOTP2D uses nine different limiter functions, six of which are TVD. The TVD limiters range from very compressive (Roe superbee) to very dissipative (minmod) [13]. The limiter functions are defined as

0. $\Phi(r) = 0$ upstream differencing

1. $\Phi(r) = 1$ central differencing

2. $\Phi(r) = w$ weighted upstream and central differencing

3. $\Phi(r) = minmod\ (1, r)$

4. $\Phi(r) = minmod\ (1, 2r)$

5. $\Phi(r) = minmod\ (2, r)$

6. $\Phi(r) = minmod\ (2, 2r)$

7. $\Phi(r) = max\ (0, min\ (2r, \frac{r+|r|}{1+|r|}))$ van Leer's MUSCL

8. $\Phi(r) = max\ (0, min(2r, 1), min\ (r, 2))$ Roe superbee

where r is a function of the concentration gradient and direction of flow. When u is positive in x (ξ) direction, r at the $j-\frac{1}{2}, k$ interface is defined as

$$r_x = \frac{C_x\big|_{j-3/2,\,k}}{C_x\big|_{j-1/2,k}} = \frac{C_\xi \xi_x\big|_{j-3/2,\,k}}{C_\xi \xi_x\big|_{j-1/2,\,k}} \quad ; u > 0$$

where u is negative, r is defined as

$$r_x = \frac{C_x\big|_{j+1/2,\,k}}{C_x\big|_{j-1/2,k}} = \frac{C_\xi \xi_x\big|_{j+1/2,\,k}}{C_\xi \xi_x\big|_{j-1/2,\,k}} \quad ; u < 0 \quad .$$

Similarly, in the y (η) direction, r at $j,\ k-\frac{1}{2}$ interface is defined as

$$r_y = \frac{C_y\big|_{j,k-3/2}}{C_y\big|_{j,k-1/2}} = \frac{C_\eta \eta_y\big|_{j,k-3/2}}{C_\eta \eta_y\big|_{j,k-1/2}} \quad ; v > 0$$

$$r_y = \frac{C_y\big|_{j,k+1/2}}{C_y\big|_{j,k-1/2}} = \frac{C_\eta \eta_y\big|_{j,k+1/2}}{C_\eta \eta_y\big|_{j,k-1/2}} \quad ; v < 0 \quad .$$

The minmod function is given by

$$minmod(a,\ b) = sign(a)\ max\ (0,\ min\ (|a|,\ sign(a) \cdot b)) \tag{26}$$

The current implementation of TVD is explicit, in that r is evaluated at the n rather than $n+1$ time level. This treatment of TVD has been found to be oscillatory when the cell Courant number, Cr, exceeds a value of one, where

$$Cr_x = \frac{u\Delta t}{\phi_f \Delta x}$$

$$Cr_y = \frac{v\Delta t}{\phi_f \Delta y} \quad .$$

To avoid oscillatory behavior, the application of a TVD limiter function is constrained to regions in which the Courant number is less than one. This constraint is invoked by setting the limiter functions to zero when the $Cr > 1$; in effect imposing a full upwinding scheme locally.

4.1.6 Approximate Factorization

The RHS of Eq. 23 involves terms that are at time level (n, n-1). After evaluation of these terms, the solution at the new time level (n+1) is obtained by

$$\left(I + \alpha_\xi L_{\xi\xi} + \alpha_\eta L_{\eta\eta} + S\right)\Delta\hat{C} = RHS \tag{27}$$

where I is an identity matrix, $L_{\xi\xi}$ and $L_{\eta\eta}$ are the x and y operators, and S is an implicit source term that accounts for decay and matrix/fracture mass transfer. To solve Eq. 27 for $\Delta\hat{C}$, a two-dimensional banded matrix inversion is required. The banded LU solvers are adequate for small size problems; however, for large problems they require large amounts of memory and CPU time. To alleviate these shortcomings, the two-dimensional operator can be Approximately Factored to represent two one-dimensional operators [12]. This decreases the operation count for the inversion of the coefficient matrix and reduces the memory usage. The implementation of implicit boundary conditions are quite different for the factored and unfactored operators. In the case of a full two-dimensional operator, the implicit boundary conditions are applied in a straight-forward manner with no complications; however, this is not the case for the factored operators in which intermediate boundary conditions are needed to construct the implicit boundary conditions.

With one-dimensional operators, the solution at the new time level (n+l) is obtained through the following sequence

$$\left(I + \alpha_\xi L_{\xi\xi} + a_\xi S\right)\Delta\overline{C} = RHS \tag{28}$$

$$\left(I + \alpha_\eta L_{\eta\eta} + a_\eta S\right)\Delta\hat{C} = \Delta\overline{C} \tag{29}$$

$$\hat{C}^{n+1}_{j,k} = \hat{C}^{n}_{j,k} + \Delta\hat{C}^{n}_{j,k} \tag{30}$$

where a_ξ and a_η are constants that establish the fraction of the implicit source allocated in each of the one-dimensional sweeps. Mathematically, the two constants are only constrained by the condition that they must sum to one, $a_\xi + a_\eta = 1$. Ramsey and Treadway [14], however, showed that **the choice of the source term splitting constants can dramatically influence the accuracy of the simulation and the mass conservation properties of the model**. Based on this work, it is recommended that a_ξ and a_η be set to 0 and 1, or 1 and 0, respectively. Furthermore, the global mass balance error should be checked following every use of the code.

Eq. 28 describes the initial sweep in the x or logical ξ direction. At the end of this sweep the intermediate solution, $\Delta\overline{C}$, becomes the RHS for a sweep in the y or logical η direction. The change in concentration, $\Delta\hat{C}$, is computed during the second sweep using Eq. 29. The results of which are then substituted into Eq. 30 in order to update the scaled concentration to the new time level.

4.1.7 Implicit Boundary Conditions

Boundary conditions are implemented through ghost or imaginary grid blocks positioned one cell outside the model domain (see Figure 2). All boundary conditions are treated implicitly. The implicit treatment is somewhat complicated in the x-sweep, because *intermediate* left and right boundary conditions must be formulated to compute the intermediate solution. The following is a derivation of the intermediate boundary condition for the left boundary on the x-sweep.

A Dirichlet, Neumann, or specified flux boundary condition on the left end of the model domain, may be expressed by the general equation,

$$\Delta\hat{C}_{0,k} = b_k \, \Delta\hat{C}_{1,k} + d_k \tag{31}$$

where the subscript k refers to the row number, $\Delta\hat{C}_{0,k}$ is the change in concentration in the ghost cell, $\Delta\hat{C}_{1,k}$ is the change in concentration in the first interior cell, and b_k and d_k are coefficients specific to the boundary condition type and value.

From Eq. 29, the intermediate solution at ghost cells on the left boundary is described by

$$\Delta\overline{C}_{0,k} = \left(I + \alpha_\eta L_{\eta\eta} + a_\eta S\right)_{0,k} \begin{pmatrix} \Delta\hat{C}_{0,k-1} \\ \Delta\hat{C}_{0,k} \\ \Delta\hat{C}_{0,k+1} \end{pmatrix} . \tag{32}$$

Substituting Eq. 31 into Eq. 32 yields

$$\Delta\overline{C}_{0,k} = \left(I + \alpha_\eta L_{\eta\eta} + a_\eta S\right)_{0,k} \begin{pmatrix} b_{k-1}\,\Delta\hat{C}_{1,k-1} + d_{k-1} \\ b_k \;\; \Delta\hat{C}_{1,k} \;\; + d_k \\ b_{k+1}\,\Delta\hat{C}_{1,k+1} + d_{k+1} \end{pmatrix} . \tag{33}$$

The coefficients b_{k-1}, b_k, and b_{k+1} are equivalent when the boundary condition is uniformly Dirichlet or Neumann. Under such conditions, the constant b can be factored out of Eq. 33 resulting in

$$\Delta\overline{C}_{0,k} = b\left(I + \alpha_\eta L_{\eta\eta} + a_\eta S\right)_{0,k} \begin{pmatrix} \Delta\hat{C}_{1,k-1} \\ \Delta\hat{C}_{1,k} \\ \Delta\hat{C}_{1,k+1} \end{pmatrix} + \left(I + \alpha_\eta L_{\eta\eta} + a_\eta S\right)_{0,k} \begin{pmatrix} d_{k-1} \\ d_k \\ d_{k+1} \end{pmatrix} . \tag{34}$$

Also, because the ghost cell grid spacing is equivalent to the boundary cell grid spacing, the linear operator in Eq. 34 evaluated at 0, k is equivalent to the linear operator evaluated at 1, k,

$$\left(I+\alpha_\eta L_{\eta\eta}+a_\eta S\right)_{0,k}=\left(I+\alpha_\eta L_{\eta\eta}+a_\eta S\right)_{1,k} \quad . \tag{35}$$

The equation for the intermediate boundary condition is obtained by substituting Eq. 35 and Eq. 32 into Eq. 34, resulting in

$$\Delta\overline{C}_{0,k}=b\,\Delta\overline{C}_{1,k}+\left(I+\alpha_\eta L_{\eta\eta}+a_\eta S\right)_{1,k}\begin{pmatrix} d_{k-1} \\ d_k \\ d_{k+1}\end{pmatrix} \quad . \tag{36}$$

A similar procedure is used to compute the intermediate boundary condition for the right side of the x-sweep.

Boundary conditions can be set automatically by SECOTP2D. This option compares the magnitude of the normal velocity component at each cell on the computational domain to a value ε that is taken to be a 0.1 percent of the maximum velocity. For cells in which the normal velocity components are below ε, the boundary conditions are set to Dirichlet with a zero concentration. Boundary conditions for the remaining cells are set by examining the flow field. If the normal velocity component is outward, the boundary condition is set to a Neumann condition with zero gradient and if flow is inward, the boundary condition is set to a Dirichlet condition with a zero concentration.

