

A Numerical Algorithm for Fluid Flow in 3D Naturally Fractured Porous Media

Seongjai Kim*

Abstract

Fluid flow in three-dimensional (3D) fractured porous media is considered. The governing system of partial differential equations consists of two subsystems – one describing the flow in the fractures, and the other describing the flow in the matrix blocks. In this paper, we develop an efficient algorithm for the numerical solution of the problem. An operator splitting technique is employed, as a part of the time-stepping procedure, to decouple the system into easy subsystems. The fracture concentration equation is discretized by the modified method of characteristics (MMOC) in time due to high velocity in the fractures and by the Raviart-Thomas-Nedelec mixed method of index zero (RTN0) in space. The matrix concentration equation is discretized by a backward Euler scheme and the linear finite element method. The pressure equation is approximated by RTN0 and the linear Galerkin method for the fractures and the matrix blocks, respectively. For the fracture system, a domain decomposition (DD) iterative procedure is suggested with its convergence analysis. Numerical results are shown to demonstrate the efficiency of the algorithm.

Key words. Dual-porosity, fractured porous medium, modified method of characteristics, operator splitting, domain decomposition method.

AMS subject classifications. 65N55, 65F10, 76M10

1. Introduction

Groundwater simulation is an important environmental problem. Liquids of contaminants, such as nuclear wastes and non-aqueous phase liquids (NAPLs), can be leaking from waste sites. It is interesting to track the movement of contamination through the earth, a 3D naturally fractured porous medium. In realistic earth media, one can easily consider *fractures* which separate the porous rocks (the *matrices*) into a collection of blocks. Most of the fluid is in the matrix blocks, where it moves slowly.

*Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027 USA
Email: skim@ms.uky.edu

In contrast, the fluid in the fractures flows comparatively fast, since the fractures form paths of high permeability. Thus, a fractured porous medium possesses an additional length-scale compared to an unfractured medium. Such fractured reservoirs could be modeled by permitting the porosity and permeability to vary rapidly and discontinuously over the reservoir (the single-porosity model); see [6, 39, 41] for a standard description of flow in porous media. Both of these quantities are significantly larger in the fractures than in the matrix blocks, and therefore the computational and data requirements for the model would be impractical. Such difficulties could be overcome by replacing these quantities by their locally averaged smooth values. In this case, the numerical results are not satisfactory; they would not show delay effects related to the blocks that is necessary to correlate with physical observations.

Alternatively, we can treat the porosity and permeability as if the reservoir has *two* porous structures, one for the fractures and the other for the matrix blocks, as in the *dual-porosity model* developed by Arbogast, Douglas and their collaborators [1, 3, 4, 9, 13, 14]. In that model the interchange of fluid between the fractures and matrix blocks would be obtained by imposing continuity of both the concentration and the pressure on the boundaries of matrix blocks, and by considering the change of concentration in matrix blocks as a source for the fractures. The fractures, even though very thin, play a profound role in the flow of fluids in porous media [3, 4].

In this paper, we develop an efficient algorithm for the numerical solution of the dual-porosity model. An operator splitting technique is suggested as a part of the time-stepping procedure, to decouple the system into a series of easy subsystems. The fracture concentration equation is discretized by the modified method of characteristics (MMOC) in time, due to high velocity in the fractures, and by the Raviart-Thomas-Nedelec mixed method of index zero (RTN0) in space [38, 40]. The matrix concentration equation is discretized by a backward Euler scheme and the linear finite element method. The pressure equation is approximated by the RTN0 and the linear Galerkin method for the fractures and the matrix blocks, respectively. For the fracture system that requires a global solution, a nonoverlapping domain decomposition (DD) iterative procedure is employed: A Robin-type interface operator on the subdomain interfaces is considered, no course grid solver is applied, and the iteration is performed in a block-Jacobi manner.

An outline of the paper is as follows. In Section 2, we present the dual-porosity model governing flows in fractured porous media. Section 3 introduces MMOC and develops an operator splitting algorithm in which the computation of the fractures is decoupled from that of the matrix blocks. Remarks regarding to the implementation of the algorithm are added. In Section 4, two DD methods are introduced for solving the fracture equations by RTN0. In general convergence behavior of DD methods strongly depend on the amount of overlap and coarse grid solvers; see [25, and ref-

erences therein]. The coarse grid problems are often very large for highly oscillatory solutions or for general domains. The efficiency of the algorithms in these cases would be degenerated. As an alternative, we consider a nonoverlapping DD method and a minimum overlapping DD method incorporating a *modified* Robin interface boundary condition (RIBC) for solving the fracture velocity and the fracture concentration, respectively. Relaxation parameters are introduced to accelerate the convergence of the DD methods. Section 5 presents numerical results showing the efficiency of the algorithm.

2. The dual-porosity model

Homogenization techniques have been widely used for modeling the flow of two-phase, incompressible, immiscible fluids in naturally fractured petroleum reservoirs [1, 3, 4, 9, 14]. The models are derived as a system of equations (the equations of the flow in the matrix blocks and in the fractures), beginning from a microscopic description of the flow field by a standard immiscible model [5, 6, 19] with physical parameters varying rapidly in space. The flow in the fractures has been averaged by the homogenization process in such a way so as to take place everywhere in the fracture sheet; not just at the physical locations of the individual fractures.

For miscible flows, an analogous description is applicable [1]. Let $\Omega \subset \mathbb{R}^3$ be a cubic (or *logically cubic*) region for convenience and Ω_x be the disjoint matrix block which is attached to the point $x = (x_1, x_2, x_3)$ in the fractures. Then $\Omega = (\cup_x \Omega_x) \cup \Omega_f$, where Ω_f is the fractures. In this section, we will use a local variable $y = (y_1, y_2, y_3)$ to indicate points inside a matrix block. Capital letters are used for fracture variables, while the variables for matrix blocks are denoted by lower-case ones.

