Appendix E

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Design Document Criteria**	**Form Number: NP 19-1-4** **Page 1 of 1**

1. **Software Name:** SECOTP2D
2. **Software Version:** 1.44
3. **Document Version:** 1.44
4. **ERMS #:** 573879

Prior to sign-off of the DD, all items shall be appropriately addressed by the code sponsor so that "**Yes**" may be checked. Include this form as part of the DD.

Are the following appropriately defined and documented in the DD?

5. Major Software Components ☒ Yes
6. Technical description of the software with respect to: theoretical basis, embodied mathematical model, major control flow, control logic, and data structures ☒ Yes
7. Functionalities and interfaces of object components, functions, and subroutines. ☒ Yes
8. Allowable or prescribed ranges for inputs and outputs ☒ Yes
9. Verifiability: Is the design verifiable through testing or other means? ☒ Yes
10. Consistency and Traceability: Is the design consistent with and traceable to the software's requirements? ☒ Yes
11. Technical Feasibility: Is the design technically feasible? ☒ Yes
12. Implementation: Is the design presented in sufficient detail to allow for implementation as computer software? ☒ Yes

13.	James Bethune **Code Team/Sponsor** (*print*)	*Signature on File* Signature	12/1/21 Date
14.	Seth King **Technical Reviewer** (*print*)	*Signature on File* Signature	12/1/21 Date
15.	Shelly Nielsen **QA Reviewer** (*print*)	Shelly R. Nielsen Signature	2/2/22 Date
16.	Steve Wagner **Responsible Manager** (*print*)	*Signature on File* Signature	2/16/22 Date
17.	Jennifer Long **SCM Coordinator** (*print*)	*Signature on File* Signature	4/4/22 Date

Information Only

Appendix G

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**User's Manual Criteria**	**Form Number: NP 19-1-6** **Page 1 of 1**

Does the User's Manual contain as appropriate:

1. **Software Name:** SECOTP2D
2. **Software Version:** 1.44
3. **Document Version:** 1.44
4. **ERMS #:** 573879

Prior to sign-off of the User's Manual, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the User's Manual.

5.	A statement(s) of functional requirements (consistent with those in the RD)?	☒ Yes		
6.	An explanation of the mathematical model and numerical models, where applicable as based on code functionality?	☒ Yes	☐ N/A	
7.	Physical and mathematical assumptions, where applicable as based on code functionality?	☒ Yes	☐ N/A	
8.	The capabilities and limitations inherent in the software?	☒ Yes		
9.	Instructions that describe the user's interaction with the software?	☒ Yes		
10.	The identification of input parameters, formats, and valid ranges?	☒ Yes		
11.	Messages initiated as a result of improper input and how the user can respond?	☒ Yes		
12.	The identification and description of output specifications and formats?	☒ Yes		
13.	A description of any required training necessary to use the software?	☒ Yes		
14.	The identification of components of the code that were not tested?	☒ Yes	☐ N/A	

15.	James Bethune **Code Team/Sponsor** (*print*)	*Signature on File* Signature	12/1/2021 Date
16.	Seth King **Technical Reviewer** (*print*)	*Signature on File* Signature	12/1/2021 Date
17.	Shelly Nielsen **QA Reviewer** (*print*)	Shelly R. Nielsen Signature	2/2/22 Date
18.	Steve Wagner **Responsible Manager** (*print*)	*Signature on File* Signature	2/16/22 Date
19.	Jennifer Long **SCM Coordinator** (*print*)	*Signature on File* Signature	4/4/22 Date

Information Only

WIPP PA

Design Document and User's Manual

for

SECOTP2D Version 1.44

Document Version 1.44

ERMS # 573879

April 2022

Information Only

TABLE OF CONTENTS

1.0 INTRODUCTION AND SCOPE 4
1.1 Software Identification 4
1.2 Points of Contact 4
1.3 Code Overview 4

2.0 REQUIREMENTS 5
2.1 Functional Requirements 5
2.2 Performance Requirements 5
2.3 Design Constraints 5
2.4 Attribute Requirements 5
2.5 External Interface Requirement 5
2.6 Other Requirements 6
2.7 Other Tested Functionality 6
2.8 Functionality Not Tested 6

3.0 MAJOR SOFTWARE COMPONENTS 7

4.0 MODELS AND METHODS 9
4.1 Governing Equations 9
4.2 Numerical Discretization 12
4.2.1 Finite Volume Grid 12
4.2.2 Coordinate Transformation 13
4.2.3 Invariants of the Transformation 13
4.2.4 Fracture Equation 13
4.2.5 Flux Limiter Functions and TVD 16
4.2.6 Approximate Factorization 17
4.2.7 Implicit Boundary Conditions 18
4.2.8 Matrix Equation 20
4.2.9 Fracture-Matrix Coupling 21
4.2.10 Solution Accuracy 22

5.0 CAPABILITIES AND LIMITATIONS OF THE SOFTWARE 23

6.0 CONTROL FLOW AND LOGIC 25
6.1 Data Structures 26

Information Only

6.2 Allowable/Prescribed Ranges for Input and Output 26

7.0 REQUIRED USER TRAINING AND/OR BACKGROUND 27

8.0 USER INTERACTIONS WITH THE SOFTWARE 28

8.1 Description of Input Files 29
8.1.1 Input Control File 30
8.1.2 Property Input File 30
8.1.3 Velocity Input File 31
8.2 Description of Output Files 32
8.2.1 Binary Output File 32
8.2.2 Diagnostics Output File 33
8.2.3 Matrix Concentration File 33

9.0 ERROR MESSAGES 35

10.0 REFERENCES 39

11.0 APPENDIX A: DERIVATION OF THE IMPLICIT FRACTURE-MATRIX COUPLING PROCEDURE EMPLOYED BY SECOTP2D 42

12.0 APPENDIX B: SAMPLE INPUT CONTROL FILE 47

13.0 APPENDIX C: SAMPLE DIAGNOSTICS FILE 48

LIST OF FIGURES

Figure 1. Schematic of dual-porosity model. 11

Figure 2. Schematic of finite volume staggered mesh showing internal and ghost cells. Concentrations are defined at cell centers and velocities at cell faces. 13

Figure 3. Computational flow For SECOTP2D and related codes 29

Figure 4. Matrix discretization 43

LIST OF TABLES

Table 1. List of Temporal Discretization Schemes 15

Information Only

1.0 INTRODUCTION AND SCOPE

This Design Document and User's Manual for SECOTP2D Version 1.44, describes the following major features of software design: software components, theoretical basis, mathematical model, control flow and logic, data structures, and functionality and the interfaces of subroutines. It also describes in detail the allowed or prescribed ranges for input data and user interface, in accordance with the Nuclear Waste Management Procedure NP 19-1 Software Requirements [1]. This code is used by the Sandia National Laboratories' Performance Assessment (PA) Department in support of the performance assessment calculations for the Waste Isolation Pilot Plant (WIPP). It is controlled under the software configuration management system CVS (Concurrent Versions System) on the Linux cluster.

1.1 Software Identification

Code Name: SECOTP2D
Version: 1.44
CVS Repository: /data/cvs/CVSLIB/WIPP_CODES/PA_CODES/SECOTP2D

Platform: CentOS 7/Dell PowerEdge C6420
Host: fwm.sandia.gov
Compiler: Gfortran/GCC 8.2.0

1.2 Points of Contact

Code Sponsor: James C. Bethune jbethu@sandia.gov (503) 460-7650

1.3 Code Overview

SECOTP2D performs single or multiple component radionuclide transport in fractured or granular aquifers. Fractured porous media are represented using a dual porosity model. The code uses total variation diminishing (TVD) schemes to model the advection part of the transport equation. This helps to eliminate the guess work in predicting the amount of upwinding to control sharp gradients in the solution as opposed to specifying a number a priori as required by upwind weighted schemes.

Information Only

2.0 REQUIREMENTS

2.1 Functional Requirements

The functional requirements for SECOTP2D, Version 1.44 are listed below.

R.1 - Solve for two-dimensional advective and dispersive transport in a dual-porosity medium with a heterogeneous flow field.

R.2 Apply Dirichlet boundary conditions at specified grid blocks.

R.3 - Introduce radionuclides into the modeling grid at a point source term.

R.4 - Represent the following processes in the transport equation:

a. Fracture-matrix coupling and mass transfer between fractures and matrix
b. Matrix diffusion and retardation.
c. Radioisotope chain ingrowth and decay.

R.5 - Numerically solve the transport equation with the following features:

a. Use van Leer's Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) TVD limiter [2] with Courant number restraints.
b. Use backwards time differencing.
c. Allow for variable time stepping.
d. Allow for variable grid cell spacing.

R.6 - Ouput the following quantities at selected time steps:

a. Mass discharge across user-defined boundaries.
b. Mass balance of each radionuclide.

2.2 Performance Requirements

There are no performance requirements for SECOTP2D.

2.3 Design Constraints

There are no design constraints for SECOTP2D.

2.4 Attribute Requirements

There are no attribute requirements for SECOTP2D.

2.5 External Interface Requirement

The external interface requirements for SECOTP2D are listed below:

R.7 - Read input files generated by PRESECOTP2D to include:

Information Only

- An ASCII input control file
- A binary file containing material property values
- A binary file containing flow velocities and source terms

R.8 This code must link to the libraries CAMCON_LIB and CAMSUPES_LIB [3]

2.6 Other Requirements

There are no other requirements for SECOTP2D Version 1.44.

2.7 Other Tested Functionality

SECOTP2D Version 1.44 has the following functionality that is covered by tests but is not required by the WIPP PA.

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 - Neumann 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

2.8 Functionality Not Tested

The following capabilities of SECOTP2D version 1.44 are not tested:

- Solving the transport equation in single porosity, or discrete fracture medium.
- All of the following options for the numerical solution of the transport equation:
  - Weighted upstream and central differencing, limiter function 2
  - Minmod TVD limiters, limiter functions 3 through 6
  - Roe Superbee TVD limiter, limiter function 8
  - Second order three point backwards time differencing.
  - Second order central time differencing.
- Automatic setting of boundary conditions.
- Flux boundary conditions
- Reporting of maximum concentration.
- Reporting matrix concentration at multiple locations.
- Reporting of discharges with activity rather than mass units.