4.1.8 Matrix Equation

Because the matrix equation is one dimensional, the coordinate transformation implemented in the fracture equation is not imposed in matrix. The matrix equation, Eq. 3 is however, approximated using the same three-level implicit finite volume scheme discussed in the preceding section. The governing equation for matrix diffusion is restated here in a slightly modified form for the sake of convenience.

$$\phi' R'_k\left(\frac{\partial C'_k}{\partial t}\right)=\frac{\partial}{\partial\chi}\left(\phi' D'\frac{\partial C'_k}{\partial\chi}\right)-\phi' R'_k\lambda_k C'_k+\phi' R'_{k-1}\lambda_{k-1}C'_{k-1} \quad . \tag{37}$$

Applying the generalized three-level scheme [Ref. 10, pg. 229] to Eq. 37 yields

$$\phi' R'_{k,i}\frac{(1+\varphi)\Delta C'^{n}_{k,i}-\varphi\Delta C'^{n-1}_{k,i}}{\Delta t}=\theta\left(L_{\chi\chi}\Delta C'^{n}_{k,i}+\Delta Q'^{n}_{k,i}\right)+\left(L_{\chi\chi}C'^{n}_{k,i}+Q'^{n}_{k,i}\right) \tag{38}$$

where the subscript (k,i) refers to species k at node i, the superscript (n) refers to the time level, and

$$L_{\chi\chi}C'^{n}_{k,i} = \frac{2}{\Delta\chi_i}\left[(\phi' D')_{k,i+1/2}\frac{\left(C'^{n}_{k,i+1} - C'^{n}_{k,i}\right)}{\left(\Delta\chi_{i+1} + \Delta\chi_i\right)} - (\phi' D')_{k,i-1/2}\frac{\left(C'^{n}_{k,i} - C'^{n}_{k,i-1}\right)}{\left(\Delta\chi_i + \Delta\chi_{i-1}\right)}\right] \tag{39}$$

$$\Delta C'^{n}_{k,i} = C'^{n+1}_{k,i} - C'^{n}_{k,i} \tag{40}$$

$$L_{\chi\chi}\Delta C'^{n}_{k,i} = L_{\chi\chi}C'^{n+1}_{k,i} - L_{\chi\chi}C'^{n}_{k,i} \tag{41}$$

$$Q'^{n}_{k,i} = \left(-\phi' R'_k \lambda_k C'^{n}_{k} + \phi' R'_{k-1}\lambda_{k-1}C'^{n}_{k-1}\right)_i \tag{42}$$

$$\Delta Q'^{n}_{k,i} = Q'^{n+1}_{k,i} - Q'^{n}_{k,i} \tag{43}$$

Arranging Eq. 38 such that terms containing C'^{n+1}_{k} are grouped on the left hand side (LHS) and the remaining terms are found on the RHS results in

$$\begin{aligned}\phi' R'_{k,i}\Delta C'^{n}_{k,i} - \frac{\theta\Delta t}{(1+\varphi)}\left(L_{\chi\chi}\Delta C'^{n}_{k,i} - \phi' R'_{k,i}\lambda_k \Delta C'^{n}_{k,i}\right) \\ = \frac{\Delta t}{(1+\varphi)}\left(L_{\chi\chi}C'^{n}_{k,i} + Q'^{n}_{k,i}\right) + \\ \frac{\theta\Delta t}{(1+\varphi)}\left(\phi' R'_{k-1,i}\lambda_{k-1}\Delta C'^{n}_{k-1,i}\right) + \\ \frac{\varphi}{(1+\varphi)}\phi' R'_{k,i}\Delta C'^{n-1}_{k,i}\end{aligned} \tag{44}$$

Applying Eq. 44 yields a tridiagonal system of equations that can be solved efficiently using the Thomas algorithm [Ref. 12, pg. 183]. The efficiency of the inversion process is greatly enhanced when the time step is held constant. By maintaining a constant time step, the coefficients of the LHS remain constant, eliminating the need to regenerate and decompose the matrix. This savings has been found to reduce the computational overhead by a factor of two for a typical WIPP transport problem.

The above procedure is second-order accurate in space, and either first- or second-order accurate in time. The temporal discretization error is dependent on the selection of φ and θ as previously discussed (see Table 1).

4.1.9 Fracture-Matrix Coupling

The fracture and matrix continua are coupled through the mass transfer term, Γ_k. Defined by Eq. 4, Γ_k specifies the rate of mass transfer from the matrix to the fracture. This mass flux is

proportional to the concentration gradient in the matrix at the fracture/matrix interface. Γ_k can be evaluated at the n time level (explicit), or the n+1 time level (implicit). Huyakorn et al. [Ref. 8] investigated an explicit formulation and found that it often resulted in large mass conservation errors. He then developed a scheme to evaluate the mass transfer term implicitly for a finite element model.

A similar, implicit, coupling term is found in SECOTP2D. The implicit scheme is essentially identical to that of Huyakorn [Ref. 8]. The coupling procedure consists of three steps. **Step 1** involves writing the incremental mass transfer term $\Delta\hat{\Gamma}_k^n$ in a form that can be inserted into the implicit part of the fracture equation

$$\Delta\hat{\Gamma}_k^n = a\Delta\hat{C}_k^n + b \tag{45}$$

Step 2 involves the evaluation of a and b. This is accomplished using the inversion process (LU factorization) in the solution of the matrix equation. After the construction of the lower tridiagonal matrix L and the intermediate solution, there is enough information to evaluate the coefficients a and b. This new information is fed into the fracture equation which subsequently is solved for concentrations in the fracture at the new time level (n+1). **Step 3** involves the construction of the boundary condition for the matrix equation at the fracture-matrix interface using fracture concentrations at the (n+1) time level. Matrix concentrations are then obtained using the upper tridiagonal matrix U through back substitution.

A detailed discussion of the matrix/fracture coupling term and its implementation in SECOTP2D is provided in Appendix I.

4.1.10 Solution Accuracy

The present code uses TVD limiters with three-level time differencing and directional splitting to improve accuracy and execution time. The code is second-order accurate both in time and space with appropriate time and spatial discretizations; however, TVD limiters in the vicinity of a sharp spatial gradient such as a propagating front make the solution locally first-order accurate. The spatial accuracy of the SECOTP2D code with the TVD option on is less than second-order accurate, and the deviation from second-order accuracy will depend on how many sharp fronts exist in the computational plane. Problems with a moderately high Peclet number would greatly benefit from the TVD scheme by avoiding spurious oscillations commonly associated with the central differencing schemes. The long time scales of the problems to which the code is to be applied dictate the use of implicit algorithms.

The flow field is usually computed by SECOFL2D, but any comparable flow code can be used. It is important to note that the convergence tolerance on the flow must be smaller in magnitude than the *source* ($\hat{Q}$) for the transport calculation. A lack of convergence in the flow calculation can produce artificial source terms in the transport calculation and in some cases lead to instabilities.

5.0 CAPABILITIES AND LIMITATIONS OF THE SOFTWARE

Capabilities and limitations of SECOTP2D include the following:

- SECOTP2D can compute multiple component solute transport in fractured porous media using discrete-fracture and dual-porosity models.
- SECOTP2D allows for radioactive decay and generation of daughter products.
- For the fracture-with-matrix block system, transport in the fracture is produced by the combined effect of advection and hydrodynamic dispersion, while transport in the matrix block is assumed to be dominated by molecular diffusion.
- SECOTP2D is an implicit finite volume code that is second-order accurate in space and time. It uses operator splitting, an Approximate Factorization procedure, the delta formulation, and a finite volume staggered mesh.
- SECOTP2D models the advective terms using either Total Variation Diminishing (TVD), central differencing, or weighted upwind differencing. TVD eliminates unwanted oscillations near sharp gradients while maintaining high accuracy in the solution.
- SECOTP2D uses a generalized three-level scheme for time discretization.
- An Approximate Factorization technique is used to alleviate the shortcomings (large memory requirements and CPU time) of banded LU solvers. However, while Approximate Factorization reduces the operation count and memory usage for inverting the coefficient matrix, it does introduce complications in applying the implicit boundary conditions.
- SECOTP2D assumes the matrix porosity, fracture porosity, and fluid density remain constant in time. This assumption may be invalid for transient flow fields with large changes in hydraulic head.
- Velocity may vary in both time and space. Dispersivities, porosities, and retardation coefficients can vary spatially but are constant in time. Species decay constants are constant in time and space. Finally, sources and sinks are permitted to vary both temporally and spatially.
- Implicit coupling of the equations for the fracture and matrix block is used. This approach is more robust than explicit coupling.
- The spatial accuracy of the SECOTP2D code, with the TVD option on, is less than second-order accurate, and the deviation from second-order accuracy will depend on how many sharp fronts exist in the computational plane.
- SECOTP2D provides discharge calculations on predefined rectilinear boundaries.

6.0 USER INTERACTIONS WITH THE SOFTWARE AND EXAMPLES

To execute SECOTP2D, type SECOTP2D at the Alpha system "$" prompt. SECOTP2D will request the names of three files. Alternatively, the user may append the names of the three files (in the order listed below) to the SECOTP2D command line. The three files that the user must specify are listed below:

1. The input control file. This file specifies the processing options for SECOTP2D. Unlike the input control files for most of the WIPP PA codes, the user has no interactive control over the contents of this file. This file is generated by the upstream code PRESECOTP2D. All of the interaction with this file occurs through choices made in the input file to the preprocessor. This file in turn contains the names of the binary property and velocity files needed to run this code (see below).

2. The binary output data file. This .BIN file contains the output of SECOTP2D in binary format. This file is processed by POSTSECOTP2D.

3. The diagnostics/debug file. This file contains run-time information on the execution of SECOTP2D.

In addition, two other input files are required for SECOTP2D to run. These files, listed below, are specified by the user when the preprocessor is run. Their names are included in the input control file.

- Property data input file. This file contains the property information output from PRESECOTP2D in binary format, which is used to run SECOTP2D.

- Velocity data input file. This file contains the velocities and source information output from PRESECOTP2D in binary format used to run SECOTP2D.

6.1 Two-Dimensional Chain Decay Sample Problem

The following sample problem was obtained from the SECOTP2D software quality assurance test problem set. This test problem is referred to as test case #4 in the RD/VVP [5] and the VD [6]. Test case #4 is an appropriate problem for tutorial purposes, because of its simplicity, and because all the steps necessary to perform a SECOTP2D simulation are documented. The objective of this test is to verify that the radioisotope chain decay and mass-balance reporting are functioning properly in a two-dimensional setting. Also tested is the implementation of Neumann boundary conditions, and the setting of initial conditions.

The problem setup (Figure 3) consists of a 10- x 10-m problem domain, discretized into four equal cells. A single fracture, 0.01 m wide, passes through a 2-m thick porous matrix. The total thickness of the problem is 2.01 m.