Within each matrix block the flow can be described by Darcy's law and a mass conservation equation. Let $J = [0, T]$, $T > 0$, be the time interval. Then the equations of flow in matrix blocks Ω_x are formulated as follows:

$$\begin{aligned}
 \text{(a)} \quad & \phi \frac{\partial c}{\partial t} + \nabla_y \cdot (\mathbf{v}c - D(\mathbf{v})\nabla_y c) = q_m, & \Omega_x \times J, \\
 \text{(b)} \quad & \nabla_y \cdot \mathbf{v} = 0, & \Omega_x \times J, \\
 \text{(c)} \quad & \mathbf{v} = -\frac{k}{\mu} \nabla_y \psi, \quad \psi = p - \rho g z, & \Omega_x \times J,
 \end{aligned} \tag{2.1}$$

where c is the concentration of the contaminant in matrix blocks $\cup_x \Omega_x$, ϕ is the porosity, \mathbf{v} is the the total volumetric flow rate in the matrix block, k is the permeability of the matrix block, and μ denotes the viscosity which will be assumed constant, reflecting the fact that the contaminant is expected to have a quite low concentration. The potential ψ is defined by the difference between the fluid pressure p and the force from gravity effect. Here ρ is the fluid density, g is the gravity coefficient, and z is

the vertical coordinate. The diffusion-dispersion tensor D can be approximated by

$$D(\mathbf{v}) = D_{\text{diff}}(\mathbf{v}) + D_{\text{disp}}(\mathbf{v}), \quad (2.2)$$

where

$$\begin{aligned} D_{\text{diff}}(\mathbf{v}) &= \phi(x) d_m I, \\ D_{\text{disp}}(\mathbf{v}) &= \phi(x) |\mathbf{v}| \{d_\ell E(\mathbf{v}) + d_t (I - E(\mathbf{v}))\} \\ &= \phi(x) \frac{d_\ell - d_t}{|\mathbf{v}|} \begin{bmatrix} u_1^2 & u_1 u_2 & u_1 u_3 \\ u_2 u_1 & u_2^2 & u_2 u_3 \\ u_3 u_1 & u_3 u_2 & u_3^2 \end{bmatrix} + \phi(x) d_t |\mathbf{v}| I. \end{aligned} \quad (2.3)$$

Here, d_m is the molecular diffusion coefficient, d_ℓ and d_t are longitudinal and transversal dispersion coefficients, respectively, and $E(\mathbf{v})$ denotes the projection along the vector \mathbf{v} , i.e. $E(\mathbf{v}) = \mathbf{v}\mathbf{v}^T/|\mathbf{v}|^2$. The quantity q_m is the source function for the matrix blocks and is often assumed to be zero.

Now, we introduce a set of equations governing the flow in the fractures as follows:

$$\begin{aligned} \text{(a)} \quad & \Phi \frac{\partial C}{\partial t} + \nabla_x \cdot (\mathbf{V}C - D(\mathbf{V})\nabla_x C) = q_f + \tau_{m,f}, & \Omega \times J, \\ \text{(b)} \quad & (D(\mathbf{V})\nabla_x C) \cdot \nu = 0, & \Gamma \times J, \\ \text{(c)} \quad & \nabla_x \cdot \mathbf{V} = 0, & \Omega \times J, \\ \text{(d)} \quad & \mathbf{V} = -\frac{K}{\mu} \nabla_x \Psi, \quad \Psi = P - \rho g z, & \Omega \times J, \\ \text{(e)} \quad & \mathbf{V} \cdot \nu_{x_1} = q_b, \quad \mathbf{V} \cdot \nu_{x_2} = 0, \quad \mathbf{V} \cdot \nu_{x_3} = 0, & \Gamma \times J. \end{aligned} \quad (2.4)$$

Here we have used analogous notations (capitals) to those for the matrix blocks. Equations (2.4b) and (2.4e) specify the boundary conditions for the concentration C and the total volumetric flow rate \mathbf{V} , respectively, in the fractures. The quantity q_b is the flow rate on the boundary faces perpendicular to x_1 -axis. (It is a choice in this article, considering the case that the fluid moves with a dominant direction.) For initial values, we assume that there is no contaminant in the reservoir; i.e., when the simulation starts,

$$c = C = 0, \quad t = 0. \quad (2.5)$$

We impose the following boundary conditions on the boundaries of matrix blocks:

$$\psi = \Psi, \quad \text{and} \quad c = C, \quad y \in \partial\Omega_x. \quad (2.6)$$

Equation (2.6) imposes the continuity of the potential and the concentration on the matrix block-fracture interfaces. Also, it implies that the change of contaminant in the matrix blocks contributes to the the fracture system as a source. The matrix

source term $\tau_{m,f}$ is defined as follows: The volume of matrix fluid leaving the block Ω_x is

$$\int_{\partial\Omega_x} [\mathbf{v}c - D(\mathbf{v})\nabla_y c] \cdot \nu d\sigma = \int_{\Omega_x} \nabla_y \cdot [\mathbf{v}c - D(\mathbf{v})\nabla_y c] dV = - \int_{\Omega_x} \left[\phi \frac{\partial c}{\partial t} - q_m \right] dV;$$

consequently, let

$$\tau_{m,f} = - \frac{1}{|\Omega_x|} \int_{\Omega_x} \left[\phi \frac{\partial c}{\partial t} - q_m \right] dV. \quad (2.7)$$

Equations (2.1)-(2.7) describe contaminant transport in naturally fractured porous media. In the next section, we discuss an efficient operator splitting scheme for the system of the equations.