Information Only

3.0 MAJOR SOFTWARE COMPONENTS

SECOTP2D Version 1.44 is written in FORTRAN. The code is organized into two FORTRAN source files:

- `Almain_secotp2d.f` contains most of the FORTRAN source code. Program secoptd is the main routine. SECOTP2D calls subroutines to read input files and initialize program variables, followed by the primary loop over time steps. Within each iteration of the loop, the source terms, velocity flow field and boundary conditions are updated, the transport equation is solved, mass balance and discharges are computed and output is written. After the timestep loop final mass calculations are performed, output is written, files are closed and the program terminates.
- `Prepo.f` contains a single subroutine `PREPRO` which processes the command line input and sets up input and output files.

Multiple COMMON blocks are used to declare variables with global scope.

SECOTP2D makes use of the WIPP PA libraries CAMCON_LIB and CAMSUPES_LIB [3] to standardize common user interface and I/O functions. These libraries are linked to the SECOTP2D executable during the build process as documented in [4].

The CAMCON_LIB library provides a library of standardized QA and user interface routines, which provide a code banner with credits (authors), software ID, revision date, and current date and time. File processing routines read and parse the command line, and specify and open files; free-field input routines; and string processing routines.

The CAMSUPES_LIB library is a collection of machine dependent routines to access the system clock and allocate memory. However, dynamic memory allocation has not been implemented in SECOTP2D.

Three input and two output files associated with SECOTP2D. All of the input files are created by PRESECOTP2D. SECOTP2D writes one binary output file and one ASCII output file. Content of the binary output file is converted to a CAMDAT database by POSTSECOTP2D Version 1.06 [5]. Input and output files are summarized below.

Input Files:

secotp.inp – ASCII input file created by PRESECOTP2D to run SECOTP2D.

secotp.prp – Binary input file created by PRESECOTP2D containing the material property information.

secotp.vel – Binary input file created by PRESECOTP2D containing the Darcy velocities and source term information.

Output Files:

Information Only

secotp.bin – Binary output file containing data input to the calculation, the resulting time dependent radionuclide discharge across specified surfaces, and time-dependent mass balance at each time step.

secotp.dbg – ASCII diagnostics file summarizing data and settings used in the calculation.

secotp.dbgcon – an optional binary output file containing radionuclide concentrations in the matrix at specified grid locations and time steps.

Information Only

4.0 MODELS AND METHODS

SECOTP2D performs single or multiple component radionuclide transport in fractured porous media using a dual porosity model. The code uses total variation diminishing (TVD) schemes to model the advection part of the transport equation.

4.1 Governing Equations

The equation for the transport of the *k*th radionuclide component in a fracture (*N* species) can be written [6], [7], [8], [9]

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

where $k = 1, \ldots, N$ denotes a radionuclide decay chain in order from parent to daughter species, and each dependent variable C_k (kg/m^3) is the concentration of the *k*th radionuclide. For $k = 1$, the term involving C_{k-1} is omitted. Physical parameters include **D** (m^2/s), a 2 × 2 hydrodynamic dispersion tensor; **V** (m/s), the specific discharge (Darcy velocity); 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/s), the species decay constant; $\tilde{C}_k$ (kg/m^3), the concentration of the *k*th injected radionuclide; Q (1/s), the well specific injection rate defined as the volume of liquid injected per unit volume of porous medium; and Γ_k (kg/m^{3}s), the mass transfer term between the fracture and the matrix. The hydrodynamic dispersion tensor [9] 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 (m), D_k^* *is* the free water molecular diffusion coefficient (m^2/s), τ is the tortuosity (dimensionless), and u and v are the components of the Darcy velocity or specific discharge (m/s), i.e., $\mathbf{V} = \begin{pmatrix} u & v \end{pmatrix}$.

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 **V** is assumed to be independent of the solute concentration. In practice, the flow-field is obtained from the MODFLOW code [10].

Because the dual-continuum model [7], [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

Information Only

that there is no fluid flow (advection) in the matrix, the equation for the concentration of the *k*th species is given by [6]

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

where χ is the coordinate originating from the symmetry line of the matrix block and the prime denotes matrix properties. $D' = D_k^* \tau'$ (m/s) is the coefficient of molecular diffusion where τ' is the matrix tortuosity (dimensionless) and D_k^* (m/s) is the free water molecular diffusion coefficient. 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

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

where b is the fracture aperture, b' is the matrix block length and ϕ is the fracture porosity. The term $\frac{2\phi}{b}$ 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 matix block are given by

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

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

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

Information Only

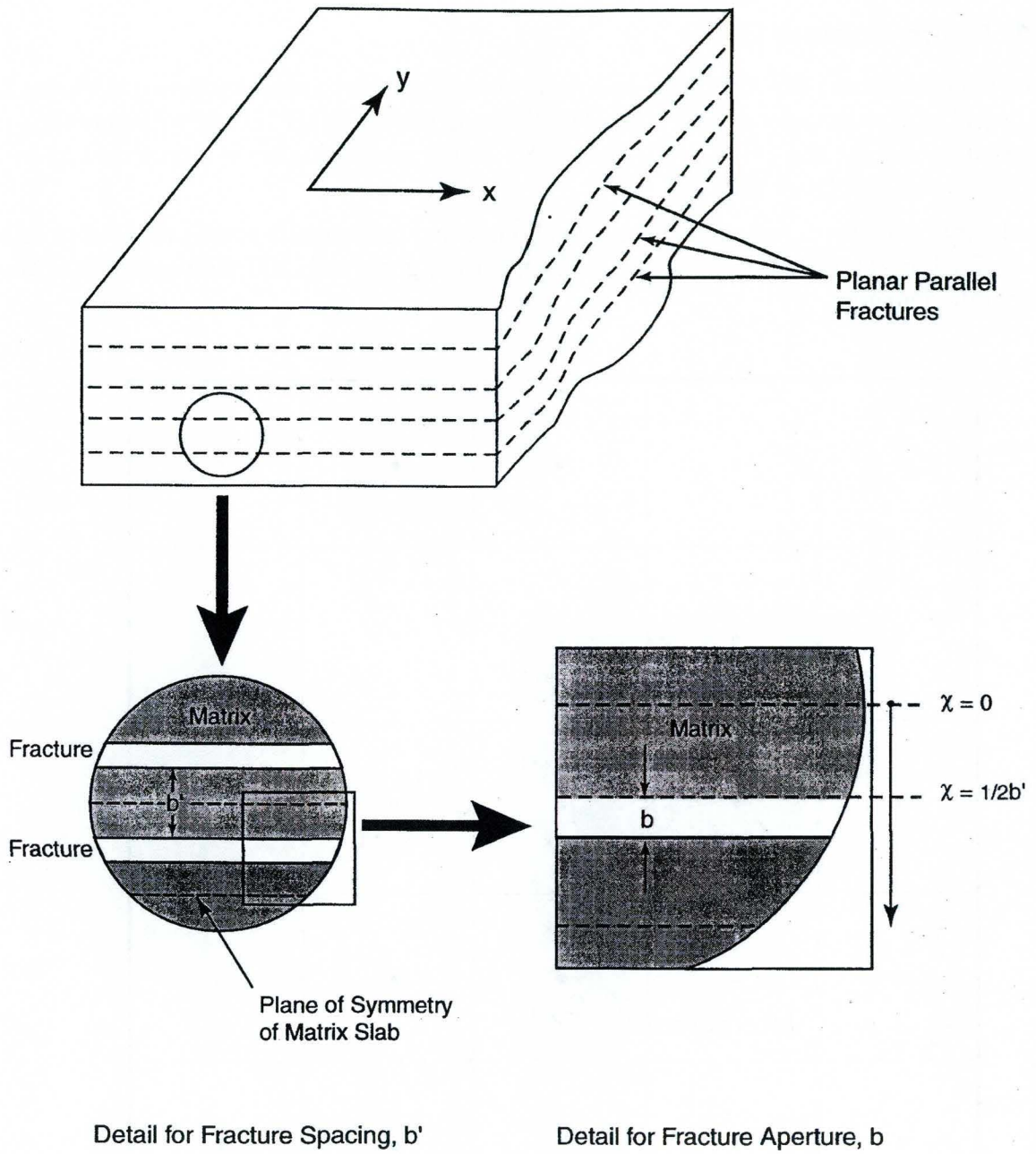


Figure 1. Schematic of dual-porosity model.