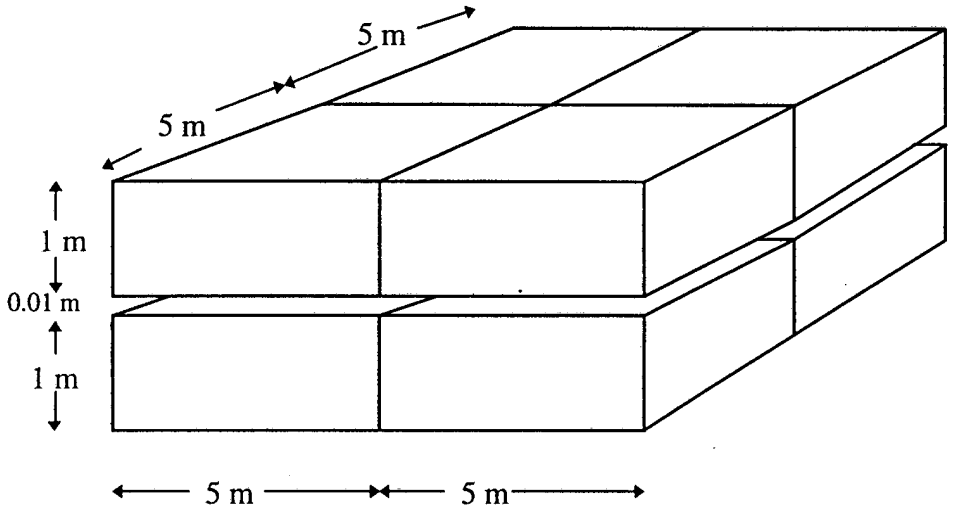


Figure 3. Diagram of the two-dimensional chain decay sample problem.

This is a rather simple problem, in which a single decaying isotope is placed in the fracture as an initial condition. With time, the isotope diffuses into the matrix and decays to a daughter product. The daughter species is generated in both the matrix and the fracture and decays itself.

The initial concentration of the parent species is set equal to one kg/m^3. The fracture dimensions are such that the initial mass of the parent is one kg. The matrix is initially void of the parent, and the daughter is not present in either the fracture or the matrix at t = 0. Also, the fracture velocity is zero and Neumann boundary conditions are used to prohibit mass from entering or leaving the problem domain.

The transport parameters, and initial and boundary conditions are

thickness, h = 2.01 m (1 fracture of 0.01 m and 2-1m blocks)
fracture aperture, b = 0.01 m
fracture porosity, $\phi_f = b/h = 0.01/2.01 = 4.9751244 \times 10^{-3}$
matrix porosity, $\phi_m = 0.10$
matrix block half length, L = 1 m
free water molecular diffusion coefficient, $D^* = 10^{-10}$ m^2/s
tortuosity, $\tau_m = \tau_f = 1$
dispersivity, $\alpha_l = \alpha_t = 0$ m
decay constants, $\lambda_1 = 6.931472 \times 10^{-10}$, $\lambda_2 = 3.465736 \times 10^{-10}$
retardation coefficient, $R_m = R_f = 1$
discharge velocity, $V_x = V_y = 0$ m/s
source-term splitting coefficients, $\alpha_x = 1$, $\alpha_y = 0$

Initial conditions:

fracture	$C_1(x,y,0) = 1.0$ kg/m^3
	$C_2(x,y,0) = 0.0$ kg/m^3
matrix	$C_1(x,y,z,0) = 0$ kg/m^3
	$C_2(x,y,z,0) = 0$ kg/m^3

Boundary Conditions:

fracture	$\partial C/\partial x = \partial C/\partial y = 0$
matrix	code specified

Time:

Simulation time	10^9 seconds
time step	10^7 seconds, constant
differencing	backward (Euler, set in PRESECOTP2D input file)

The total mass of the parent and daughter species (i.e. the mass in the fracture plus the mass in the matrix) can be compared with the analytical solution of Bateman [15]:

$$M_1(t) = M_1(0)e^{-\lambda_1 t} \tag{46}$$

$$M_2(t) = M_2(0)e^{-\lambda_2 t} + \lambda_1 M_1(0)[e^{-\lambda_1 t} - e^{-\lambda_2 t}]/[\lambda_2 - \lambda_1] \tag{47}$$

where:

M_1 and M_2 are the masses of the parent and daughter radionuclides, respectively,
t is time, λ_1 and λ_2 are decay constants for the parent and daughter, respectively.

6.1.1 Command Procedure

This sample problem can be run using the VMS command file, ST2D2_TEST4.COM (shown in Figure 4). This file is located in the directory WP$TESTROOT:[ST2D2.TESTCASES. TEST4] and can be executed by entering *@ST2D2_TEST4* at the VMS prompt. This command file performs the following functions:

- Runs GENMESH, which creates the finite grid
- Runs MATSET, which defines the materials used in the simulation
- Runs ICSET to set initial conditions
- Runs PRESECOTP2D, which creates the SECOTP2D input files
- Runs SECOTP2D
- Transfers the SECOTP2D results to a CDB file by running POSTSECOTP2D
- Extracts selected data from the CDB file using SUMMARIZE
- Runs a FORTRAN program from this directory to generate the analytical solution for the parent and daughter at comparable times (Figure 5); no input file is required.

```
$!*************************************************************************
$!
$! File: ST2D2_TEST4.COM - Command file for running Test Case #4 for
$!                         SECOTP2D Version 1.41 (Parent/daughter decay)
$!
$! Date: July 22, 1997
$!
$!*************************************************************************
$!
$! Reassign sys$output to record the screen messages
$ ASSIGN ST2D2_TEST4.LOG SYS$OUTPUT
$!
$! Go to the test directory and assign code symbols
$ SET DEFAULT       wp$testroot:[st2d2.testcases.test4]
$ PRESECOTP2D  == "$wp$prodroot:[st2d.exe]PRESECOTP2D_QA0122.EXE"
$ SECOTP2D     == "$wp$prodroot:[st2d.exe]SECOTP2D_QA0141.EXE"
$ POSTSECOTP2D == "$wp$prodroot:[st2d.exe]POSTSECOTP2D_QA0104.EXE"
$ GENMESH      == "$wp$prodroot:[GM.EXE]GM_pa96.EXE
$ MATSET       == "$wp$prodroot:[MS.EXE]MATSET_pa96.EXE
$ ICSET        == "$wp$prodroot:[IC.EXE]ICSET_pa96.EXE
$!
$! Set up the 2x2, 5m block matrix for the problem
$ GENMESH []ST2D2_TEST4_GENMESH.INP -
          ST2D2_TEST4_GENMESH.CDB -
          ST2D2_TEST4_GENMESH.DBG
$! Set up materials
$ MATSET []ST2D2_TEST4_GENMESH.CDB -
         []ST2D2_TEST4_MATSET.INP -
         CANCEL -
         CANCEL -
         []ST2D2_TEST4_MATSET.CDB -
         []ST2D2_TEST4_MATSET.DBG
$! Set initial conditions
$ ICSET   []ST2D2_TEST4_MATSET.CDB -
          ST2D2_TEST4_ICSET.INP -
          ST2D2_TEST4_INPUT.CDB -
          ST2D2_TEST4_ICSET.DBG
```

Figure 4. ST2D2_TEST4.COM - command file for running test case #4.

```
$!-------------------------------------------------------------------------
$! Run the SECO suite of codes for test case #4...
$!-------------------------------------------------------------------------
$ PRESECOTP2D []ST2D2_TEST4_INPUT.CDB -
     CANCEL -
     ST2D2_TEST4_PRESECO.INP -
     ST2D2_TEST4.INP -
     ST2D2_TEST4.PRP -
     CANCEL -
     ST2D2_TEST4.VEL -
     ST2D2_TEST4_PRESECO.DBG
$!
$ SECOTP2D ST2D2_TEST4.INP -
     ST2D2_TEST4.BIN -
     ST2D2_TEST4.OUT
$!
$ POSTSECOTP2D []ST2D2_TEST4_INPUT.CDB -
     ST2D2_TEST4.BIN -
     ST2D2_TEST4_POSTSECO.CDB -
     CANCEL
$ RENAME FOR023.DAT ST2D2_TEST4.DAT
$! Get time series of output variables (mass & mass balance)
$ SUMMARIZE ST2D2_TEST4_SUMMARIZE.INP
$! Create output file with analytical solutions for the parent
$! and Daughter; this file is named "ST2D2_TEST4_ANALYTIC.DAT":
$ RUN ST2D2_TEST4_DECAY2
$! Data from 4 "summarize" runs must be manually added to file
$! ST2D2_TEST4_ANALYTIC.DAT file with analytical solutions,
$! the matrix and fracture masses added for a total mass, etc.
$! We did this with EXCEL.
$ EXIT
```

Figure 4. ST2D2_TEST4.COM - command file for running test case #4 (continued).

```
C     Program to calculate analytical solution to the parent/daughter
C     double mass decay problem.
C     Program written 7-23-1997, Ross A. Wolford, GRAM, Inc.
      IMPLICIT REAL*8(A-H, O-Z)
      INTEGER*4 nmax, i
C     Initial mass of species #1 (parent):
      aM10      = 1.
C     Decay constants for species 1&2, respectively:
      aL1       = 6.931472E-10
      aL2       = 3.465736E-10
C     Initial mass of Daughter product:
      aM20      = 0.
      nmax      = 101
      t = -1.E+7
      OPEN(UNIT=8,FILE='ST2D2_TEST4_ANALYTIC.DAT',STATUS='unknown')
      WRITE(8,11) nmax
11    FORMAT( ' # of time increments used=', I8//
     & '            t        M1        M2' )
      DO 100 i = 1, Nmax
           t = t + 1.E+7
C          aM1 = remains of IC species 1
           aM1 = aM10 * DEXP( -aL1 * t )
C          aM2 = amount of species 2 after formation, decay
           aM2 = aM20 * DEXP( -aL1 * t) +
     &     (aL1*aM10) * (DEXP( -aL1*t ) - DEXP( -aL2*t ) ) /
     &     ( aL2 - aL1 )

           WRITE(8,90) t, aM1, aM2
90         FORMAT( E12.6, 4F10.7 )
100   CONTINUE
      CLOSE(UNIT=8)
      STOP
      END
```

Figure 5. ST2D2_TEST4_DECAY2.FOR - FORTRAN program that generates the analytical solution for test case #4.