3. Time-stepping and operator splitting

In reservoir simulation of convection-dominated problems, the modified method of characteristics (MMOC) has been used widely [24, 26]. However, MMOC often fails to preserve certain integral identities satisfied by the solution of the differential system. Recently, Douglas and his colleagues developed the so called modified method of characteristics with adjusted advection (MMOCAA) and locally conservative Eulerian-Lagrangian method (LCELM) and applied to various reservoir simulations. MMOCAA preserves the global form of the identity associated with conservation of mass, but does not preserve a localized form of the identity [12, 15, 20, 42]. LCELM has been designed to satisfy the local conservation of mass [17, 21, 22]. However, LCELM is very expensive, in particular, in 3D; mass-conservative incomplete characteristic methods (ICMs) are being developed to overcome the high cost of LCELM [16].

The main purpose of the section is to present an operator splitting technique to decouple the system into easy subsystems. For the basic time-stepping scheme, either of MMOC, MMOCAA, LCELM, and ICM can be applied. Here we will choose MMOC for a simple presentation.

3.1. The modified method of characteristics

Since the flow in the fracture system is convection-dominated, standard finite element discretization can introduce nonphysical oscillations and upwind finite difference schemes may smear out the solution (numerical diffusion). MMOC has been a successful technique for solving such convection-dominated problems [2, 24, 26]. It is a time-stepping procedure which can be combined with any spatial discretization. In spite of the difficulties involved for an efficient implementation, MMOC is

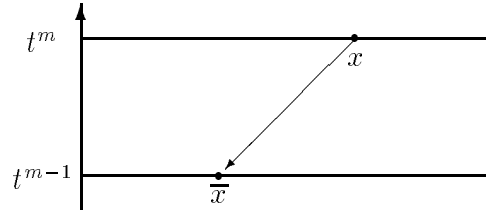


Figure 1: The method of characteristics.

very appealing, for at least two reasons: a clear physical meaning and good stability properties of the methods. (These are true for MMOCAA, LCELM, and ICM, too.)

The basic idea of the method is to consider the hyperbolic part $\Phi C_t + \mathbf{V} \cdot \nabla C$ as a directional derivative. Let s be the unit vector in the direction of (V_1, V_2, V_3, Φ) . Set

$$\gamma(x) = \sqrt{|\mathbf{V}(x)|^2 + \Phi(x)^2}.$$

Since

$$\Phi \frac{\partial C}{\partial t} + \mathbf{V} \cdot \nabla C = \gamma \frac{\partial C}{\partial s},$$

(2.4a) can be rewritten as

$$\gamma \frac{\partial C}{\partial s} - \nabla \cdot (D(\mathbf{V})\nabla C) = q_f + \tau_{m,f}. \quad (3.1)$$

Note that (3.1) has the form of a heat equation, so that its numerical solutions would be better behaved than those of (2.4) provided that a reasonable discretization for $\partial C/\partial s$ can be found.

Partition J into $0 = t^0 < t^1 < \dots < t^N = T$, with $\Delta t^m = t^m - t^{m-1}$; for a function f defined on $\Omega \times J$, we write $f^m(x)$ for $f(x, t^m)$. Approximate $\frac{\partial C^m}{\partial s}(x) = \frac{\partial C}{\partial s}(x, t^m)$ by the backward difference (see Figure 1)

$$\begin{aligned} \gamma(x) \frac{\partial C^m}{\partial s}(x) &\cong \gamma(x) \frac{C(x, t^m) - C(\bar{x}, t^{m-1})}{[|x - \bar{x}|^2 + (\Delta t^m)^2]^{1/2}} \\ &= \Phi(x) \frac{C(x, t^m) - C(\bar{x}, t^{m-1})}{\Delta t^m}, \end{aligned} \quad (3.2)$$

where

$$\bar{x} = x - \frac{\mathbf{V}(x)}{\Phi(x)} \Delta t^m.$$

Even though x is a nodal point of the discretized mesh, the point \bar{x} is not a node in general, so that the value $C(\bar{x}, t^{m-1})$ must be computed by interpolation from the values at neighboring nodes. Thus, the domain of dependence of the node x is

enlarged: this may cause a numerical diffusion. The practical efficiency of MMOC strongly depends on the ability of the numerical code to track back the foot of the characteristic line departing from x (i.e. the point \bar{x}) and to evaluate the value $C(\bar{x}, t^{m-1})$. It should be noticed that the method is unconditionally stable and the *numerical* domain of dependence of the point x at time t (i.e. the set of points x' where a value at the previous time $C(x', \hat{t})$, $\hat{t} < t$, is used to compute $C(x, t)$) always contains the *physical* domain of dependence.

The point \bar{x} may be outside of the domain for x near $\partial\Omega$. Since we are interested mainly in the behavior of the flow in the interior of the domain, we will assume that the domain is Ω -periodic or $\bar{C}^{m-1} := C(\bar{x}, t^{m-1}) = 0$ if $\bar{x} \notin \Omega$. (In this paper, we choose the latter. For certain problems, one can reflect the characteristic back into the domain as well.) Also, we assume that the velocity of the fluid is not affected by the injection of the contaminants, i.e. \mathbf{V} (and therefore \mathbf{v}) is independent on time. For now, we will omit the subscripts x and y used to indicate the fracture variable and the matrix block variable, respectively.

The time discretization for (2.1)-(2.7) can be formulated as follows.