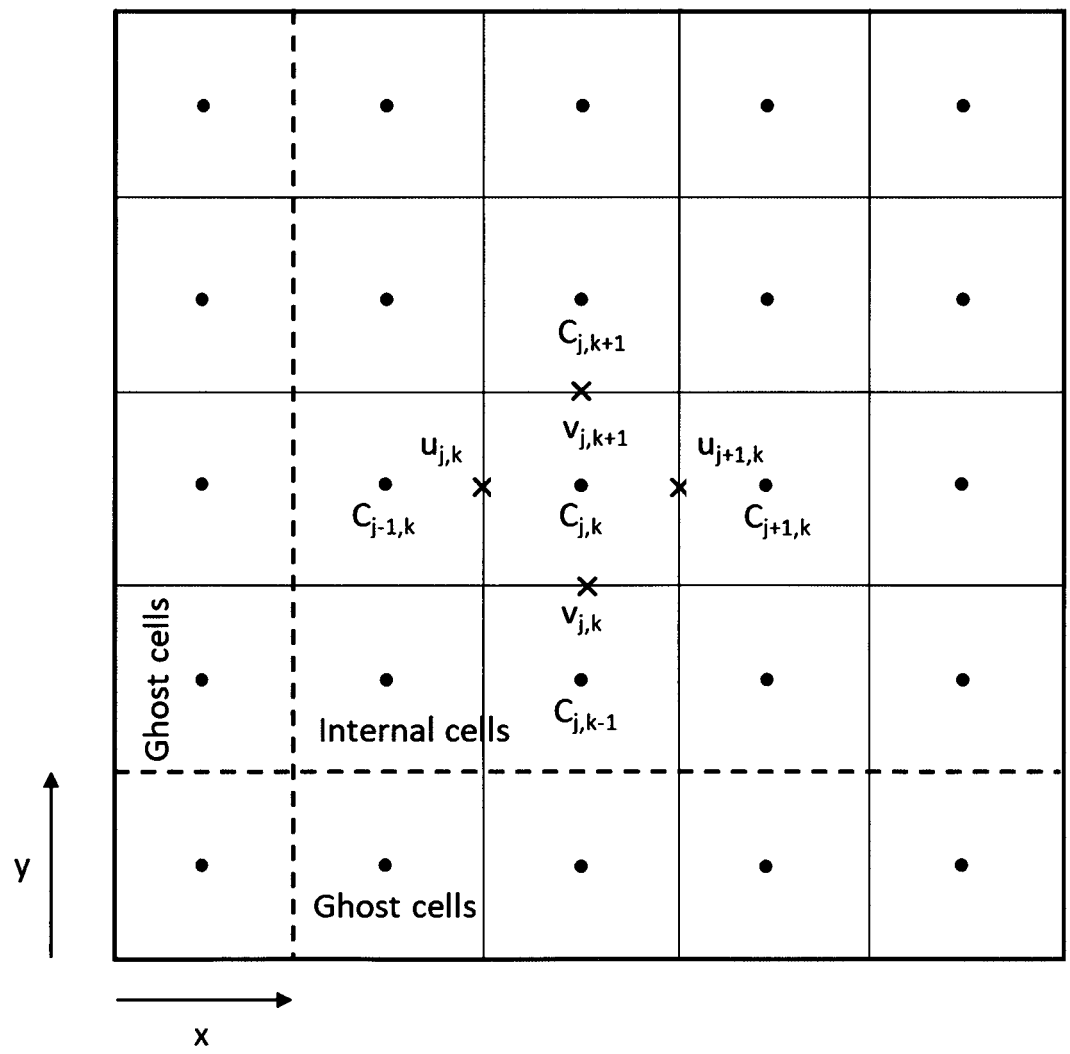
Information Only

4.2 Numerical Discretization

4.2.1 Finite Volume Grid

For the fracture domain SECOTP2D uses a finite volume staggered mesh as shown in Figure 2. An x, y grid with rectangular cells is created by the code PRESECOTP2D and written to the Property input file for SECOTP2D; see Section 8.1.2. The matrix domain is represented by a one-dimensional grid (see Figure 4).

The velocity components (u,v) are defined at cell faces and concentrations are evaluated at cell centers. The boundary conditions are applied by means of ghost cells added on each side of the fracture grid domain.



Information Only

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

4.2.2 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 which 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 written in strong conservation form. For additional information on coordinate transformation, see [14].

4.2.3 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 2-D, standard central differencing satisfies this requirement.

4.2.4 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 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 resulting transport equation in computational space is given by

Information Only

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

where

$$\hat{C}_k=\frac{C_k}{J} \qquad (11)$$

$$\hat{E}=\xi_x u \hat{C}_k \qquad (12)$$

$$\hat{F}=\eta_y v \hat{C}_k \qquad (13)$$

$$\hat{E}_{v1}=\frac{\xi_x^2 \phi D_{11}}{J}\frac{\partial \hat{C}_k}{\partial \xi} \qquad (14)$$

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

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

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

$$\hat{Q}=\frac{Q\tilde{C}_k}{J} \qquad (18)$$

$$\hat{\Gamma}=\frac{\Gamma}{J} \qquad (19)$$

Eq. 10 is solved using an implicit Approximate Factorization procedure [12]. The advective terms are modeled by Total Variation Diminishing (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

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

where

$$\Delta \hat{C}_k^n=\hat{C}_k^{n+1}-\hat{C}_k^n \qquad (21)$$

Information Only

The ΔC_k^n term can be thought of as a correction to advance the solution to a new time level $n+1$. Eq. 20, with appropriate choice of the parameters θ and φ, produce 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	$O(\Delta t)$
½	0	Trapezoidal, implicit	$O(\Delta t^2)$
1	½	3-point-backward, implicit	$O(\Delta t^2)$

Applying Eq. 20 to Eq. 10, we have

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

or

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

Information Only

Note that the delta cross-derivative terms are time-lagged from the n to the $n-1$ time level in both Eq. 22 and Eq. 23. The explicit treatment of the cross-derivative terms is necessary to facilitate the operator splitting/factorization scheme used to solve Eq. 23. A discussion of the operator splitting solver is presented later in this section.

4.2.5 Flux Limiter Functions and TVD

The advective terms $\hat{E}$ and $\hat{F}$ in Eq. 10 are approximated using a TVD flux scheme developed [14] 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 the non-physical oscillations near sharp concentration gradients. As the mesh is refined, the upstream weighting decreases, and the method is formally second-order accuracy.

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

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

In computational space the equivalent expression is

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

where $a=\xi_x u$ and $\sigma=\mathrm{sgn}(a)$.

The function Φ is called a limiter function [13]. SECOTP2D provides nine different limiter functions, six of which are TVD (#4 through #9 below). The TVD limiters range from very compressive (Roe superbee) to very dissipative (minmod) [13]. The limiter functions are defined as:

1. $\Phi(r) = 0$ upstream differencing
2. $\Phi(r) = 1$ central differencing
3. $\Phi(r) = w$ weighted upstream and central differencing
4. $\Phi(r) = minmod\ (1, r)$
5. $\Phi(r) = minmod\ (1, 2 * r)$
6. $\Phi(r) = minmod\ (2, r)$
7. $\Phi(r) = minmod\ (2, 2 * r)$
8. $\Phi(r) = max\ (0, min(2, 2r), (1 + r)/2)$ van Leer's MUSCL

Information Only

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

where r is a function of the concentration gradient and direction of flow. At the $j-1/2\,,k$ interface r is defined in terms of its partial derivatives as

$$r_x = \begin{cases} \dfrac{C_x\big|_{j-3/2,k}}{C_x\big|_{j-1/2,k}} = \dfrac{C_\xi \xi_x\big|_{j-3/2,k}}{C_\xi \xi_x\big|_{j-1/2,k}}, & u > \varepsilon \\ 0 & |u| \le \varepsilon \\ \dfrac{C_x\big|_{j+1/2,k}}{C_x\big|_{j-1/2,k}} = \dfrac{C_\xi \xi_x\big|_{j+1/2,k}}{C_\xi \xi_x\big|_{j-1/2,k}}, & u < -\varepsilon \end{cases}$$

$$r_y = \begin{cases} \dfrac{C_y\big|_{j,k-3/2}}{C_y\big|_{j,k-1/2}} = \dfrac{C_\eta \eta_y\big|_{j,k-3/2}}{C_\eta \eta_y\big|_{j,k-1/2}}, & v > \varepsilon \\ 0 & |v| \le \varepsilon \\ \dfrac{C_y\big|_{j,k+1/2}}{C_y\big|_{j,k-1/2}} = \dfrac{C_\eta \xi_y\big|_{j,k+1/2}}{C_\eta \eta_y\big|_{j,k-1/2}}, & v < -\varepsilon \end{cases} \tag{26}$$

where here $\varepsilon = 1\mathrm{E}-37$. The *minmod* function is given by

$$\min \bmod (a,b) = \mathrm{sgn}\,(a) \max\left\{0, \min\left\{|a|, \mathrm{sgn}\,(a) \times b\right\}\right\} \tag{27}$$

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

$$Cr_x = \frac{u\Delta t}{\phi_f \Delta x}$$
$$Cr_y = \frac{v\Delta t}{\phi_f \Delta y} \tag{28}$$

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

The SECOTP2D default limiter is van Leer's MUSCL [2] (limiter function 7).

4.2.6 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

Information Only

$$\left(\mathbf{I}+\alpha_{\xi\xi}\mathbf{L}_{\xi\xi}+\alpha_{\eta}\mathbf{L}_{\eta\eta}+\mathbf{S}\right)\Delta\hat{C}=RHS \tag{29}$$

where **I** is an identity matrix, $\mathbf{L}_{\xi\xi}$ and $\mathbf{L}_{\eta\eta}$ are the ξ and η operators, and **S** is an implicit source term that accounts for decay and matrix/fracture mass transfer. To solve Eq. 29 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 amount 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 is 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 $n+1$ is obtained through the following sequence

$$\left(\mathbf{I}+\alpha_{\xi\xi}\mathbf{L}_{\xi\xi}+a_{\xi}\mathbf{S}\right)\Delta\bar{C}=RHS \tag{30}$$

$$\left(\mathbf{I}+\alpha_{\eta}\mathbf{L}_{\eta\eta}+a_{\eta}\mathbf{S}\right)\Delta\hat{C}=\Delta\bar{C} \tag{31}$$

$$\hat{C}_{j,k}^{n+1}=\hat{C}_{j,k}^{n}+\Delta\hat{C}_{j,k}^{n} \tag{32}$$

where a_{ξ} and a_{η} are constants that establish the fraction of the implicit source allocated in each of the one-dimensional sweeps. The sum of these two constants must be one. Ramsey and Treadway [15] showed that the choice of the source term splitting constants can dramatically influence the accuracy of the simulation and 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.

Eq. 30 describes the initial sweep in the x or logical ξ direction. At the end of this sweep the intermediate solution $\Delta\bar{C}$ 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. 31. The result is substituted into Eq. 32 to update the concentration to the new time level.