6.1.2 Input/Output Files

This test case requires the following input files and outputs the following output files:

User-Created Input Files

- ST2D2_TEST4_GENMESH.INP - GENMESH user-created input file (Figure 6) used to set up the finite grid. This includes the number of nodes, matrix dimensions, and block size. GENMESH creates an input file (ST2D2_TEST4_GENMESH.CDB, computational data base binary format) for program MATSET.
- ST2D2_TEST4_MATSET.INP - user-created input file for MATSET (Figure 7) used to specify material properties including dispersivity, tortuosity, porosity, retardation, and distance between fractures (i.e., ½ block length). MATSET creates an input file (ST2D2_TEST4_MATSET.CDB, in computational data base binary format) for program ICSET.
- ST2D2_TEST4_ICSET.INP - user-created input file for ICSET (Figure 8) used to specify initial conditions of velocity, concentration, and time. ICSET creates an input file (ST2D2_TEST4_INPUT.CDB, in computational data base binary format) for program PRESECOTP2D.
- ST2D2_TEST4_PRESECO.INP - user-created input file for PRESECOTP2D that defines the SECOTP2D input files and sets many parameters (Figure 9).
- ST2D2_TEST4_SUMMARIZE.INP - user-created input file for SUMMARIZE used to extract data for species 1 and species 2 from the output CDB file. These data include the mass in the fracture and matrix, the mass flux across the problem boundaries, the mass decayed, the mass change due to sources and sinks, and the mass balance error (Figure 10).

Program-Created ASCII Input Files

- ST2D2_TEST4.INP - Input file for SECOTP2D that is written by PRESECOTP2D

ASCII Output Files

- ST2D2_TEST4.DAT - Contains the SECOTP2D concentration results for two species as a function of time and distance into the matrix. This file is output by SECOTP2D as FOR023.DAT and renamed by the command file; it is not used in this evaluation.
- ST2D2_TEST4.OUT - Summary run information for SECOTP2D that normally would be displaced on the screen as the model runs.
- ST2D2_TEST4_SUMMARIZE.OUT - This file contains the SECOTP2D results for species 1 and species 2. These include the mass in the fracture, mass in the matrix, mass flux across the boundary, mass decayed, the mass balance error, and the mass change from sources and sinks. Referred to as the SUMMARIZE output file.

```
!=============================================================================
!   FILETYPE: GENMESH input text file
!   DATE:     23-JUL-1997
!=============================================================================
!
  *SETUP
    DIM=      2
    ORIGIN= 0.0, 0.0
    IJKMAX= 3, 3
  *GRID
!
!============================== X direction ==============================
!
    DEL, COORD=X, DEL=5.0, INRANGE=1,3, FACTOR= 1.0
!
!============================== Y direction ==============================
!
    DEL, COORD=Y, DEL=5.0, INRANGE=1,3, FACTOR= 1.0
!
!============================ Material Regions ===========================
!
  *REGIONS
!
   REGION= 1, IRANGE= 1,3, JRANGE= 1, 3
!=============================================================================
!
  *ELEVATION
!
   LOCATION, THICK=2.01, ELEV=0.0, IRANGE=1,3, JRANGE=1,3
!=============================================================================
!
  *END
```

Figure 6. ST2D2_TEST4_GENMESH.INP - GENMESH input file for test case #4.

```
!=========================================================================
!   FILETYPE: MATSET input text file
!   DATE:     23-JUL-1997
!   PURPOSE:  Define material and property names and selected values
!             that are not in the PROPERTY.SDB
!=========================================================================
!
 *PRINT_ASSIGNED_VALUES
!
 *HEADING
   TITLE, DUAL POROSITY MASS CONSERVATION STUDY
   SCALE, LOCAL
   SCENARIO, DUAL SPECIES DECAY
!
 *UNITS=SI
 *RETRIEVE
!
   COORD, DIM=2, NAMES=X,Y
!
! ...Define region names
!
   MATERIAL, 1=CULEBRA
!
!...Define Material property names
!
  PROPERTY, MATerial=CULEBRA, NAMES=RM_PU, RF_PU, RM_SP, RF_SP, &
              HMBL, SKINREST
!
!...Define Material attribute names
!
  ATTRIBUTE, MATerial=CULEBRA, NAMES=DISP, MTORT, FTORT, &
             MPOROS, FPOROS
!
 *SET_VALUES
!
!#### Assign values to material property names not   ####
!#### found in the Secondary Database (PROPERTY.SDB) ####
!
   PROPERTY, MATerial=CULEBRA, NAMES*VALUES: RF_PU=1.0, RM_PU=1.0, &
             RM_SP=1.0, RF_SP=1.0, HMBL=1.0, SKINREST=0.0
!
   ATTRIBUTE, MATerial=CULEBRA, NAMES*VALUES:  DISP=0.0, &
             MTORT=1.0, FTORT=1.0, MPOROS=0.10, FPOROS=4.9751244E-3
!
 *END
```

Figure 7. ST2D2_TEST4_MATSET.INP - MATSET input file for test case #4.

```
!=========================================================================
!   FILETYPE: ICSET input text file
!   ANALYSTS: Ross A. Wolford, GRAM, Inc.
!   DATE:     23-JUL-1997
!   PURPOSE:  Set initial conditions
!=========================================================================
!
 *SET_NAMES
   INITIAL_NAMES TYPE=ELEMENT, NAMES=VELX, VELY, CONC_PU, CONC_SP
!
 *SET_VALUES
!
!-------------------------------------------------------------------------
! Define flow field
!-------------------------------------------------------------------------
!
   INITIAL_VALUE, TYPE=ELEMENT, NAME=VELX, &
                  IRANGE=1,3, JRANGE=1,3, VALUE=0.0E+00
!
   INITIAL_VALUE, TYPE=ELEMENT, NAME=VELY, &
                  IRANGE=1,3, JRANGE=1,3, VALUE=0.0E+00
!
!-------------------------------------------------------------------------
! Define initial fracture concentration
!-------------------------------------------------------------------------
!
   INITIAL_VALUE, TYPE=ELEMENT, NAME=CONC_PU, &
                  IRANGE=1,3, JRANGE=1,3, VALUE=1.0E+00
!
   INITIAL_VALUE, TYPE=ELEMENT, NAME=CONC_SP, &
                  IRANGE=1,3, JRANGE=1,3, VALUE=0.0E+00
!
!-------------------------------------------------------------------------
! Define start time
!-------------------------------------------------------------------------
!
   INITIAL_VALUE, TYPE=TIME, VALUE=0.0E+00
!
!=========================================================================
 *END
```

Figure 8. ST2D2_TEST4_ICSET.INP - ICSET input file for test case #4.

```
!=============================================================================
! PRESECOTP2D Input file
! Created by Jim Ramsey, SNL; Modified by Ross Wolford, GRAM, Inc.
! Jul 23, 1997
!=============================================================================
*CONTROL
    MEDIUM=DUAL
    TIME_SCHEME=EULER
    LIMITER=CENTERED
    SOURCE_COEFF, AX=1.0, AY=0.0
!-----------------------------------------------------------------------------
*VELOCITY
    X_DARCY=VELX
    Y_DARCY=VELY
    FLOW_CODE=OTHER
    STEADY=YES
!-----------------------------------------------------------------------------
*OUTPUT
    STEP=1
    SCREEN_IO=off
    PRINT_MATRIX, I=1, J=1
    DISCHARGE_STEP=1
!-----------------------------------------------------------------------------
*TIME
   NUM_STEP=100
   TIME_GEN=AUTO
   START_TIME=0.0
   STOP_TIME=1.E+9
!-----------------------------------------------------------------------------
*SPECIES
 NUCLIDE SYMBOL=PU239, INDEX=1, LAMBDA=6.931472E-10, FREE_H2O_DIFF=1.E-10, &
         CURIE=1.0
 NUCLIDE SYMBOL=SP002, INDEX=2, LAMBDA=3.465736E-10, FREE_H2O_DIFF=1.E-10, &
         CURIE=1.0
 CHAIN CHAIN_NUM=1  NUM_SPECIES=2  NUC_INDICES=1,2
!-----------------------------------------------------------------------------
*PROPERTY
 DIFF   TORT=MTORT, POROSITY=MPOROS, RETARD=RM
 DUAL   BLOCK_LEN=HMBL SKIN_RESIST=SKINREST
 ADVEC  DISP_LNG=DISP, DISP_TRN=DISP, TORT=FTORT, &
        POROSITY=FPOROS, RETARD=RF
!-----------------------------------------------------------------------------
*INITIAL_CONDITIONS
  TIME_STEP=1
  DEF    NAME=CONC, SYMBOL=PU239
  DEF    NAME=CONC, SYMBOL=SP002
!-----------------------------------------------------------------------------
*BOUNDARY_CONDITIONS
 TOP    TYPE=NEUMAN, SYMBOL=PU239, NRANGE=1,2
 BOTTOM TYPE=NEUMAN, SYMBOL=PU239, NRANGE=1,2
 LEFT   TYPE=NEUMAN, SYMBOL=PU239, NRANGE=1,2
 RIGHT  TYPE=NEUMAN, SYMBOL=PU239, NRANGE=1,2
 TOP    TYPE=NEUMAN, SYMBOL=SP002, NRANGE=1,2
 BOTTOM TYPE=NEUMAN, SYMBOL=SP002, NRANGE=1,2
 LEFT   TYPE=NEUMAN, SYMBOL=SP002, NRANGE=1,2
 RIGHT  TYPE=NEUMAN, SYMBOL=SP002, NRANGE=1,2
!-----------------------------------------------------------------------------
*DP_MESH
 AUTO INIT_DIST=1.E-2 NUM_NODES=21
!-----------------------------------------------------------------------------
*DISCHARGE_BOUND
   NUM_BNDS=1
   BOUND_DEF TOP_LEFT=1,2, BOTTOM_RIGHT=2,1
!-----------------------------------------------------------------------------
*END
```

Figure 9. ST2D2_TEST4_PRESECO.INP - PRESECOTP2D input file for test case #4.

```
*input files
   name=ST2D2_TEST4_POSTSECO
   disk=WP$TESTROOT
   directory=[ST2D2.TESTCASES.TEST4]
   type=CDB

*times
   read=   seconds
   input=  seconds
   output= seconds
      steps=  all

*items
   type= history
        names= MSFPU239, &
               MSMPU239, &
               MSBPU239, &
               MSDPU239, &
               MSEPU239, &
               MSSPU239, &
               MSFSP002, &
               MSMSP002, &
               MSBSP002, &
               MSDSP002, &
               MSESP002, &
               MSSSP002

*output
   driver= text
   write=time vs item
   name=ST2D2_TEST4_SUMMARIZE.OUT
*end
```

Figure 10. ST2D2_TEST4_SUMMARIZE.INP - SUMMARIZE input file, test case #4.