Find $\{c^m, C^m\}$, $m \geq 1$, satisfying (3.3)-(3.7):

Matrix blocks

$$\begin{aligned} \phi \frac{c^m - c^{m-1}}{\Delta t^m} + \mathbf{v} \cdot \nabla c^m - \nabla \cdot (D(\mathbf{v}) \nabla c^m) &= q_m^m, & \Omega_x, \\ \nabla \cdot \mathbf{v} &= 0, & \Omega_x, \\ \mathbf{v} &= -\frac{k}{\mu} \nabla \psi, & \Omega_x. \end{aligned} \quad (3.3)$$

Fracture system

$$\begin{aligned} \Phi \frac{C^m - \bar{C}^{m-1}}{\Delta t^m} - \nabla \cdot (D(\mathbf{V}) \nabla C^m) &= q_f^m + \tau_{m,f}^m, & \Omega, \\ (D(\mathbf{V}) \nabla C^m) \cdot \nu &= 0, & \Gamma, \\ \nabla \cdot \mathbf{V} &= 0, & \Omega, \\ \mathbf{V} &= -\frac{K}{\mu} \nabla \Psi, & \Omega, \\ \mathbf{V} \cdot \nu_{x_1} = q_b, \quad \mathbf{V} \cdot \nu_{x_2} = 0, \quad \mathbf{V} \cdot \nu_{x_3} = 0, & & \Gamma. \end{aligned} \quad (3.4)$$

Initial condition

$$c^0 = C^0 = 0. \quad (3.5)$$

Continuity condition

$$\psi = \Psi, \text{ and } c^m = C^m, \quad y \in \partial\Omega_x. \quad (3.6)$$

Source from matrix blocks

$$\tau_{m,f}^m = -\frac{1}{|\Omega_x|} \int_{\Omega_x} \left[\phi \frac{c^m - c^{m-1}}{\Delta t^m} - q_m^m \right] dV. \quad (3.7)$$

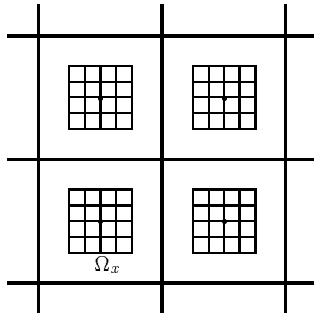


Figure 2: The computational domain illustrated in 2D.

When the system of equations (3.3)-(3.7) is spatially discretized by the standard finite element methods or mixed methods, it is not easy to find an iterative algorithm which solves the coupled system with the conditions (3.6) and (3.7) satisfied. In §3.2, we develop a four-step (direct) algorithm.

Remark. The injection (or transport) of the contaminants may affect the flow velocity. (The viscosity μ can depend on the concentrations C and c , or it can be multiplied by some quantity which is dependent on the concentrations.) In that case, one should recompute the velocity at each timestep or after several timesteps. Since the velocity varies relatively slowly, one can use the velocity approximation for \mathbf{V}^m which is based on \mathbf{V}^{m-1} and earlier values. For example, the velocity \mathbf{V} in the concentration equation in (3.4) can be approximated by the following extrapolation

$$\mathbf{V}_E^m := \begin{cases} \left(1 + \frac{\Delta t^m}{\Delta t^{m-1}}\right) \mathbf{V}^{m-1} - \frac{\Delta t^m}{\Delta t^{m-1}} \mathbf{V}^{m-2}, & m \geq 2, \\ \mathbf{V}^0, & m = 1. \end{cases} \quad (3.8)$$

Then, one can find \mathbf{V}^m by solving the pressure equation in (3.4) using the value C^m .

3.2. Operator splitting

In this section, we develop a numerical scheme in which the computations of the fractures are decoupled from those of matrix blocks at each timestep t^m . It is a four-step process to carry out the computation for C^m and c^m from the former step solutions; it is dependent of the spatial discretization methods and the interpolation schemes applied. Note that the problem requires the continuity of C^m and c^m on the matrix block boundaries. The continuity can be satisfied by matching the values between c^m and the interpolation of C^m on the boundaries of matrix blocks.

We consider the RTN0 for the fracture concentration C and the trilinear finite element method for the matrix block concentration c . For an error analysis for these

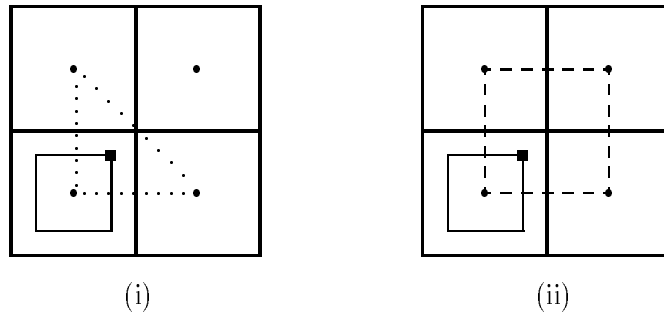


Figure 3: Interpolation schemes in 2D.

methods, see [1]. Consider the computational domain depicted in Figure 2, where the bold lines denote the grid lines for the fractures. (For clarity the presentation would be illustrated in 2D. The same is in Figures 2, 3, 4, and 5.) Dots at the center of the fracture elements denote the degrees of freedom for the fracture concentration. Matrix blocks Ω_x which are partitioned by thin lines are attached to the centroids of the fracture elements. For convenience, we consider a cubic domain Ω and cube-shaped finite elements. Let (N_x, N_y, N_z) be the number of fracture finite elements in each direction. Let the matrix blocks, attached in fracture elements, be partitioned into (m_x, m_y, m_z) finite elements. Here, m_x , m_y , and m_z can be chosen to be different in different matrix blocks. We assume the following:

(A₁). Each finite element in a matrix block Ω_x has the same volume V_x , i.e., each matrix block is partitioned into uniform cells.

The assumption is imposed to significantly reduce storage requirements and computational costs. In different matrix blocks, the volume of elements may be different.