4.2.7 Implicit Boundary Conditions

Boundary conditions are implemented trhough ghost or imaginary grid blocks positions one cell outside the model domain (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 compte the intermediate solution. The following is a derivation of the intermediate boundary condition for the left boundary of 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

Information Only

$$\Delta C_{0,k} = b_k \Delta C_{1,k} + d_k \tag{33}$$

where the subscript k refers to the row number, $\Delta C_{0,k}$ 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. 31 the intermediate solution at ghost cells on the left boundary is described by

$$\Delta \bar{C}_{0,k} = \left(\mathbf{I} + \alpha_\eta \mathbf{L}_{\eta\eta} + a_\eta \mathbf{S}\right)_{0,k} \begin{pmatrix} \Delta \hat{C}_{0,k-1} \\ \Delta \hat{C}_{0,k} \\ \Delta \hat{C}_{0,k+1} \end{pmatrix} \tag{34}$$

Substituting Eq. 33 into Eq. 34 yields

$$\Delta \bar{C}_{0,k} = \left(\mathbf{I} + \alpha_\eta \mathbf{L}_{\eta\eta} + a_\eta \mathbf{S}\right)_{0,k} \begin{pmatrix} b_{k-1}\Delta \hat{C}_{1,k-1} + d_{k-1} \\ b_k \Delta \hat{C}_{1,k} + d_k \\ b_{k+1}\Delta \hat{C}_{1,k+1} + d_{k+1} \end{pmatrix} \tag{35}$$

The coefficients b_{k-1}, b_k and b_{k+1} are equivalent when the boundary condition is uniformly Dirichelt or Neumann. Under such conditions, the constant b can be factored out of Eq. 35 resulting in

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

Also, because the ghost cell grid spacing is equivalent to the boundary cell grid spacing, the linear operator in Eq. 36 evalued at $0, k$ is equivalent to the linear operator evaluated at $1, k$

$$\left(\mathbf{I} + \alpha_\eta \mathbf{L}_{\eta\eta} + a_\eta \mathbf{S}\right)_{0,k} = \left(\mathbf{I} + \alpha_\eta \mathbf{L}_{\eta\eta} + a_\eta \mathbf{S}\right)_{1,k} \tag{37}$$

The equation for the intermediate boundary condition is obtained by substituting Eq. 37 and Eq. 34 into Eq. 36, resulting in

$$\Delta \bar{C}_{0,k} = b\Delta \bar{C}_{1,k} + \left(\mathbf{I} + \alpha_\eta \mathbf{L}_{\eta\eta} + a_\eta \mathbf{S}\right)_{1,k} \begin{pmatrix} d_{k-1} \\ d_k \\ d_{k+1} \end{pmatrix} \tag{38}$$

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. However, with SECOTP2D v1.44 this option should not be used for WIPP PA calculations because it may result in non-uniform

Information Only

boundary conditions along a side of the model domain. 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 fow 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 zero concentration.

4.2.8 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 approximated using the same three-level implicit finite volume scheme discussed in the preceding section. The governing equation for matrix diffusion (Eq. 3) is restated here in a slightly modified form for convenience:

$$\phi' R'_k \left(\frac{\partial C'_k}{\partial t} \right) = \frac{\partial}{\partial \chi} \left(\phi' D' \frac{\partial C'_k}{\partial \chi} \right) - \phi' R'_k \lambda_k C'_k + \phi' R'_{k-1} \lambda_{k-1} C'_{k-1} \tag{39}$$

Applying the generalized three-level scheme ([9], pg. 229) to Eq. 39 yields

$$\phi' R'_{k,i} \frac{(1+\varphi)\Delta C'^{n}_{k,i} - \varphi \Delta C'^{n-1}_{k,i}}{\Delta t} = \theta \left(\mathbf{L}_{\chi\chi} \Delta C'^{n}_{k,i} + \Delta Q'^{n}_{k,i} \right) + \left(\mathbf{L}_{\chi\chi} C'^{n}_{k,i} + Q'^{n}_{k,i} \right) \tag{40}$$

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

$$\mathbf{L}_{\chi\chi} C'^{n}_{k,i} = \frac{2}{\Delta \chi_i} \left[\left(\phi' D' \right)_{k,i+1/2} \frac{\left(C'^{n}_{k,i+1} - C'^{n}_{k,i} \right)}{\left(\Delta \chi_{i+1} + \Delta \chi_i \right)} - \left(\phi' D' \right)_{k,i-1/2} \frac{\left(C'^{n}_{k,i} - C'^{n}_{k,i-1} \right)}{\left(\Delta \chi_i + \Delta \chi_{i-1} \right)} \right] \tag{41}$$

$$\Delta C'^{n}_{k,i} = C'^{n+1}_{k,i} - C'^{n}_{k,i} \tag{42}$$

$$\mathbf{L}_{\chi\chi} \Delta C'^{n}_{k,i} = \mathbf{L}_{\chi\chi} C'^{n+1}_{k,i} - \mathbf{L}_{\chi\chi} C'^{n}_{k,i} \tag{43}$$

$$Q'^{n}_{k,i} = \left(-\phi' R'_k \lambda_k C'^{n}_{k} + \phi' R'_{k-1} \lambda_{k-1} C'^{n}_{k-1} \right)_i \tag{44}$$

$$\Delta Q'^{n}_{k,i} = Q'^{n+1}_{k,i} - Q'^{n}_{k,i} \tag{45}$$

Arranging Eq. 40 such that terms containing $C'^{n+1}_{k,i}$ are grouped on the left hand side (LHS) and the remaining terms are grouped on the RHS results in

Information Only

$$\begin{aligned}\phi' R'_{k,i}\Delta C'^{n}_{k,i} &- \frac{\theta\Delta t}{1+\varphi}\left(\mathbf{L}_{\chi\chi}\Delta C'^{n}_{k,i} - \phi' R'_{k,i}\lambda_k \Delta C'^{n}_{k,i}\right) \\ &= \frac{\Delta t}{1+\varphi}\left(\mathbf{L}_{\chi\chi}\Delta C'^{n}_{k,i} + Q'^{n}_{k,i}\right) \\ &+ \frac{\theta\Delta t}{1+\varphi}\left(\phi' R'_{k-1,i}\lambda_{k-1}\Delta C'^{n}_{k-1,i}\right) \\ &+ \frac{\varphi}{1+\varphi}\phi' R'_{k,i}\Delta C'^{n-1}_{k,i}\end{aligned} \tag{46}$$

Applying Eq. 46 yields a tridiagonal system of equations that can be solved efficiently using the Thomas algorithm ([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 (Table 1).

4.2.9 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 levels (explicit) or the n+1 time level (implicit). Huyakorn et al. [7] investigated an explicit formulation and found that it often resulted in large mass conservation errors. They 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 et al. [7]. The coupling procedure consists of three steps. Step 1 involves writing the incremental mass transfer term $\Delta\hat{\Gamma}^n_k$ in a form that can be inserted into the implicit part of the fracture equation

$$\Delta\hat{\Gamma}^n_k = a\Delta C^n_k + b$$

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. Matrix concentrations are then obtained using the upper tridiagonal matrix U through back substitution.

Information Only

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

4.2.10 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 MODFLOW, 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.

Information Only

5.0 CAPABILITIES AND LIMITATIONS OF THE SOFTWARE

SECOTP2D performs single or multiple component radionuclide transport in two-dimensional fractured porous media using a dual porosity model. The code uses total variation diminishing (TVD) schemes to model the advection part of the transport equation.

SECOTP2D does not calculate flow in the porous medium. SECOTP2D requires, as input, a file containing specific discharge (Darcy velocity) as described in Section 9.1. SECOTP2D requires, as input, a set of source points (grid locations) and at each source point, a time series of radionuclide masses entering the transport domain.

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.
- The matrix block equation can model both a matrix material and a clay lining.
- 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) or central differencing. TVD helps eliminate the guess work in predicting the amount of upwinding to control sharp gradients in the solution as opposed to specifying a number of a priori as required by upwind weighted schemes.
- 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 (see Section 4.2.7).
- Implicit coupling of the equations for the fracture and matrix block is used. This approach is more robust than explicit coupling (see Section 4.2.9).
- 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 (see Section 4.2.10).
- Lack of smoothness of the flow field near boundaries can pose difficulties for the transport calculation. The SECOTP2D code, in most cases, can eliminate these difficulties by automatically assigning the boundary conditions using the flow field (see Section 4.2.10).

Information Only

- SECOTP2D provides discharge calculations on different pre-defined closed boundaries (see Section 8.1.2 and Section 8.2.1). Boundaries must be placed interior to the model domain, otherwise, mass flux across the boundary may be inaccurate.

Information Only

6.0 CONTROL FLOW AND LOGIC

In this section you will have a description of the major control flow, control logic and data structures. This should include allowable/prescribed ranges for input and output.

The main program for SECOTP2D is found in file *A1_main_secotp2d.f*. The computational flow proceeds as follows:

1. Subroutine *prepro* processes command-line arguments using functions from the CAMCON_LIB and CAMSUPES_LIB libraries, opens input and output files, and writes QA-related information to output files.
2. Subroutine *input* initializes internal variables, reads the input control file and sets default values where the control input file does not specify a value. Input quantities related to grid dimensions, radionuclide species and chains, numerical settings and timesteps are checked against the limits described in Section 6.2 violation aborts the run.
3. Subroutine *init* assigns logical file numbers, reads the input property file, sets up the computational grid, assigns initial and boundary conditions and sets up the variables that accumulate discharge across surfaces. Subroutine *init* also reads the input velocity file and sets up the internal velocity field, and reads the source term information. Finally, subroutine *init* sets up to print matrix concentrations at the requested cells. If cell indices are outside of the model domain, the run is aborted.
4. Subroutine *init_matrix* sets initial mass in the fracture and matrix grid, computes matrix-related quantities and coefficients for the LU factorization (see Section 4.2.9).
5. The code then iterates over the time steps. At each time step:
    a. Subroutine *source* introduces new radionuclide mass into the transport domain from the source terms.
    b. Subroutine *step* solves the governing equations (Section 4.1) using the numerical scheme (Section 4.2) to update the mass in each grid cell to the next time step.
    c. Subroutine *discharge* computes mass flux across the specified surfaces and writes to output files.
    d. Subroutine *mass_bal* calculates the mass balance for each radionuclide. If the maximum mass balance error (over radionuclides) exceeds the limit (1.05 kg) the run is aborted and an error is written to output.
6. At the end of the time step loop, subroutine *output* is called three times to write:
    a. a summary of the run control information to the ASCII diagnostics output file (Section 8.2.2),
    b. flow velocity, radionuclide concentrations in the fracture cells, and integrated radionuclide discharge across each surface to the binary output file (Section 8.2.1),
    c. optionally, radionuclide concentrations in selected matrix cells, to the ASCII matrix concentration output file (Section 8.2.3).
7. A summary of radionuclide mass balance errors is written to the screen output and to the debug file.