- ST2D2_TEST4_ANALYTIC.DAT - Contains the analytical solution for Species 1 and Species 2. This file is generated by ST2D2_TEST4_DECAY2.EXE.

Binary Output/Input Files

- ST2D2_TEST4_POSTSECO.CDB - CAMDAT database file containing the SECOTP2D results. This is the primary output data file, referred to hereafter as the Output CDB File. It is created by POSTSECOTP2D and read by SUMMARIZE.
- ST2D2_TEST4_GENMESH.CDB - File containing finite mesh information. It is created by GENMESH and read by MATSET.
- ST2D2_TEST4_MATSET.CDB - File containing material information. It is created by MATSET and read by ICSET.
- ST2D2_TEST4_INPUT.CDB - CDB file which defines the grid and corresponding properties; it is created by "ICSET" and read by PRESECOTP2D.
- ST2D2_TEST4.BIN - Binary output data file from SECOTP2D containing concentration data.
- ST2D2_TEST4.PRP - Binary file containing property variables that is written by PRESECOTP2D and read by SECOTP2D.
- ST2D2_TEST4.VEL - Binary file containing velocity data that is written by PRESECOTP2D and read by SECOTP2D.

6.1.3 Sample Problem Results

The fracture concentration of the parent and daughter species at grid block locations (1,1) and (2,2) are presented in Figures 11 and 12, respectively. The solution at both locations is essentially identical because of the symmetry of the test problem. Note the drop in concentration of species 1 (parent) as a result decay and matrix diffusion, as well as the gain of species 2 (daughter), also a consequence of species 1 decay.

The matrix concentrations for the parent and daughter species at the same two locations are given in Figures 13 and 14, respectively. Once again the solution is symetric and the results appear to be reasonable.

SECOTP2D outputs the integrated quantities of mass (in space and time) contained in the matrix, contained in the fracture, and lost or gained as a result of decay. These quantities are used in the computation of mass balance error, however, they can also be used as analysis tools.

In this problem, the integrated results are used to demonstate the chain decay feature of the code is properly implemented. This is accomplished by comparing the integrated mass results with the Bateman equations. Such a comparison is presented in Figure 15 for Species 1 (parent) and Figure 16 for Species 2 (daughter).

Figure 15 shows the parent isotope starting with a total mass of 1.0 kg positioned in the fracture. After one half-life (10^9 seconds), 0.5 kg remain. In Figure 16, the mass of the daughter isotope increases throughout time from an initial mass of zero. Both integrated results agree favorably with the Bateman equations.

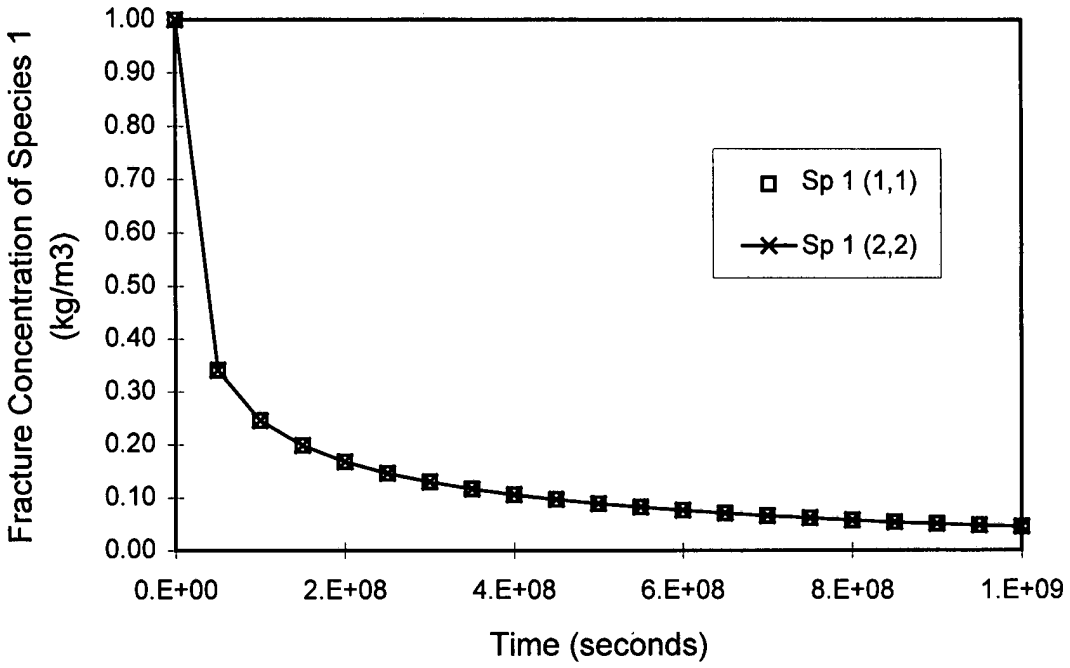


Figure 11. Fracture concentration of species 1 (parent).

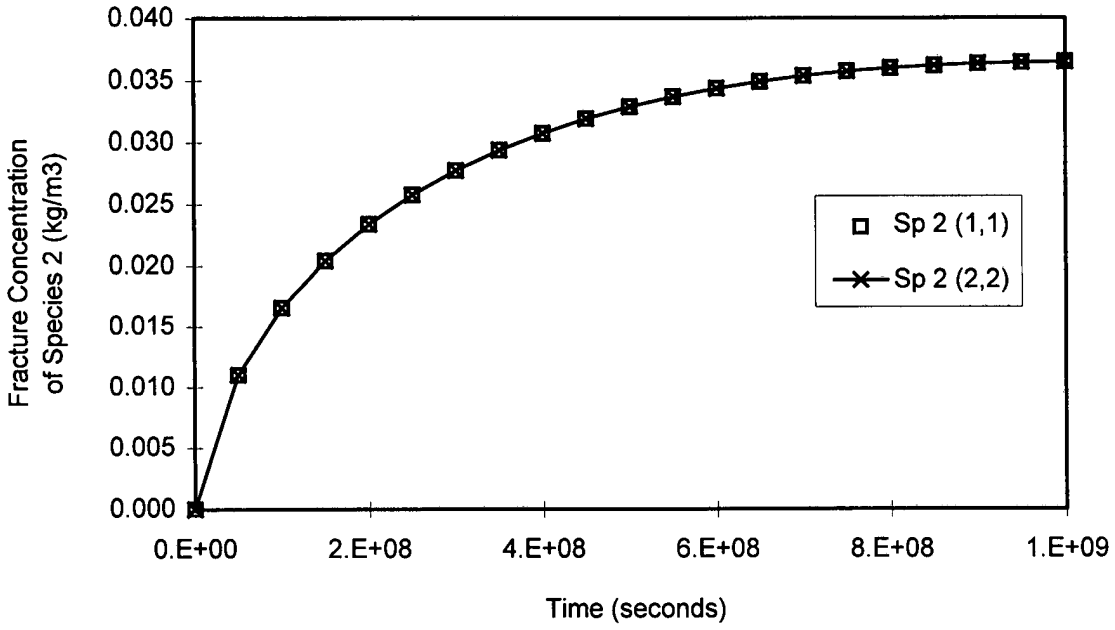


Figure 12. Fracture concentration of species 2 (daughter).

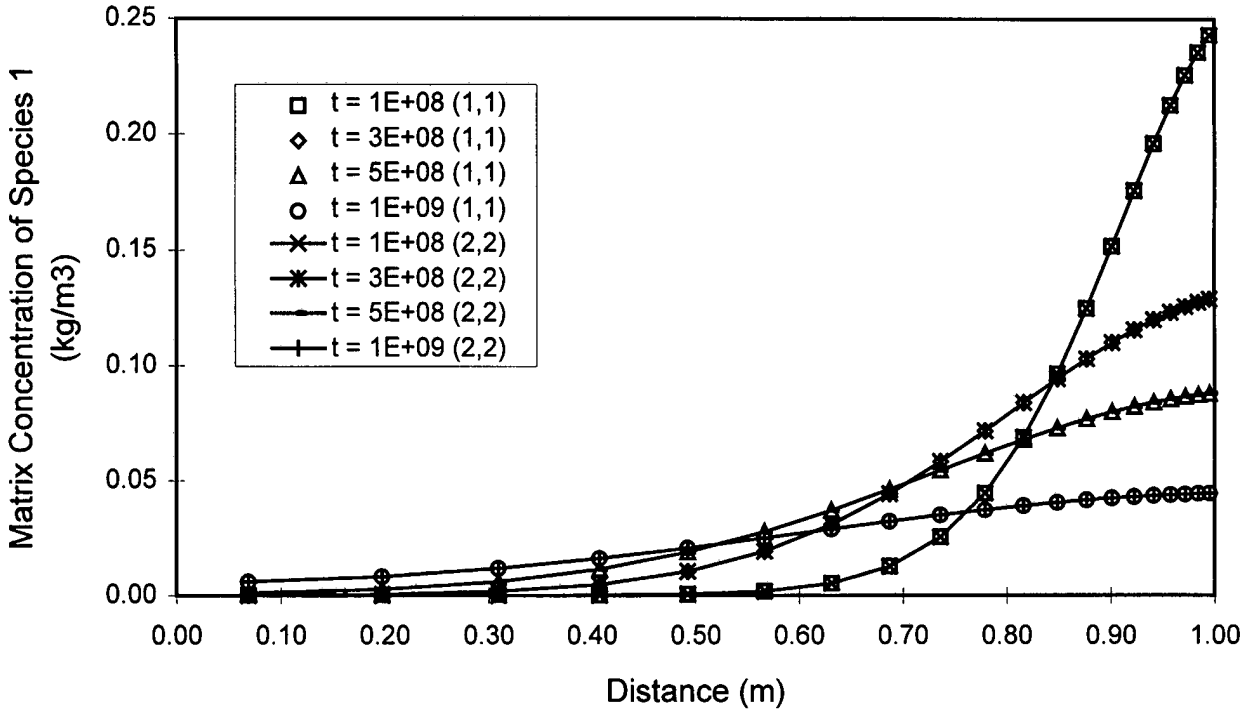


Figure 13. Matrix concentration of species 1 (parent).