Now, we consider interpolation schemes. Two interpolation schemes (in 2D) are depicted in Figure 3. The value of the fracture concentration C^m at a corner point of the matrix block Ω_x denoted by ■ can be obtained by linear (Figure 3.(i)) or bilinear (Figure 3.(ii)) interpolation of the neighboring quantities. Similarly, one can consider linear and trilinear interpolations in 3D. We will use the linear interpolation scheme (using 4 fracture elements in 3D). Let the corners of matrix blocks be ordered in the row-wise manner: x -direction first, y -direction next, and z -direction the last. Let C_i^m , $i = 1, 2, \dots, 8$, be the interpolated value of C^m at the i -th corner of the matrix block Ω_x . The quantities C_i^m can be expanded in a bilinear manner to determine the values of c^m on $\partial\Omega_x$.

We assume the following for the porosity of the matrix blocks:

(A₂). The porosity ϕ in each matrix block Ω_x is constant.

The above is again assumed to reduce the storage requirements. The right side in (3.7) will be integrated numerically. Note that c^m is a piecewise linear function with each element having the same volume V_x . So,

$$\int_{\Omega_x} c^m dV = V_x \sum_{\ell} c_{\ell}^m,$$

where the summation is carried out on the finite elements of the matrix block Ω_x and c_{ℓ}^m is the value of c^m at the centroid of the ℓ -th element of Ω_x . The matrix source q_m^m is often assumed to be zero; in this paper, we assume that q_m^m is such that the numerical quadrature using the centroids of the elements is exact.

Now, we are ready to present the operator splitting scheme. First, we need to solve the pressure equations: Find $\{\Psi, \mathbf{V}\}$ such that

$$\begin{aligned} \nabla \cdot \mathbf{V} &= 0, & \Omega, \\ \mathbf{V} &= -\frac{K}{\mu} \nabla \Psi, & \Omega, \\ \mathbf{V} \cdot \nu_{x_1} &= q_b, \quad \mathbf{V} \cdot \nu_{x_2} = 0, \quad \mathbf{V} \cdot \nu_{x_3} = 0, & \Gamma; \end{aligned} \quad (3.9)$$

find the velocity \mathbf{v} on each matrix block Ω_x by solving the following Dirichlet boundary value problem

$$\begin{aligned} \nabla \cdot \mathbf{v} &= 0, & \Omega_x, \\ \mathbf{v} &= -\frac{k}{\mu} \nabla \psi, & \Omega_x, \\ \psi &= \Psi, & \partial\Omega_x. \end{aligned} \quad (3.10)$$

The global procedure, obtaining C^m and c^m from C^{m-1} and c^{m-1} , can be written as follows. (In the following operator splitting scheme, the quantities \hat{c}_i^m , \check{c}^m , C^m , and c^m should be viewed as the discrete solutions on the m -th time level.)

Step 1 (Matrix blocks): For each matrix block Ω_x , find \hat{c}_i^m , $i = 1, 2, \dots, 8$, such that

$$\begin{aligned} \phi \frac{\hat{c}_i^m}{\Delta t^m} + \mathbf{v} \cdot \nabla \hat{c}_i^m - \nabla \cdot (D(\mathbf{v}) \nabla \hat{c}_i^m) &= 0, & \Omega_x, \\ \hat{c}_i^m &= \begin{cases} 1, & \text{on the } i\text{-th corner of } \Omega_x, \\ 0, & \text{on the other corners,} \\ \text{and interpolate the values to } \partial\Omega_x \text{ bilinearly.} \end{cases} \end{aligned} \quad (3.11)$$

Step 2 (Matrix blocks): For each matrix block Ω_x , find \check{c}^m such that

$$\begin{aligned} \phi \frac{\check{c}^m - c^{m-1}}{\Delta t^m} + \mathbf{v} \cdot \nabla \check{c}^m - \nabla \cdot (D(\mathbf{v}) \nabla \check{c}^m) &= q_m^m, & \Omega_x, \\ \check{c}^m &= C^{m-1}, & \partial\Omega_x. \end{aligned} \quad (3.12)$$

Then, from (3.11) and (3.12),

$$c^m = \check{c}^m + \sum_{i=1}^8 (C_i^m - C_i^{m-1}) \hat{c}_i^m. \quad (3.13)$$

The matrix concentration c^m in (3.13) is still coupled with C^m . In the next step, c^m (appeared in (3.7) as a source term for the fracture system) would be replaced by other variables using (3.13).

Step 3 (Fracture system): Find C^m such that

$$\begin{aligned} \Phi \frac{C^m - \bar{C}^{m-1}}{\Delta t^m} - \nabla \cdot (D(\mathbf{V})\nabla C^m) &= q_f^m + \tau_{m,f}^m, & \Omega, \\ (D(\mathbf{V})\nabla C^m) \cdot \nu &= 0, & \Gamma, \end{aligned} \quad (3.14)$$

where

$$\begin{aligned} \tau_{m,f}^m &= -\frac{1}{|\Omega_x|} \int_{\Omega_x} \left[\phi \frac{c^m - c^{m-1}}{\Delta t^m} - q_m^m \right] dV \\ &= -\frac{\phi}{\Delta t^m} \frac{1}{|\Omega_x|} \int_{\Omega_x} \left[\check{c}^m + \sum_{i=1}^8 (C_i^m - C_i^{m-1}) \hat{c}_i^m - c^{m-1} \right] dV + \frac{1}{|\Omega_x|} \int_{\Omega_x} q_m^m dV \\ &= -\frac{\phi}{\Delta t^m} \frac{|V_x|}{|\Omega_x|} \sum_{\ell} \left(\check{c}^m + \sum_{i=1}^8 (C_i^m - C_i^{m-1}) \hat{c}_i^m - c^{m-1} \right)_{\ell} + \frac{|V_x|}{|\Omega_x|} \sum_{\ell} q_{m,\ell}^m \\ &= -\sum_{i=1}^8 \left(\frac{\phi}{\Delta t^m} \frac{|V_x|}{|\Omega_x|} \sum_{\ell} \hat{c}_{i,\ell}^m \right) C_i^m + \sum_{i=1}^8 \left(\frac{\phi}{\Delta t^m} \frac{|V_x|}{|\Omega_x|} \sum_{\ell} \hat{c}_{i,\ell}^m \right) C_i^{m-1} \\ &\quad - \frac{\phi}{\Delta t^m} \frac{|V_x|}{|\Omega_x|} \sum_{\ell} (\check{c}^m - c^{m-1})_{\ell} + \frac{|V_x|}{|\Omega_x|} \sum_{\ell} q_{m,\ell}^m. \end{aligned} \quad (3.15)$$