If the run completes normally, the message `'SECOTP2D Normal Completion'` is written to the screen output.

Information Only

6.1 Data Structures

SECOTP2D v1.44 uses a rectangular 2D grid for the fracture portion of the domain (Figure 1). This grid is indexed by 'j' and 'k' (index 'i' is used for the matrix portion of the domain. Face-centered coordinates for fracture grid cells are stored in variables `xf` and `yf` which are dimensioned as [-1 : MAXJ+2] and [-1 : MAXK+2]; the index values extend the 2D grid by adding ghost cells to account for boundary conditions. All quantities in the fracture grid (e.g., radionuclide concentration) are indexed using the fracture grid coordinates.

The matrix portion of the domain is a 1D slab model (Figure 1) indexed by 'i'. Face-centerd coordinates for the matrix grid are sotred in variable `sx` which is indexed [-1 : MAXI+2] which extends the matrix grid with ghost cells. All matrix quantities are indexed using the matrix grid coordiantes.

6.2 Allowable/Prescribed Ranges for Input and Output

SECTOP2D sets the following maximums:

- Number of decay chains: 6
- Number of radionuclide species: 6
- Maximum dimensions of the grid representing the fractured part of the domain: 200 cells (j-direction) by 110 cells (k-direction)
- Maximum dimension of the grid representing the matrix part of the domain: 31 cells (i-direction)
- Maximum number of source points for a radionuclide: 100
- Maximum number of points in a source term time series: 201
- Maximum number of time steps: 20000
- Maximum number of surfaces for output of radionuclide discharge: 5
- Maximum number of points for output of matrix concentration: 155

Numerical control parameters are checked against the following criteria:

- $0.5 \le \theta \le 1$ and $0 \le \varphi \le 0.5$ (see Table 1. Note that in the code, φ is stored in the variable *omega*).
- $0 \le w \le 1$ applies to limiter option 3; see Section 4.2.5.
- TIME_BEGIN is less than TIME_END; see [16], Section 8.2.2.

Information Only

7.0 REQUIRED USER TRAINING AND/OR BACKGROUND

In order to run the SECOTP2D code successfully, the user will need a basic knowledge of the FORTRAN and the Linux operating system. In addition, users should also have access to the WIPP cluster or an equivalent.

In order for the theoretical section of this manual to be useful, the user will need the following:

- An understanding of partial differential equations and linear algebra
- An understanding of numerical methods applied to partial differential equations.
- An understanding of groundwater flow and transport.

To apply SECOTP2D effectively, the user should be aware of the code's capabilities and limitations.

Information Only

8.0 USER INTERACTIONS WITH THE SOFTWARE

SECOTP2D Version 1.44 executes on a Linux platform. To execute SECOTP2D, the program executable name (secotp2d) must be in the user's current PATH or referenced by its path name.

SECOTP2D expects a number of command line arguments. The command line arguments may either be positional (provided in order) or flagged (preceded by the appropriate flag). Flags always start with "–" and may be abbreviated. All positional arguments must be listed before any flagged arguments. The only difference between using positional and flagged arguments is that optional arguments may be omitted when using flagged arguments.

Interactive users may request that the program prompt for any missing command line arguments by appending "?" to the command line.

SECOTP2D expects the following command line arguments:

1. –user The input control file. This file specifies the processing options for SECOTP2D. Unlike the input control files for most WIPP PA codes, the user has no interactive control over the contents of this file. This file is generated by PRESECOTP2D. The input file to PRESECOTP2D determines the contents of this file.
2. –prop The binary input property file. This file is generated by PRESECOTP2D.
3. –vel The binary input velocity file, containing velocities and source information. This file is generated by PRESECOTP2D.
4. –output The binary output transfer file. This file is input to POSTSECOTP2D.
5. –debug The output diagnostics file. This file name is also used (with extension '.dbgcon') for the optional matrix concentration output file.

The three input files to SECOTP2D are created with a single PRESECOTP2D execution. Thus, these three files serve as a single set of transfer files from PRESECOTP2D to SECOTP2D. SECOTP2D 1.44 reads the names of all three files from the command line. In previous versions of SECOTP2D, the names of the property and velocity files were read from the input control file. The input control file still contains the names of the binary property and velocity files, but these names are ignored by SECOTP2D 1.44.

Information Only

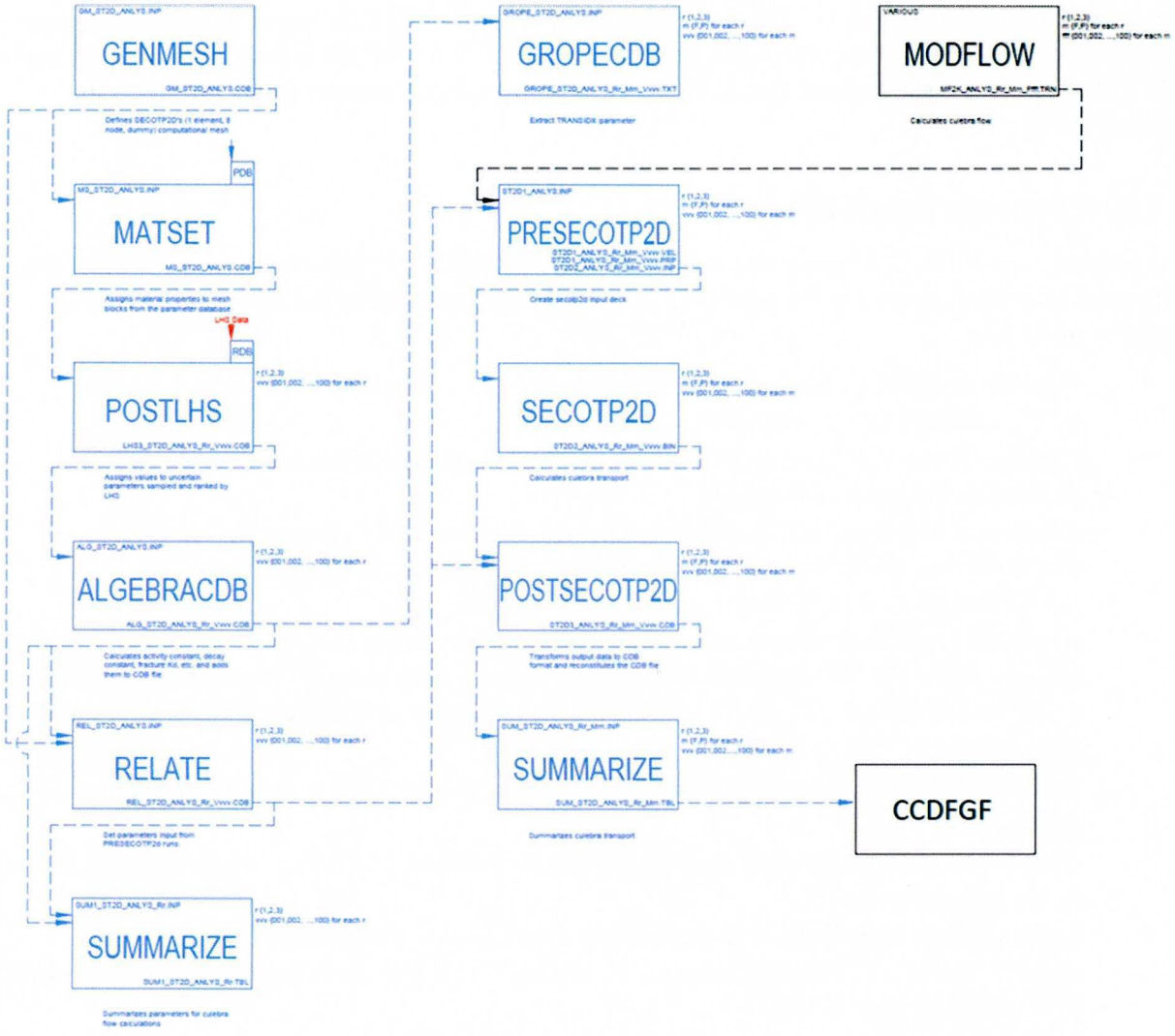


Figure 3. Computational flow For SECOTP2D and related codes

8.1 Description of Input Files

SECOTP2D requires three input files:

- An ASCII input control file containing parameters including numerical settings and output selections.
- A binary file containing grid coordinates, material properties and radionuclide properties
- A binary file containing Darcy velocities and radionuclide source terms.

All three files can be produced by PRESECOTP2D Version 1.24 [16].

Information Only

8.1.1 Input Control File

The input control file to SECOTP2D is generated by PRESECOTP2D based in part on its own input control file. The input control file generated by PRESECOTP2D is fully documented in the PRESECOTP2D's User Manual ([16], Table 1). A sample input control file is provided in Appendix B.

8.1.2 Property Input File

The property input file is a binary file output by PRESECOTP2D. This binary file contains the following in order (names are the SECOTP2D variables used to store the data read from the property input file:

Coordinates for the fracture grid, in order: `xf[1:JMAX+1, 1:KMAX+1]` and `yf[1:JMAX+1, 1:KMAX+1]`
(Optional) variable time steps for SECOTP2D, expected if keyword VAR_DT is present in the input control file.
Fracture grid properties, in order: tortuosity, longitudinal dispersivity, transverse dispersitivity, effective porosity and half fracture aperture. Each is indexed `[0:JMAX+1, 0:KMAX+1]`.
(Optional) matrix grid properties if diffusion in the matrix is considered (when keyword DUAL_POROSITY is present in the input control file): matrix block length and skin resistance, indexed `[1:JMAX, 1:KMAX]`; tortuosity, effective porosity, each indexed `[0:JMAX+1, 0:KMAX+1]`.
Radionuclide species names, indexed `[1:NSP]` where NSP is found in the input control file.