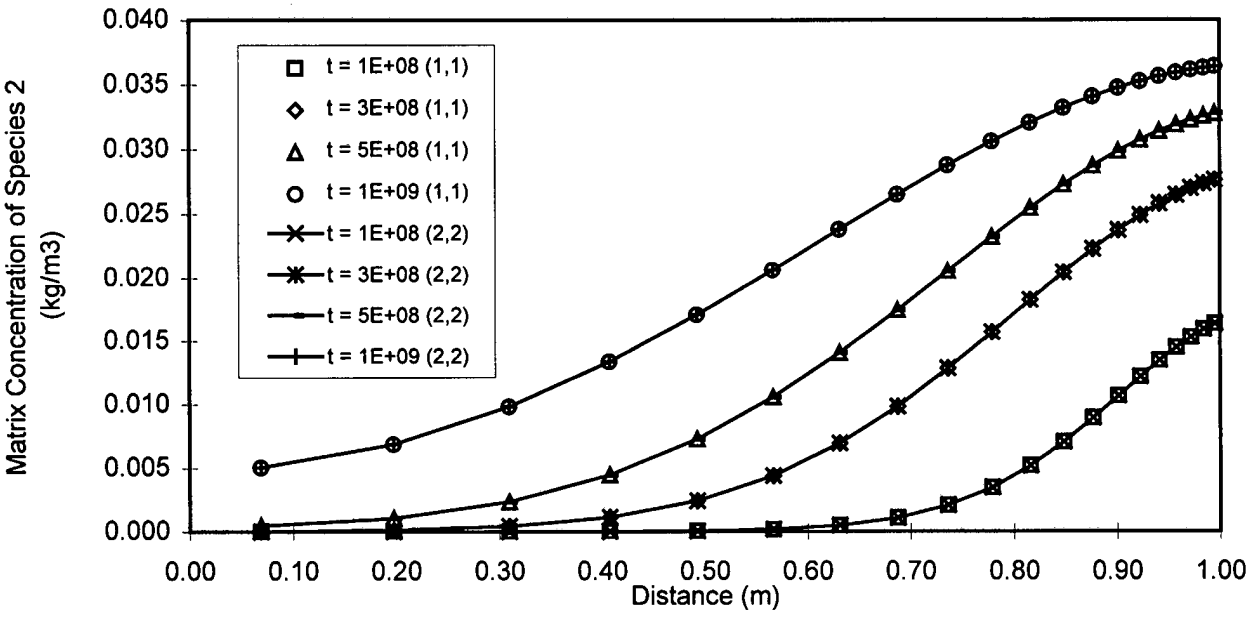
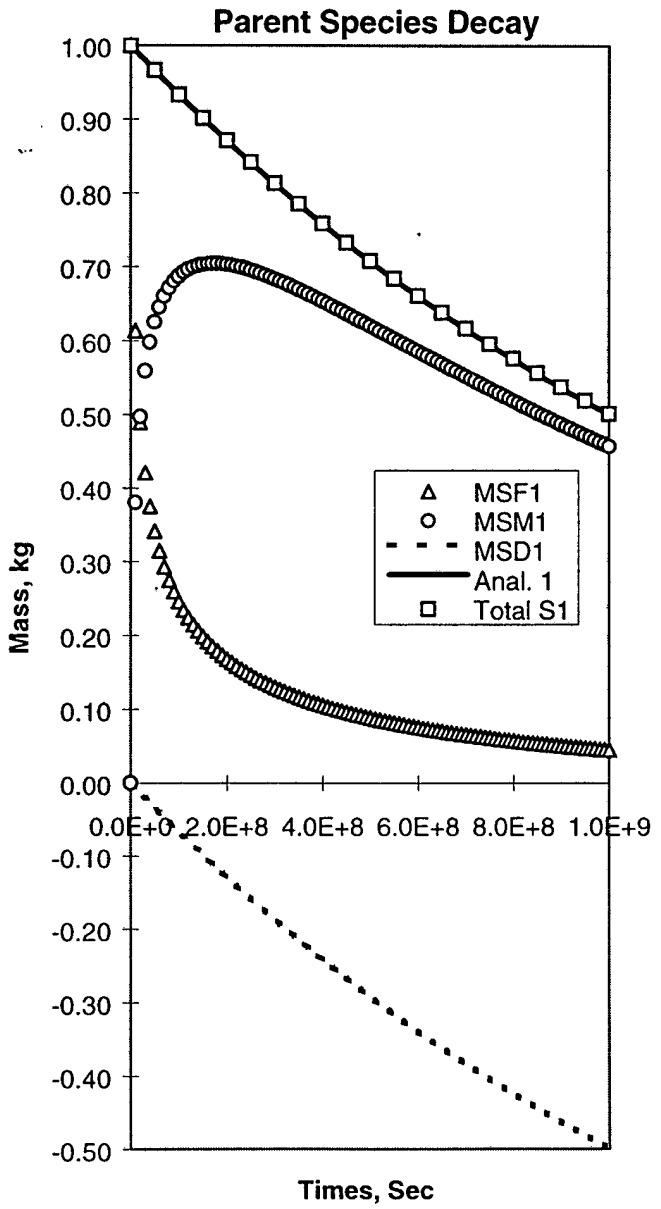


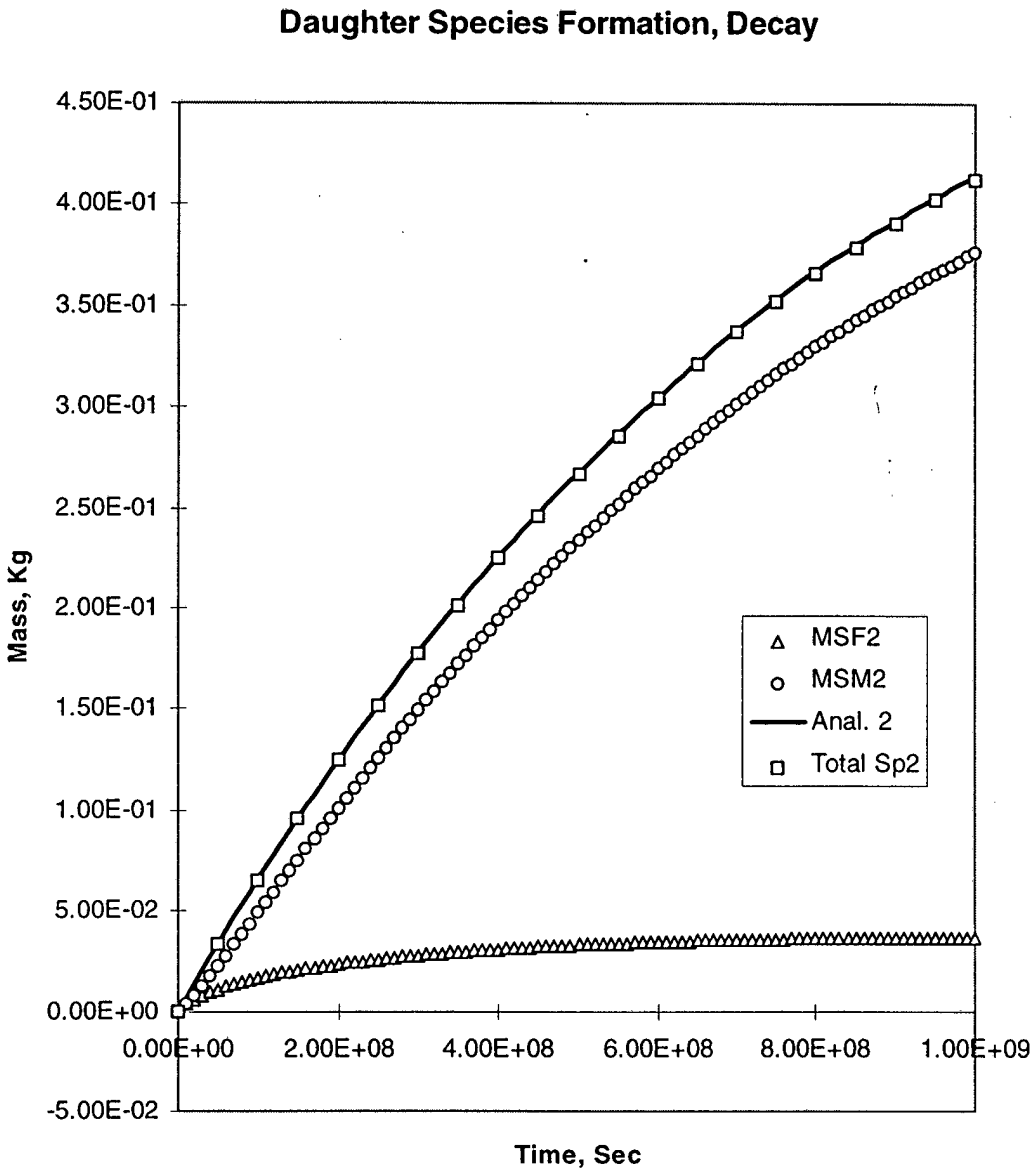
Figure 14. Matrix concentration of species 2 (daughter).



MSF1 = mass in fracture MSM1 = mass in matrix MSD1 = decayed mass

Anal. 1 = analytical solution Total S1 = MSF1 + MSM1

Figure 15. Radioactive chain decay - species 1, test case #4.



MSF2 = mass in fracture MSM2 = mass in matrix

Anal. 2 = analytical solution Total Sp2 = MSF2 + MSM2

Figure 16. Radioactive chain decay - species 2, test case #4.

7.0 DESCRIPTION OF INPUT FILES

7.1 Input Control File

The input control file to SECOTP2D is generated by PRESECOTP2D based in part on its own input control file. For all practical purposes, the code-generated input control file to SECOTP2D is transparent to the user.