Note that $\tau_{m,f}^m$ does not depend explicitly on c^m . When (3.14) is fully discretized, the resulting algebraic system can be modified by moving the sub-system corresponding to

$$\sum_{i=1}^8 \left(\frac{\phi}{\Delta t^m} \frac{|V_x|}{|\Omega_x|} \sum_{\ell} \hat{c}_{i,\ell}^m \right) C_i^m \quad (3.16)$$

to the left side.

Step 4 (Matrix blocks): For each matrix block Ω_x , recover c^m by evaluating (3.13) or by solving

$$\begin{aligned} \phi \frac{c^m - c^{m-1}}{\Delta t^m} + \mathbf{v} \cdot \nabla c^m - \nabla \cdot (D(\mathbf{v})\nabla c^m) &= q_m^m, & \Omega_x, \\ c^m &= C^m, & \partial\Omega_x. \end{aligned} \quad (3.17)$$

3.3. Technical remarks on the operator splitting

Step 1 can be skipped when $\Delta t^m = \Delta t^{m-1}$. After Step 3 is performed, c^m can be computed by using (3.13). In the case, one should save $\{\hat{c}_i^m : i = 1, 2, \dots, 8\}$ for every nodal point in the matrix blocks; the storage requirement is very high. Step 4 is designed to overcome the high storage requirement. It should be noticed that Step 3 requires the terms

$$\sum_{\ell} \hat{c}_{i,\ell}^m, \quad i = 1, 2, \dots, 8, \quad (3.18)$$

which can be computed from $\{\hat{c}_i^m : i = 1, 2, \dots, 8\}$. In fact, the quantity in (3.18) is equal to $\frac{1}{|V_x|} \int_{\Omega_x} \hat{c}_i^m dV$. So, one can store enough information from Step 1 to carry out Step 3 by saving 8 values per matrix block.

The quantity (3.16) can be moved from the right side to the left side in (3.14). So, Step 3 can be solved for the fracture concentration with the source terms being known values. The system (3.14) and (3.15) can be explained through the following algebraic representation:

$$\begin{bmatrix} A & B \\ -B^t & D \end{bmatrix} \begin{pmatrix} \mathbf{P}^m \\ C^m \end{pmatrix} = \begin{pmatrix} 0 \\ F^m \end{pmatrix} + \begin{pmatrix} 0 \\ -GC^m + GC^{m-1} \end{pmatrix}, \quad (3.19)$$

or, equivalently,

$$\begin{bmatrix} A & B \\ -B^t & D + G \end{bmatrix} \begin{pmatrix} \mathbf{P}^m \\ C^m \end{pmatrix} = \begin{pmatrix} 0 \\ F^m \end{pmatrix} + \begin{pmatrix} 0 \\ GC^{m-1} \end{pmatrix}, \quad (3.20)$$

for some G , where $\mathbf{P}^m = -D(\mathbf{V})\nabla C^m$ and $F^m = F^m(\bar{C}^{m-1}, \bar{c}^m, c^{m-1}, q_m^m)$.

In Steps 1, 2, and 4, the problems on the matrix blocks are independent of each other; they are naturally parallelizable. Each matrix block can usually be partitioned by 4 to 8 grid lines in each coordinate direction [13]. Therefore, its algebraic system can be easily solved by an iterative algorithm such as SOR. It should be noticed that SOR requires very few iterations for matrix blocks which are far from the main body of the contaminants. So the computational cost for the matrix blocks is relatively cheap. On the other hand, the problem in Step 3 is global and presumably expensive. Designing an efficient iterative algorithm for Step 3 is an interesting subject; the next section introduces DD methods for solving (3.9) and (3.14).

4. Mixed finite element DD method

The operator splitting scheme presented in Section 3 decouples the problem (2.1)-(2.7) into four steps of easy problems. Except for (3.9) and Step 3, the computation is local and naturally parallelizable. They require solving a problem of the form: Find p such that

$$\begin{aligned} \text{(a)} \quad & -\nabla \cdot (D(\mathbf{x})\nabla p) + \eta(\mathbf{x})p = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \\ \text{(b)} \quad & (D(\mathbf{x})\nabla p) \cdot \nu = 0, \quad \mathbf{x} \in \Gamma, \end{aligned} \quad (4.1)$$

where $0 \leq \eta_* \leq \eta(\mathbf{x}) \leq \eta^* < \infty$ and the diffusion-dispersion tensor is of the form

$$D = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \quad (4.2)$$

and is symmetric positive definite. We assume the following ellipticity:

$$D_* \mathbf{y}^T \mathbf{y} \leq \mathbf{y}^T D(\mathbf{x}) \mathbf{y} \leq D^* \mathbf{y}^T \mathbf{y}, \quad \forall \mathbf{y} \in \mathbb{R}^3, \quad \forall \mathbf{x} \in \Omega,$$

for some positive constants D_* and D^* .

In this section, we introduce DD iterative procedures for the mixed finite element approximate solution of (3.9) and (3.14). For a nonoverlapping DD method incorporating a RIBC on the subdomain interfaces, a special treatment of the RIBC is required. Note that RIBC imposes the continuity of both the pressure and the normal flux on the subdomain interfaces, while RTN0 admits discontinuities of the discrete pressure on the element interfaces. To avoid the technical difficulty, one can introduce Lagrange multipliers or modify the RIBC. We prefer to modify the RIBC in such a way that the decomposed problem solves the original discrete problem. However, the modification of the RIBC is not easy for the problems in which the diffusion-dispersion tensor is full. For such problems, we will employ a minimum overlapping DD method with a relaxation process introduced on the subdomain interfaces.