- Number of radionuclides in each decay chain, indexed `[1:NCH]` where NCH is found in the input control file.
  Index of each radionuclide in each decay chain (index determined by ordering of species names and chains), indexed `[1:NCH, 1:NSP]` with the indexes being looped in order NCH then NSP.
  Radionuclide decay constants indexed `[1:NSP]`
  Free water molecular diffusion coefficients indexed `[1:NSP]`
  (Optional) radionuclide discharge surface definitions (when keyword COMPUTE_DISCHARGE is present in the input control file): number of surfaces, j,k index of upper left corner of discharge surface, j,k index of lower right corner of discharge surface, radionuclide specific activity indexed `[1:NSP]`
  For each radionuclide:
  - fracture retardation coefficient for each grid cell, indexed `[0:JMAX+1, 0:KMAX+1]`
  - (Optional) matrix retardation coefficient for each matrix grid cell (when keyword DUAL_POROSITY is present in the input control file), indexed `[0:IMAX+1, 1:JMAX, 1:KMAX]`
  - Boundary condition, ordered by four indexes: a boundary cell index generated by PRESECOTP2D, side (1 – north, 2 – south, 3 – east, 4 – west), boundary

Information Only

condition type (1 – Dirichlet, 2 – Neumann, 3 – flux, 4 – auto) and radionuclide species (from loop to read the data).
- (Optional) initial concentration in each fracture grid cell (when keyword INIT is present in the input control file), indexed `[1:JMAX, 1:KMAX]`
- Number of points at which matrix concentrations are written to output, `matrix_pts_pr`.
- (Optional) if matrix concentrations are to be written, fracture grid coordinates j,k at which matrix concentrations are written, indexed `[1: matrix_pts_pr, 1:NSP]`

8.1.3 Velocity Input File

The velocity data input file contains double precision flow velocities and source information in binary format. The velocity file is written by PRESECOTP2D [16]. The content differs between steady-state and transient flow conditions, as specified by the input control file keyword STEADY_UV

8.1.3.1 Steady-state velocity input file

A steady-state velocity input file contains in order:

Darcy velocity in the x-direction (j index) of the fracture domain, indexed `[1:JMAX+1, 0:KMAX+1]`
Darcy velocity in the y-direction (k index) of the fracture domain, indexed `[0:JMAX+1, 1:KMAX+1]`
(Optional) radionuclide point source data, when the keyword POINT_SOURCE is present in the input control file. As currently configured, PRESECOTP2D v1.24 always writes the keyword POINT_SOURCE = True to the control input file ([16], Table 1). Point source data include:
- Number of point sources for each radionuclide, indexed [1:NSP]
- For each radionuclide:
  - For each point source, j,k location of point source in fracture grid
- Number of points in the time series for every point source.
- For each radionuclide:
  - For each point source, time values for time series of radionuclide mass entering the fracture domain.
  - For each point source, radionuclide mass values for each time point.

8.1.3.2 Transient velocity input file

A transient velocity input file contains in order:

(Optional) radionuclide point source data, when the keyword POINT_SOURCE is present in the input control file. As currently configured, PRESECOTP2D Version 1.24 always writes the keyword POINT_SOURCE = True to the control input file ([16], Table 1). Point source data include:
- Number of point sources for each radionuclide, indexed `[1:NSP]`

Information Only

- For each radionuclide:
  - For each point source, j,k location of point source in fracture grid
- Number of points in the time series for every point source.
- For each radionuclide:
  - For each point source, time values for time series of radionuclide mass entering the fracture domain.
  - For each point source, radionuclide mass values for each time point.

For each time step:

- Time value of the time step
- Darcy velocity in the x-direction (j index) of the fracture domain, indexed `[1:JMAX+1, 0:KMAX+1]`
- Darcy velocity in the y-direction (k index) of the fracture domain, indexed `[0:JMAX+1, 1:KMAX+1]`

8.2 Description of Output Files

SECOTP2D produces three output files:

1. A binary file containing radionuclide discharge, concentrations and mass balance information, along with other data as described in Section 8.2.1.
2. An ASCII output file containing information describing the SECOTP2D calculation.
3. (Optional) An ASCII output file containing matrix concentrations, when keyword PRINT_MATRIX_CON is present in the input control file.

8.2.1 Binary Output File

The binary output file contains the time series of radionuclide discharge across the specified surfaces.

The binary output file first contains a block of data describing the SECOTP2D calculation and the radionuclides, including in order:

- Initial time
- Number of radionuclides, number of timesteps to print, number of discharge surfaces, number of time steps, and number of time steps between writing discharge data, maximum fracture grid dimension in the j direction, maximum fracture grid dimension in the k direction
- Text symbols for each radionuclide
- Initial Darcy velocity in the x-direction (j index) of the fracture domain, indexed [1:JMAX, 1:KMAX]
- Initial Darcy velocity in the y-direction (k index) of the fracture domain, indexed [1:JMAX, 1:KMAX]
- For each radionuclide, initial concentration in the fracture domain, indexed [1:JMAX, 1:KMAX]
- For each radionuclide, total mass in the fracture domain per unit thickness of the fracture domain.

Information Only

- For each radionuclide, total mass in the matrix domain per unit thickness of the fracture domain.

Next, the binary output file contains a block of data for each time step, in order:

- Time and an internal signal flag `iflag` (=0, 1, 2, or 3) which is used by the post-processing code to distinguish during-run timesteps from the end-of-run timestep.
- Discharge information for each radionuclide and discharge surface, to include: radionuclide index, discharge surface index, dispersive flux, advective flux, time-integrated discharge mass, , time-integrated discharge activity, maximum flux across any cell boundary on the surface, j index of cell with maximum flux, k index of cell with maximum flux
- Mass balance information for each radionuclide to include:
  - Mass in fracture domain
  - Mass in matrix domain
  - Mass change due to decay and ingrowth
  - Mass discharged from model domain
  - Net mass entering domain from sources and sinks
  - Mass balance error
- At the last time step or (optional) at timesteps selected for writing matrix concentrations (when keywords OUTPUT – STEP are present in the control input file, see [16], Section 8.1.4):
  - Darcy velocity in the x-direction (j index) of the fracture domain, indexed `[1:JMAX, 1:KMAX]`
  - Darcy velocity in the y-direction (k index) of the fracture domain, indexed `[1:JMAX, 1:KMAX]`
  - For each radionuclide, concentration in the fracture domain, indexed `[1:JMAX, 1:KMAX]`
- For each radionuclide and each discharge surface, time-integrated discharge.

8.2.2 Diagnostics Output File

SECOTP2D produces an ASCII file containing diagnostic information. A sample file is provided in Appendix C. This file contains:

- Radionuclide and decay chain information
- Fracture and matrix grid dimensions
- Time step information
- Numerical control parameters
- Names of input files
- Output options

8.2.3 Matrix Concentration File

Optionally, SECOTP2D creates an ASCII output file containing matrix concentrations. This file is created when keyword PRINT_MATRIX is present in the *OUTPUT block of the

Information Only

PRESECOTP2D control file. The file name is the same as the Diagnostics Output File but the file extension is '.dbgcon' rather than '.dbg'.

For time steps selected by parameters *OUTPUT – STEP, and fracture grid locations selected by parameters *OUTPUT – PRINT_MATRIX (parameters are in the input control file), this file contains, for each radionuclide, a table of concentration at each matrix cell center.

Information Only

9.0 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 maximum number of iterations has been exceeded.

```
WRITE(6,*) 'SECO2D-TRANSPORT:'
WRITE(6,*) 'ERROR, Subroutine FRACTURE,  maximum number of '
WRITE(6,*) 'iteration exceeded, MAXITER =',MAXITER
WRITE(6,*) 'It is possible intra-time step iteration is'
WRITE(6,*) 'diverging.'
```

To recover from this error, rerun SECOTP2D with a greater number of time steps.

* 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 a location where matrix concentration is requested is not in the computational domain.

```
              write(6,*) 'secotp2d: error, subroutine init.'
              write(6,*) 'location requested to print matrix '
>                        'concentration is out of computational',
>                        'domain,  j,k =',j,k
```

* These write statements generate an error message indicating that the value of the source function QC cannot be evaluated.

```
WRITE(6,*) 'SECO2D-TRANSPORT:
     >              'Error, subroutine QC_INTEGRATE.'
          WRITE(6,*) 'value of source function QC cannot be',
     >              ' evaluated for a specified time.'
          WRITE(6,*) 'TIME = ',TB
```

This error occurs when the time domain of the source function is not aligned with the time domain of the simulation. The error is corrected by making sure those time domains are aligned and then rerunning SECOTP2D.

* These write statements generate an error message when the location of a source point is not in the computational domain.

Information Only

```
      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 when an end-of-file has been detected where SECOTP2D expects to read data.

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

* This write statement generates an error message if the number of variable time steps exceeds the number of time steps read from the input control file.

```
      write(6,490) numstep, istep
490   format(5x,'numstep = ',i10,' is only dimensioned for ',i10)
```

* This write statement generates an error message when the number of cells in the matrix domain exceeds the allowable range.

```
      write(6,500) imax, idim
500   format(5x,'imax = ',i3,' is only dimensioned for ',i3)
```

* This write statement generates an error message when the j (x) dimension of the fracture domain exceeds the allowable range.

```
      write(6,510) jmax, jdim
510   format(5x,'jmax = ',i3,' is only dimensioned for ',i3)
```

* This write statement generates an error message when the k (y) dimension of the fracture domain exceeds the allowable range.