7.2 Property Data Input and Velocity Data Input Files

The property data input file, which contains CAMDAT property information, is binary and is not user-specified. It is therefore transparent to the user.

The velocity data input file, which contains CAMDAT velocities and source information, is binary and is not user-specified. It is therefore transparent to the user.

8.0 ERROR MESSAGES

SECOTP2D uses the following lines of code to report error messages. Errors cause program execution to abort.

- These write statements generate an error message indicating that the corner point of discharge surface is not in the computational domain.

```
      WRITE(6,*) 'SECO2D-TRANSPORT: ERROR, Subroutine VEL.'
      WRITE(6,*) 'Corner point of discharge surface is out of',
        >                  ' computational domain.'
```

- These write statements generate an error message indicating that location of a source point is not in the computational domain.

```
   WRITE(6,*) 'SECO2D-TRANSPORT: ',
        >            'ERROR, Subroutine SOURCE.'
                WRITE(6,*) 'Location of a source point is out',
        >            ' of computational domain.'
```

- These write statements generate an error message indicating that an end-of-file has been detected.

```
   WRITE(6,*) 'SECO2D-TRANSPORT:'
              WRITE(6,*) 'ERROR, End-of-file detected (velocity)'
```

9.0 DESCRIPTION OF OUTPUT FILES

9.1 Binary Output File

The binary output file contains the output of SECOTP2D in binary format. Because it is binary, it cannot be read by the user.

9.2 Diagnostics/Debug Output File

A sample diagnostics/debug file, the only output file from SECOTP2D that can be read by the user, is provided in Appendix II.

10.0 REFERENCES

[1] Roache, P. J., "Computational Fluid Dynamics Algorithms Developed for WIPP Site Simulations," *Proceedings: IX International Conference on Computational Methods in Water Resources, Denver, Colorado, 9-12 June 1992*, T. Russell, et al., eds., pp. 375-382.

[2] Roache, P. J., "Computational Fluid Dynamics Algorithms and Codes Developed for WIPP Site Simulations," *Proceedings: Asian Pacific Conference on Computational Mechanics, 11-13 December 1991*, University of Hong Kong, J.H.W. Lee, et al., eds., H. Balkeema, Amsterdam.

[3] Salari, K., Knupp, P., Roache, P., and Steinberg, S., "TVD Applied to Radionuclide Transport in Fractured Porous Media," *Proceedings: IX International Conference on Computational Methods in Water Resources, Denver, Colorado, 9-12 June 1992*, T. Russell, et al., eds., pp. 141-148.

[4] Roache, P. J., "The SECO Suite of Codes for Site Performance Assessment," *Proceedings: 1993 Intl. High-Level Radioactive Waste Management Conf., April 26-30, 1993*, Las Vegas, NV.

[5] WIPP PA Requirements Document and Verification and Validation Plan for SECOTP2D (Version 1.41). WPO #45732, Sandia WIPP Central Files, June 9, 1997.

[6] WIPP PA Validation Document for SECOTP2D (Version 1.41). WPO #45735, Sandia WIPP Central Files, June 9, 1997.

[7] Huyakorn, P. S. and Pinder, G. F., *Computational Methods in Subsurface Flow,* Academic Press, New York, 1983.

[8] Huyakorn, P. S., Lester, B. H., and Mercer, J. W. "An Efficient Finite Element Technique for Modeling Transport in Fractured Porous Media: Single Species Transport," *Water Res.,* Vol. 19, No. 3, pp. 841-854, 1983.

[9] Bear, J. and Bachmat, Y. *Introduction to Modeling of Transport Phenomena in Porous Media,* Kluwer Academic Publishers, Dordrecht, Netherlands, 1990.

[10] Scheidegger, A. E., *The Physics of Flow Through Porous Media,* University of Toronto Press, 1974.

[11] Streltsova-Adams, T. D., "Well Hydraulics in Heterogeneous Aquifer Formations," *Advances in Hydroscience,* Vol. 11, (Ed., Chow, V.T.), pp. 357-423 Academic Press, New York, 1978.

[12] Fletcher, C. A. J., *Computational Techniques for Fluid Dynamics,* Vol. I and II, Springer-Verlag, 1988.

[13] Sweby, P.K., "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws", *SIAM J. Numer. Anal.* 21: 995-1011.

[14] Ramsey, J., and A. Treadway. Test of Mass Balance of SECOTP2D, WPO# 44700, Sandia WIPP Central Files, 1997, pg 22-23.

[15] Bateman, H., 1910. "The solution of a system of differential equations occurring in the theory of radio-active transformations." *Proc. Cambridge Phil. Soc.*, 16, 423.

11.0 APPENDICES

The following section provides the appendices for this document.

Appendix I. Derivation of the implicit fracture-matrix coupling procedure employed by SECOTP2D.

The partial differential equation (PDE) solved by SECOTP2D to predict radioisotope transport of the k^{th} species within the matrix portion of a dual-porosity system is given by Salari et al. (1996) as

$$\frac{\partial}{\partial x}\left(D\frac{\partial C_k}{\partial x}\right) = \phi_m R_k\left(\frac{\partial C_k}{\partial t}\right) + \phi_m R_k \lambda_k C_k - \phi_m R_{k-1}\lambda_{k-1}C_{k-1} \tag{1}$$

where, ϕ_m (dimensionless) is the matrix porosity defined as the matrix pore volume per unit volume of matrix, D (m^2/s) is the effective matrix diffusion coefficient, R (dimensionless) is the matrix retardation coefficient, λ (1/s) is the decay constant, C (kg/m^3) is the isotope concentration, and x and t are the spatial and temporal coordinates, respectively. D in Eq. 1 is defined as $D = \tau\,\phi_m\,D^*$, where τ (dimensionless) is the matrix tortuosity and D^* (m^2/s) is the free water molecular diffusion coefficient.

The spatial coordinate system is defined such that x = 0 at the point of symmetry in the center of the matrix block, and x = L at the interface between the matrix and fracture.

The initial and boundary conditions imposed on Eq. 1 are

$$C_k(x,0) = 0 \tag{2}$$

$$\frac{\partial C_k}{\partial x}(0,t) = 0 \tag{3}$$

$$C_k(L,t) = \tilde{C}_k \tag{4}$$

where, L (m) is half the matrix block length, and $\tilde{C}_k$ (kg/m^3) is the concentration of species k in the fracture.

The equations for the matrix and fracture continua are coupled through a mass transfer term, Γ_k, and the application of Fick's law at the interface between the two continua. Γ_k appears explicitly in the fracture equation, and can be expressed as a function of C_k in the matrix. Fick's Law applied at the interface between the matrix and fracture continua of a parallel plate dual porosity system yields

$$\Gamma_k = -\frac{2\phi_f}{b}\left(D\frac{\partial C_k}{\partial x}\bigg|_{x=L}\right) \tag{5}$$

where, ϕ_f (dimensionless) is the porosity of the fracture continuum, and b is the fracture aperture in the parallel plate formulation (m), defined by

$$b = \frac{\phi_f L}{1-\phi_f} \tag{6}$$

In formulating SECOTP2D, the governing PDEs are transformed from physical space (Cartesian coordinate system) to computational space (stretched coordinate system) in what is referred to as a strongly conservative fashion. In the transformation process, C_k is scaled by the determinate of the transformation Jacobian and additional terms associated with the transformation appear in the PDE. However, for the purpose of this discussion it is not necessary to consider the transformation or the transformation metrics other than to recognize they do exist in the code. Furthermore, the arguments presented here are independent of the number of species considered, so the discussion is limited to a single species and the subscript k has been dropped from this point forward.

Following the coordinate transformation, the resulting PDEs are discretized using a general three-level implicit finite volume scheme (Fletcher, 1987, p. 229, 317). The matrix equation is discretized spatially using a block-centered approach as shown in Figure 1. Imaginary or "ghost cells" are located at both ends of the matrix block to impose the boundary conditions stipulated in Eq. 3 and Eq. 4.

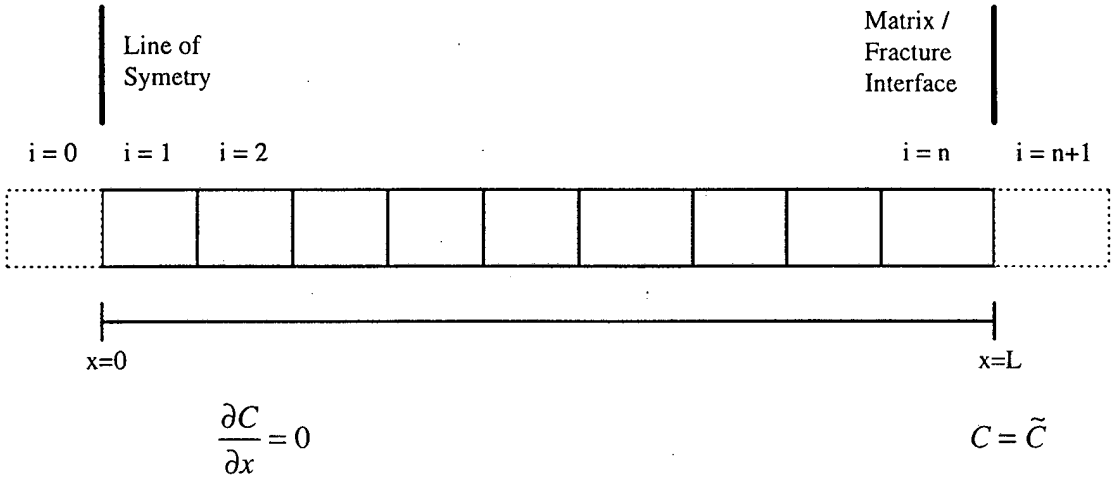


Figure 1. Matrix discretization.