The nonoverlapping DD method incorporating a RIBC was first analyzed by Lions [37], where the convergence of the algorithm was proved in differential, rather than discrete, level. Hybrid mixed finite element DD methods introducing Lagrange multipliers were employed by Després [8] for solving the Helmholtz problem, and Douglas *et al.* [18] analyzed the convergence of the method for symmetric second-order partial differential equations. Recently, the author studied finite difference and finite element DD methods for solving complex-valued scalar waves without introducing Lagrange multipliers [32, 34, 35]. DD techniques for flows in porous media distinct from the one suggested here can be found in [7, 27, 30, 31].

4.1. A nonoverlapping DD method

Let $\{\Omega_j, j = 1, \dots, M\}$ be a nonoverlapping partition of Ω :

$$\overline{\Omega} = \bigcup_{j=1}^M \overline{\Omega}_j; \quad \Omega_j \cap \Omega_k = \emptyset, \quad j \neq k.$$

Assume also that each Ω_j is a (logically) cubic region. The method to be presented is equally applicable to the decompositions of the domain into either individual finite elements or larger subdomains (blocks of finite elements). Let

$$\Gamma_j = \Gamma \cap \partial\Omega_j, \quad \Gamma_{jk} = \Gamma_{kj} = \partial\Omega_j \cap \partial\Omega_k.$$

Let us consider a mixed formulation of the differential problem (4.1). Let

$$\mathbf{u} = -D\nabla p; \tag{4.3}$$

the problem (4.1) can be rewritten in the form of the first order system

$$\begin{aligned} \text{(a)} \quad & D^{-1}\mathbf{u} + \nabla p = 0, \\ \text{(b)} \quad & \nabla \cdot \mathbf{u} + \eta p = f, \quad \mathbf{x} \in \Omega, \\ \text{(c)} \quad & \mathbf{u} \cdot \nu = 0, \quad \mathbf{x} \in \Gamma. \end{aligned} \tag{4.4}$$

(It is implicitly imposed that $\int_{\Omega} p \, dx = 0$ or $p = 0$ at a point on the boundary.) Let $\mathbf{V} = \{\mathbf{v} \in \mathbf{H}(\text{div}; \Omega) : \mathbf{v} \cdot \nu = 0 \text{ on } \Gamma\} = \{\mathbf{v} \in (L^2(\Omega))^3 : \nabla \cdot \mathbf{v} \in L^2(\Omega); \mathbf{v} \cdot \nu = 0 \text{ on } \Gamma\}$, provided with the norm

$$\|\mathbf{v}\|_{\mathbf{H}(\text{div}, D)} = (\|\mathbf{v}\|_{0, D}^2 + \|\nabla \cdot \mathbf{v}\|_{0, D}^2)^{1/2},$$

and $W = L^2(\Omega)$. Then, the weak form of (4.4) is given by seeking a pair $\{\mathbf{u}, p\} \in \mathbf{V} \times W$ such that

$$\begin{aligned} \text{(a)} \quad & (D^{-1}\mathbf{u}, \mathbf{v})_{\Omega} - (\nabla \cdot \mathbf{v}, p)_{\Omega} = 0, \quad \mathbf{v} \in \mathbf{V}, \\ \text{(b)} \quad & (\nabla \cdot \mathbf{u}, w)_{\Omega} + (\eta p, w)_{\Omega} = (f, w)_{\Omega}, \quad w \in W. \end{aligned} \tag{4.5}$$

Let us consider the decomposition of (4.4) over $\{\Omega_j\}$: Find $\{\mathbf{u}_j, p_j\}, j = 1, \dots, M$, such that

$$\begin{aligned} \text{(a)} \quad & D^{-1}\mathbf{u}_j + \nabla p_j = 0, \quad \mathbf{x} \in \Omega_j, \\ \text{(b)} \quad & \nabla \cdot \mathbf{u}_j + \eta p_j = f, \quad \mathbf{x} \in \Omega_j, \\ \text{(c)} \quad & \mathbf{u} \cdot \nu = 0, \quad \mathbf{x} \in \Gamma_j, \\ \text{(d)} \quad & -\beta \mathbf{u}_j \cdot \nu_j + p_j = \beta \mathbf{u}_k \cdot \nu_k + p_k, \quad \mathbf{x} \in \Gamma_{jk}, \end{aligned} \tag{4.6}$$

where ν_j is the unit outer normal to Ω_j , and β is a positive function defined on the interfaces. The RIBC (4.6d) is equivalent to (and more convenient than) the consistency conditions [8, 18, 34, 37]

$$p_j = p_k, \quad \mathbf{u}_j \cdot \nu_j + \mathbf{u}_k \cdot \nu_k = 0, \quad \text{on } \Gamma_{jk}. \tag{4.7}$$

Now, move toward a new weak formulation by testing (4.6a) against a vector $\mathbf{v} \in \mathbf{V}_j = \mathbf{V}|_{\Omega_j}$:

$$(D^{-1}\mathbf{u}_j, \mathbf{v})_{\Omega_j} - (p_j, \nabla \cdot \mathbf{v})_{\Omega_j} + \langle p_j, \mathbf{v} \cdot \nu_j \rangle_{\partial\Omega_j} = 0, \quad \mathbf{v} \in \mathbf{V}_j. \tag{4.8}$$