```
      write(6,520) kmax, kdim
520   format(5x,'kmax = ',i3,' is only dimensioned for ',i3)
```

* This write statement generates an error message when the number of radionuclide species exceeds the allowable range.

```
      write(6,530) nsp, isp
530   format(5x,'nsp = ',i3,' is only dimensioned for ',i3)
```

* This write statement generates an error message when the number of radionuclide species is less than 1.

```
      write(6,580) nsp
580   format(5x,'nsp = ',i3,' is out of bound' )
```

Information Only

* This write statement generates an error message when the number of radionuclide decay chains exceeds the allowable range.

```
      write(6,540) nch, ich
540   format(5x,'nch = ',i3,' is only dimensioned for ',i3)
```

* This write statement generates an error message when the number of radionuclide decay chains is less than 1.

```
      write(6,590) nch
590   format(5x,'nch = ',i3,' is out of bound' )
```

* This write statement generates an error message when the temporal discretization parameter θ exceeds limits (see Table 1). This parameters are specified by the input control file parameter *CONTROL – TIME_SCHEME ([18] Section 8.1.4).

```
      write(6,550) theta
550   format(5x,'value of theta is out of bound, theta =',f7.3)
```

* This write statement generates an error message when the temporal discretization parameter φ exceeds limits (see Table 1). This parameters are specified by the input control file parameter *CONTROL – TIME_SCHEME ([18] Section 8.1.4).

```
      write(6,560) omega
560   format(5x,'value of omega is out of bound, omega =',f7.3)
```

* This write statement generates an error message when the approximate factorization parameters a_ξ and/or a_χ violate limits; see Section 4.2.6. These parameters are specified in the input control file using parameters *CONTROL – SOURCE_COEFF ([18], Section 8.1.4).

```
      write(6,570) ax,ay
570   format(5x,'value of ax,ay are out of bound, ax,ay =',2f8.3)
```

* This write statement generates an error message when the dimension of the matrix domain is outside the allowable range.

```
      write(6,600) imax
600   format(5x,'imax = ',i3,' is out of bound' )
```

* This write statement generates an error message when the j (x) dimension of the fracture domain is outside the allowable range.

```
      write(6,610) jmax
610   format(5x,'jmax = ',i3,' is out of bound' )
```

Information Only

* This write statement generates an error message when the k (y) dimension of the fracture domain is outside the allowable range.

```
                write(6,620) kmax
620       format(5x,'kmax = ',i3,' is out of bound' )
```

* This write statement generates an error message when the *w* parameter for limiter function 3 (see Section 4.2.5) is outside the allowable range. The *w* parameter is set in the input control file by parameter *CONTROL – LIMITER – W0; see [18], Section 8.1.4.

```
          write(6,630) w0
630       format(5x,'w0 = ',f7.3,' is out of bound' )
```

* This write statement generates an error message when the number of time steps is less than 0. The number of time steps is set in the input control file by parameter *TIME – NUM_STEP; see [16], Section 8.1.4.

```
          write(6,650) numstep
650       format(5x,'numstep = ',i8,' is out of bound' )
```

* This write statement generates an error message when the number of time steps between printing concentrations is less than 0. The number of time steps between printing concentrations is set in the input control file by parameter *OUTPUT – STEP; see [18], Section 8.1.4.

```
          write(6,660) nprint
660       format(5x,'nprint = ',i4,' is out of bound' )
```

* This write statement generates an error message when the start time is greater than the end time. Start and end times are set in the input control file by parameters *TIME – START_TIME and END_TIME, respectively; see [16], Section 8.1.4.