The transformation and discretization process results in a set of algebraic equations of the form

$$A^{\ell}\Delta C^{\ell} = F^{\ell-1} \tag{7}$$

where, the superscript ℓ denotes the time level, ΔC^{ℓ} is an n+2 vector array containing the elements $\Delta C_i^{\ell} = C_i^{\ell} - C_i^{\ell-1}$, $F^{\ell-1}$ is an n+2 vector array containing the forcing elements, and A^{ℓ} is a tridiagonal matrix possessing a structure of

$$A = \begin{bmatrix} a2_o & a3_o & 0 & 0 & 0 & 0 & 0 \\ a1_1 & a2_1 & a3_1 & 0 & 0 & 0 & 0 \\ 0 & a1_2 & a2_2 & a3_2 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & a1_n & a2_n & a3_n \\ 0 & 0 & 0 & 0 & 0 & a1_{n+1} & a2_{n+1} \end{bmatrix} \tag{8}$$

The linear system in Eq. 7 is solved at each time step once for each element of the fracture domain by LU decomposition. The advantage of using LU decomposition is that at the end of the forward elimination process there is enough information available to express the mass transfer term, $\Delta\Gamma^{\ell}$, as a linear function of the fracture concentration,

$$\Delta\Gamma^{\ell} = \gamma\, \Delta\tilde{C}^{\ell} + \delta \tag{9}$$

Once the coefficients γ and δ in Eq. 9 are determined, $\Delta\Gamma^{\ell}$ is eliminated from the system of equations developed from the fracture PDE. The fracture system is then solved yielding $\Delta\tilde{C}^{\ell}$ at each point in the two-dimensional fracture domain. The time step is completed by solving for the concentration in the matrix through back substitution in which $\Delta\tilde{C}^{\ell}$ is used as a boundary condition.

Because LU decomposition is fundamental to the implicit coupling procedure, a brief presentation of the solution method follows:

$$\mathrm{L\,U} = \mathrm{A}^{\ell} \tag{10}$$

$$\mathrm{L\,G} = \mathrm{F}^{\ell-1} \qquad \text{forward elimination} \tag{11}$$

$$\mathrm{U}\,\Delta\mathrm{C}^{\ell} = \mathrm{G} \qquad \text{back substitution} \tag{12}$$

where,

$$L = \begin{bmatrix} \alpha_o & 0 & 0 & 0 & 0 & 0 & 0 \\ a1_1 & \alpha_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & a1_2 & \alpha_2 & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & a1_n & \alpha_n & 0 \\ 0 & 0 & 0 & 0 & 0 & a1_{n+1} & \alpha_{n+1} \end{bmatrix} \tag{13}$$

$$U = \begin{bmatrix} 1 & \beta_o & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \beta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \beta_2 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 1 & \beta_n \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$$\alpha_o = a2_o \quad (15)$$

$$G_o = F_o / \alpha_o \quad (16)$$

$$\alpha_i = a2_i - a1_i \beta_{i-1}, \quad \text{for } i = 1, n+1 \quad (17)$$

$$\beta_i = a3_i / \alpha_i, \quad \text{for } i = 0, n \quad (18)$$

$$G_i = (F_i - a1_i G_{i-1}) / \alpha_i, \quad \text{for } i = 1, n+1 \quad (19)$$

From Eq. 5 and the relationship $\Delta\Gamma^{\ell} = \Gamma^{\ell} - \Gamma^{\ell-1}$, it can be shown that

$$\Delta\Gamma^{\ell} = \frac{2\phi_f D}{b}\left(\frac{\partial C^{\ell}}{\partial x} - \frac{\partial C^{\ell-1}}{\partial x}\right)\Bigg|_{x=L} \quad (20)$$

By approximating the partial derivatives in Eq. 20 with the centered finite difference formula

$$\left.\frac{\partial C}{\partial x}\right|_{x=L} = \frac{2(C_{n+1} - C_n)}{(\Delta x_{n+1} + \Delta x_n)} \quad (21)$$

and restricting $\Delta x_{n+1} = \Delta x_n$, a second-order approximation of Eq. 20 is obtained

$$\Delta\Gamma^{\ell} = \frac{-2\phi_f D}{b}\frac{1}{\Delta x_n}\left(\Delta C^{\ell}_{n+1} - \Delta C^{\ell}_n\right) \quad (22)$$

The expression $\Delta C^{\ell}_{n+1} - \Delta C^{\ell}_n$ in Eq. 22 is obtained by beginning the back substitution process of the LU decomposition. From Eqs. 12 and 14 it is clear that

$$\Delta C^{\ell}_{n+1} = G_{n+1} \quad (23)$$

$$\Delta C^{\ell}_n = G_n - \beta_n G_{n+1} \quad (24)$$

and it follows that

$$\Delta C^{\ell}_{n+1} - \Delta C^{\ell}_n = G_{n+1}(1 + \beta_n) - G_n \quad (25)$$

Substituting Eq. 25 into 22 yields

$$\Delta\Gamma^{\ell} = \frac{-2\phi_f D}{b}\frac{1}{\Delta x_n}\left(G_{n+1}(1+\beta_n) - G_n\right) \tag{26}$$

Eq. 26 is a second-order approximation of the mass transfer function in terms of the n and n+1 coefficients of the LU decomposition. Eq. 26 takes the form of Eq. 9 when the term G_{n+1} is expressed using Eq. 19, and the Dirichlet boundary condition given by Eq. 4 is imposed on the solution. From Eq. 19, G_{n+1} can be expressed as

$$G_{n+1} = \frac{\left(F_{n+1} - a1_{n+1} G_n\right)}{\alpha_{n+1}} \tag{27}$$

A second-order finite difference formula is used to represent the Dirichlet boundary condition

$$\Delta\tilde{C}^{\ell} = \frac{\Delta C^{\ell}_{n+1} + \Delta C^{\ell}_{n}}{2} \tag{28}$$

and the boundary condition is imposed on the solution by setting $a1_{n+1} = a2_{n+1} = 1$ in Eq. 8, and $F_{n+1} = 2\,\Delta\tilde{C}^{\ell}$.

Substituting Eq. 27 into 28, and $2\,\Delta\tilde{C}^{\ell}$ for F_{n+1}, yields

$$\Delta\Gamma^{\ell} = \frac{-2\phi_f D}{b}\frac{1}{\Delta x_n}\left[\frac{2(1+\beta_n)}{\alpha_{n+1}}\Delta\tilde{C}^{\ell} - G_n\left(\frac{a1_{n+1}(1+\beta_n)}{\alpha_{n+1}} + 1\right)\right] \tag{29}$$

which is equivalent to Eq. 9 with the coefficients

$$\gamma = \frac{-2\phi_f D}{b}\frac{1}{\Delta x_n}\left[\frac{2(1+\beta_n)}{\alpha_{n+1}}\right] \tag{30}$$

$$\delta = \frac{2\phi_f D}{b}\frac{1}{\Delta x_n}\left[G_n\left(\frac{a1_{n+1}(1+\beta_n)}{\alpha_{n+1}} + 1\right)\right] \tag{31}$$

References

Salari, K., Knupp, P., Roache, P., and Steinberg, S., "TVD Applied to Radionuclide Transport in Fractured Porous Media," *Proceedings: IX International Conference on Computational Methods in Water Resources, Denver, Colorado, 9-12 June 1992*, T. Russell, et al., eds., pp. 141-148.

Scheidegger, A. E., *The Physics of Flow Through Porous Media,* University of Toronto Press, 1974.

Appendix II: Sample Diagnostics/Debug File

THE SAMPLE DIAGNOSTICS/DEBUG FILE INCLUDED IN THIS USER'S MANUAL IS NOT THE DIAGNOSTICS/DEBUG FILE USED IN THE 1996 WIPP PA CALCULATION. IT IS INCLUDED *SOLELY FOR ILLUSTRATIVE PURPOSES*.

```
 SSSSSS   EEEEEEE   CCCCC    OOOOO   TTTTTT   PPPPPP     2222   DDDDDD
SS        EE       CC   CC  OO   OO    TT     PP   PP   2    2  DD   DD
SS        EE       CC       OO   OO    TT     PP   PP        2  DD   DD
 SSSSS    EEEEE    CC       OO   OO    TT     PPPPPP       2   DD   DD
     SS   EE       CC       OO   OO    TT     PP          2    DD   DD
     SS   EE       CC   CC  OO   OO    TT     PP         2     DD   DD
SSSSSS    EEEEEEE   CCCCC    OOOOO     TT     PP        222222 DDDDDD

                              SECOTP2D

                         SECOTP2D Version 1.21ZO
                          Version Date 08/16/93
                         Written by Kambiz Salari
                       Sponsored by rebecca blaine

                          Run on 09/22/95 at 14:15:32
                      Run on ALPHA AXP BEATLE OpenVMS V6.1

SECOTP2D 1.21ZO (08/16/93)                                    09/22/95
14:15:32

        ************************************************************

        Prepared for
        Sandia National Laboratories
        Albuquerque, New Mexico 87185-5800
        for the United States Department of Energy
        under Contract DE-AC04-76DP00789

        ************************************************************

                                Disclaimer
                                ----------

        This computer program 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, 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. 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, any agency thereof or any of their
        contractors or subcontractors. The views and opinions
```

```
          expressed  herein  do not necessarily state or reflect those
          of the United States Government, any agency thereof  or  any
          of their contractors or subcontractors.

          ************************************************************


*************************************************************************
**

FILE ASSIGNMENTS:
-----------------


    Number of chains  =  1
    Number of species =  1


    Chain:  1     number of species  1
         Species 1 = U233


    Dual-porosity is used
    Number of cells in the block, IMAX =  9


    Number of cells in X-direction, JMAX =  46
    Number of cells in Y-direction, KMAX =  53


    TIME_BEGIN = 3.1557E+10     TIME_END = 3.1557E+11


    Time step is computed, DT = 5.6802E+08


    Type of input velocity: time dependent


    Maximum number of steps, NUMSTEP = 500


    Algorithm parameters:
        Second order implicit 3-point backward time differencing is used.
        Coefficient of source term:   AX = 0.50    AY = 0.50
        Spatial differencing:
           TVD, van Leer MUSCL limiter, LIMITER = 7


    Property and velocity field data files:
        Property file = SECOTP.PRP

        Velocity file = SECOTP.VEL


    Intermediate results are printed at every   50  time steps.

SECOTP2D CPU time is 1:20 (minute:second)
```

```
*** END OF SECOTP2D ***
SECOTP2D 1.21ZO (08/16/93)                                    09/22/95
14:15:32
```

Appendix III: Review Forms

This appendix contains the review forms for the SECOTP2D User's Manual.

NOTE: Copies of the User's Manual Reviewer's Forms are available in the Sandia WIPP Central Files