```
          write(6,670) time_begin, time_end
           670      format(5x,'time_begin = ',1pe10.4,'  is larger than ',
              >             'time_end = ',1pe10.4 )
```

Information Only

10.0 REFERENCES

1. Long, J. J. 2020. Nuclear Waste Management Procedure NP 19-1 Software Requirements, Revision 19. Sandia National Laboratories, Carlsbad, NM. ERMS #573707.

2. van Leer, B. (1979), "Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method", J. Com. Phys.., 32, 101–136.

3. Rechard, R. P., ed., "User's Reference Manual for CAMCON: Compliance Assessment Methodology Controller Version 3.0," Sandia Report SAND90-1983 UC-721, August 1992.

4. Bethune, J.C. 2022. "Implementation Document for SECOTP2D Version 1.44." Carlsbad, NM: Sandia National Laboratories. ERMS #573880

5. Bethune, J.C. 2022. "Design Document and User's Manual for POSTSECOTP2D Version 1.06." Carlsbad, NM: Sandia National Laboratories. ERMS #573842

6. Huyakorn, P. S. and Pinder, G. F., Computational Methods in Subsurface Flow, Academic Press, New York, 1983.

7. 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. Res., Vol. 19, No. 3, pp. 841 - 854, 1983.

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

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

10. Harbaugh, A.W., E.R. Banta, M.C. Hill, and M.G. McDonald. 2000. MODFLOW-2000: The U.S. Geological Survey Modular Ground-Water Model–User Guide to Modularization Concepts and the Ground-Water Flow Process. Open File Report 00-92. Reston, VA: U.S. Geological Survey.

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. Salari, K., Knupp, P., Roache, P., and Steinberg, S., 'TVD Applied to Radionuclide Transport in Fractured Porous Media,' Proceedings IX International Conference on Computational

Information Only

Methods in Water Resources, Denver, Colorado, 9-12 June 1992, T. Russell, et al., eds., pp. 141-148.

15. Ramsey, J. and Treadway, A., 'Test of Mass Balance of SECOTP2D', WPO 44700, pg. 22-23.

16. Bethune, J.C. 2022. "Design Document and User's Manual for PRESECOTP2D v1.24." Carlsbad, NM: Sandia National Laboratories. ERMS #573861.

Information Only

Information Only

11.0 APPENDIX A: DERIVATION OF THE IMPLICIT FRACTURE-MATRIX COUPLING PROCEDURE EMPLOYED BY SECOTP2D

The partial differential equation (PDE) solved by SECOTP2D to predict radionuclide transport of the *k*th species within the matrix portion of a dual-porosity system is given by Salari et al. [14] as

$$\frac{\partial}{\partial \chi}\left(\phi' D_k' \frac{\partial C_k'}{\partial \chi}\right) = \phi' R_k' \left(\frac{\partial C_k'}{\partial t}\right) - \phi' R_k' \lambda_k C_k' + \phi' R_{k-1}' \lambda_{k-1} C_{k-1}' \tag{47}$$

where k indexes the radionuclide decay chain in order from parent to daughter, ϕ' (dimensionless) is the matrix porosity defined as the matrix pore volume per unit volume of matrix, D_k' (m^2/s) is the effective matrix diffuse coefficient with $D_k' = \tau' D_k^*$ where τ' is the matrix tortuosity (dimensionless) and D_k^* (m^2/s) is the free water molecular diffusion coefficient, R_k' (dimensionless) is the matrix retardation coefficient, λ_k (1/s) is the decay constant, C_k' (kg/m^3) is the isotope concentration, and χ, t are the spatial and temporal coordinates, respectively. Eq. 47 is the same as Eq. 3.

The 1D spatial coordinate system is defined such that $\chi = 0$ at the point of symmetry in the center of the matrix block, and $\chi = L$ at the interface between the matrix and fracture.

The initial and boundary conditions imposed on Eq. 47 are

$$C_k'(\chi, 0) = 0 \tag{48}$$

$$\frac{\partial C_k'}{\partial \chi}(0, t) = 0 \tag{49}$$

$$C_k'(L, t) = C_k(x, y, t) \tag{50}$$

where L (m) is one half of the matrix block length and $C_k(x, y, t)$ is the concentration of species k in the fracture at the associated location.

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

$$\Gamma_k = -\frac{2\phi}{b}\left(D_k' \left.\frac{\partial C_k'}{\partial \chi}\right|_{\chi=L}\right) \tag{51}$$

Information Only

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

$$b = \frac{\phi L}{1-\phi} \tag{52}$$

In formulating SECOTP2D the governing PDEs are transformed from physical to computational coordinates 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.

Following the coordinate transformation, the resulting PDEs are discretized using a general three-level implicit finite volume scheme ([12], p. 229, 317). The matrix equation is discretized spatially using a block-centered approach as shown in Figure 4. Imaginary or "ghost" cells are located at both ends of the matrix block to impose the boundary conditions stipulated in Eq. 49 and Eq. 50.

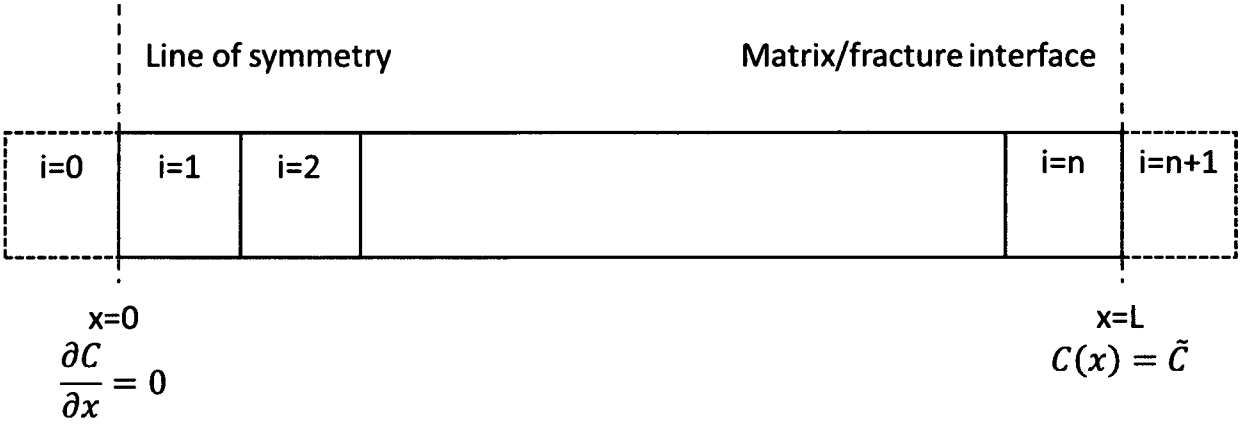


Figure 4. Matrix discretization.

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

$$\mathbf{A}^n \Delta C'^n = F^{n-1} \tag{53}$$

where the superscript n denotes the time level, $\Delta C'^n$ is a M+2 vector containing the elements $\Delta C_i'^n = C_i'^n - C_i'^{n-1}$, F^{n-1} is a M+2 vector containing the forcing elements, and $\mathbf{A}^n$ is a tridiagonal matrix possessing a structure of

Information Only

$$\mathbf{A} = \begin{bmatrix} a2_0 & a3_0 & 0 & 0 & 0 & 0 & 0 \\ a1_1 & a2_1 & a3_1 & 0 & 0 & 0 & 0 \\ 0 & a1_2 & a2_2 & a3_2 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & a1_n & a2_n & a3_n \\ 0 & 0 & 0 & 0 & 0 & a1_{n+1} & a2_{n+1} \end{bmatrix} \tag{54}$$

The linear system in Eq. 53 is solved at each time step at each location in 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 $\Delta\Gamma^n$ as a function of the fracture concentration

$$\Delta\Gamma^n = \gamma\Delta C^n + \delta \tag{55}$$

Once the coefficients γ and δ of Eq. 55 are determined, $\Delta\Gamma^n$ is eliminated from the system of equations developed from the fracture PDE. The fracture system is then solved yielding ΔC^n 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 ΔC^n is used as a boundary condition.

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

$$\mathbf{LU} = \mathbf{A}^n \tag{56}$$

$$\mathbf{LG} = F^{n-1} \qquad \text{forward elimination} \tag{57}$$

$$\mathbf{U}\Delta C'^n = G \qquad \text{back substitution} \tag{58}$$

where,

$$\mathbf{L} = \begin{bmatrix} \alpha_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha 1_1 & \alpha_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha 1_2 & \alpha_2 & 0 & 0 & 0 & 0 \\ & & & & & & \\ & & & & & & \\ 0 & 0 & 0 & 0 & \alpha 1_n & \alpha_n & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha 1_{n+1} & \alpha_{n+1} \end{bmatrix} \tag{59}$$

Information Only

$$\mathbf{U}=\begin{bmatrix} 1 & \beta_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \beta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \beta_2 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 1 & \beta_n \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{60}$$

$$\alpha_0 = a2_0 \tag{61}$$

$$G_0 = F_0/\alpha_0 \tag{62}$$

$$\alpha_i = a2_i - a1_i\beta_{i-1}, \quad \text{for } i=1,n+1 \tag{63}$$

$$\beta_i = a3_i/\alpha_i, \quad \text{for } i=0,n \tag{64}$$

$$G_i = \left(F_i - a1_i G_{i-1}\right)/\alpha_i \quad \text{for } i=1,n+1 \tag{65}$$

From Eq. 51 and the relationship $\Delta\Gamma^n = \Gamma^n - \Gamma^{n-1}$, it can be shown that

$$\Delta\Gamma^n = -\frac{2\phi D'}{b}\left(\frac{\partial C^n}{\partial \chi} - \frac{\partial C^{n-1}}{\partial \chi}\right)\Bigg|_{\chi=L} \tag{66}$$

By approximating the partial derivatives in Eq. 66 with the centered finite different formula

$$\left.\frac{\partial C'}{\partial \chi}\right|_{\chi=L} = \frac{2\left(C'_{M+1} - C'_M\right)}{\Delta\chi_{M+1} + \Delta\chi_M} \tag{67}$$

and restricting $\Delta\chi_{M+1} = \Delta\chi_M$ a second-order approximation of Eq. 66 is obtained

$$\Delta\Gamma^n = -\frac{2\phi D'}{b}\frac{1}{\Delta\chi_M}\left(\Delta C''^n_{M+1} - \Delta C''^n_M\right) \tag{68}$$

where $\Delta C''^n_M = C''^n_M - C''^{n-1}_M$. The expression $\left(\Delta C''^n_{M+1} - \Delta C''^n_M\right)$ in Eq. 68 is obtaind by beginning the back substitution process of the LU decomposition. From Eq. 58 and 60

$$\Delta C''^n_{M+1} = G_{M+1} \tag{69}$$

$$\Delta C''^n_M = G_M - \beta_M G_{M+1} \tag{70}$$

and it follows that

$$\Delta C''^n_{M+1} - \Delta C''^n_M = G_{M+1}\left(1+\beta_M\right) - G_M \tag{71}$$

Information Only

Substituting Eq. 71 into Eq. 68 yields

$$\Delta\Gamma^n = -\frac{2\phi D'}{b}\frac{1}{\Delta\chi_M}\left(G_{M+1}\left(1+\beta_M\right)-G_M\right) \tag{72}$$

Eq. 72 is a second-order approximation of the mass transfer function in terms of the M and M+1 coefficients of the LU decomposition. Eq. 71 takes the form of Eq. 55 when the term G_{M+1} is expressed using Eq. 65, and the Dirichlet boundary condition given by Eq. 50 is imposed on the solution. From Eq. 65, G_{M+1} can be expressed as

$$G_{M+1} = \frac{F_{M+1} - a1_{M+1}G_M}{\alpha_{M+1}} \tag{73}$$

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

$$\Delta C^n = \frac{\Delta C''^n_{M+1} + \Delta C''^n_M}{2} \tag{74}$$

and the boundary condition is imposed on the solution by setting $a1_{M+1} = a2_{M+1} = 1$ in Eq. 54 and $F_{M+1} = 2\Delta C^n$.

Substituting $F_{M+1} = 2\Delta C^n$ and Eq. 73 into Eq. 74, yields

$$\Delta\Gamma^n = -\frac{2\phi D'}{b}\frac{1}{\Delta\chi_M}\left[\frac{2\left(1+\beta_M\right)}{\alpha_{M+1}}\Delta C^n - G_M\left(\frac{a1_{M+1}\left(1+\beta_M\right)}{\alpha_{M+1}}+1\right)\right] \tag{75}$$

which is equivalent to Eq. 55 with the coefficients

$$\gamma = -\frac{2\phi D'}{b}\frac{1}{\Delta\chi_M}\frac{2\left(1+\beta_M\right)}{\alpha_{M+1}} \tag{76}$$

$$\delta = \frac{2\phi D'}{b}\frac{1}{\Delta\chi_M}\left[G_M\left(\frac{a1_{M+1}\left(1+\beta_M\right)}{\alpha_{M+1}}+1\right)\right] \tag{77}$$

Information Only

12.0 APPENDIX B: SAMPLE INPUT CONTROL FILE

```
&INPUTS
 FILE_VEL="st2d2_preseco_test1.vel

 FILE_PROP="st2d2_preseco_test1.prp

 NCH=1
 NSP=1
 PRINT_SCR=F,
 NPRINT=1000
 DUAL_POROSITY=T,
 IMAX=20
 JMAX=140
 KMAX=1
 NUMSTEP=40000
 TIME_BEGIN=  0.0000000000000000
 TIME_END=  631140000000.00000
 VAR_DT=F,
 COMPUTE_DT=T,
 DT=  0.0000000000000000
 INITIAL_COND=F,
 STEADY_UV=T,
 COMPUTE_DISCHARGE=F,
 THETA=  1.0000000000000000
 OMEGA=  0.0000000000000000
 AX=  1.0000000000000000
 AY=  0.0000000000000000
 LIMITER=1
 W0=  0.0000000000000000
 POINT_SOURCE=F,
 PRINT_MATRIX_CON=T,
 NWRITE_DISCHARGE=100
 /
```

Information Only

13.0 APPENDIX C: SAMPLE DIAGNOSTICS FILE

```
 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 1.44
                          PRODUCTION Built 08/20/2020
                            Written by Kambiz Salari

                         Run on 08/20/2020 at 14:17:13
                       Run on x86_64 cb001 Linux 3.10.0-957
SECOTP2D 1.44 PRODUCTION 08/20/2020                          08/20/2020 14:17:13


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

          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:
-----------------
```

Information Only

```
SECOTP2D input file:
   SECOTP2D/Test/Output/Linux_144/st2d2_preseco_test1.seco
      Written on 08/20/2020 14:17:49
Property data binary input file:
   SECOTP2D/Test/Output/Linux_144/st2d2_preseco_test1.prp
      Written on 08/20/2020 14:17:49
Velocity data binary input file:
   SECOTP2D/Test/Output/Linux_144/st2d2_preseco_test1.vel
      Written on 08/20/2020 14:17:49
SECOTP2D binary output file:
   SECOTP2D/Test/Output/Linux_144/st2d2_144_test1.bin
SECOTP2D diagnostics file:
   SECOTP2D/Test/Output/Linux_144/st2d2_144_test1.dbg


    number of chains  =  1
    number of species =  1


    chain:  1     number of species  1
         species 1 = SX


    dual-porosity is used

    number of cells in the block, imax = 20


    number of cells in x-direction, jmax = 140
    number of cells in y-direction, kmax =   1


    time_begin = 0.0000E+00     time_end = 6.3114E+11


    time step is computed, dt = 1.5778E+07


    type of input velocity: steady state


    maximum number of steps, numstep =     40000


    algorithm parameters:
        first order implicit euler time differencing is used.
        coefficient of source term:   ax = 1.00    ay = 0.00
        spatial differencing:
           centeral, limiter = 1


    property and velocity field data files:
        property file = SECOTP2D/Test/Output/Linux_144/st2d2_preseco_test1.prp
        velocity file = SECOTP2D/Test/Output/Linux_144/st2d2_preseco_test1.vel


    intermediate results are printed at every       1000  time steps.


Mass balance errors (unit thickness):
   SX       -8.57E-11
```

Information Only

```
### Maximum mass balance error at end = 8.57E-11, all steps = 8.57E-11, Limit = 1.05
```

Information Only