Apply (4.6c) and (4.6d) to (4.8) to obtain (4.9a) below, and test (4.6b) against $w \in W_j = L^2(\Omega_j)$ to obtain the second equation in the system below. Thus, the weak mixed formulation of (4.1) over $\{\Omega_j\}$ is given by seeking $\{\mathbf{u}_j, p_j\} \in \mathbf{V}_j \times W_j, j = 1, \dots, M$, such that

$$\begin{aligned} \text{(a)} \quad & (D^{-1}\mathbf{u}_j, \mathbf{v})_{\Omega_j} - (p_j, \nabla \cdot \mathbf{v})_{\Omega_j} + \sum_k \langle \beta \mathbf{u}_j \cdot \nu_j, \mathbf{v} \cdot \nu_j \rangle_{\Gamma_{jk}} \\ & = - \sum_k \langle \beta \mathbf{u}_k \cdot \nu_k + p_k, \mathbf{v} \cdot \nu_j \rangle_{\Gamma_{jk}}, \quad \mathbf{v} \in \mathbf{V}_j, \\ \text{(b)} \quad & (\nabla \cdot \mathbf{u}_j, w)_{\Omega_j} + (\eta p_j, w)_{\Omega_j} = (f, w)_{\Omega_j}, \quad w \in W_j. \end{aligned} \tag{4.9}$$

There is a technical difficulty in (4.9a); the product of $\mathbf{u}_j \cdot \nu_j$ and $\mathbf{v} \cdot \nu_j$ is not necessarily integrable on Γ_{jk} . Also, it is not clear what the restriction of an $L^2(\Omega_j)$ -function to Γ_{jk} would be. So, we may view the system (4.9) as a heuristic motivation for the treatment of the discrete cases.

The objective of a DD iterative method is to localize the computations to smaller subdomains. It is feasible to solve on each Ω_j in parallel by evaluating the quantities in (4.9) related to Ω_j at the new iterate level and those related to neighboring subdomains Ω_k such that $\Gamma_{jk} \neq \emptyset$ at the old level. The iterative algorithm in the differential case would be as follows: Choose $\{\mathbf{u}_j^0, p_j^0\} \in \mathbf{V}_j \times W_j$, $j = 1, \dots, M$, arbitrarily; then recursively build the sequences $\{\mathbf{u}_j^n, p_j^n\} \in \mathbf{V}_j \times W_j$, $n \geq 1$, by solving

$$\begin{aligned}
 \text{(a)} \quad & (D^{-1}\mathbf{u}_j^n, \mathbf{v})_{\Omega_j} - (p_j^n, \nabla \cdot \mathbf{v})_{\Omega_j} + \sum_k \langle \beta \mathbf{u}_j^n \cdot \nu_j, \mathbf{v} \cdot \nu_j \rangle_{\Gamma_{jk}} \\
 & = - \sum_k \langle \beta \mathbf{u}_k^{n-1} \cdot \nu_k + p_k^{n-1}, \mathbf{v} \cdot \nu_j \rangle_{\Gamma_{jk}}, \quad \mathbf{v} \in \mathbf{V}_j, \\
 \text{(b)} \quad & (\nabla \cdot \mathbf{u}_j^n, w)_{\Omega_j} + (\eta p_j^n, w)_{\Omega_j} = (f, w)_{\Omega_j}, \quad w \in W_j.
 \end{aligned} \tag{4.10}$$

4.2. Mixed finite element procedure

Let

$$\mathbf{V}^h \times W^h = \mathbf{V}(\Omega, \mathcal{T}_h, 0) \times W(\Omega, \mathcal{T}_h, 0) \subset \mathbf{V} \times W$$

be the RTN0 [38, 40] associated with a quasi-regular partition \mathcal{T}_h of Ω into simplices (tetrahedra, cubes, prisms) not crossing the interfaces in the partition $\{\Omega_j\}$ of Ω and of diameters not greater than h . Note that the requirement $\mathbf{v} \in \mathbf{H}(\text{div}, \Omega)$ is equivalent to asking for continuity in the normal component across the faces of the elements [23]. We define the corresponding subspaces

$$\mathbf{V}_j^h = \{\mathbf{v}|_{\Omega_j} : \mathbf{v} \in \mathbf{V}^h\}, \quad W_j^h = \{w|_{\Omega_j} : w \in W^h\}, \quad M = 1, \dots, M.$$

The mixed finite element approximation of (4.1) is given by restricting (4.5) to the space $\mathbf{V}^h \times W^h$: Find $\{\mathbf{u}^h, p^h\} \in \mathbf{V}^h \times W^h$ such that

$$\begin{aligned}
 \text{(a)} \quad & (D^{-1}\mathbf{u}^h, \mathbf{v})_{\Omega} - (\nabla \cdot \mathbf{v}, p^h)_{\Omega} = 0, \quad \mathbf{v} \in \mathbf{V}^h, \\
 \text{(b)} \quad & (\nabla \cdot \mathbf{u}^h, w)_{\Omega} + (\eta p^h, w)_{\Omega} = (f, w)_{\Omega}, \quad w \in W^h;
 \end{aligned} \tag{4.11}$$

for the existence and uniqueness of the approximate solution and convergence properties of the method, we refer to [23, 40]. Then, the decomposition of (4.11) over $\{\Omega_j\}$ could be defined by restricting (4.9) to the space $\mathbf{V}_j^h \times W_j^h$. However, one can easily check that the restriction may not solve the original discrete problem (4.11) unless the discrete solution p^h is constant over the domain Ω . To avoid this difficulty, we proceed to modify the RIBC. In this section, we assume that the diffusion-dispersion tensor is diagonal, $D = aI$, and that each element is a cube-shaped region of the uniform mesh size $\mathbf{h} = (h_x, h_y, h_z)$. For the case of general tensors and meshes, see §4.3.

Let $\alpha(x) = 1/a(x)$ and consider the following problem: Find $\{\mathbf{u}_j^h, p_j^h\} \in \mathbf{V}_j^h \times W_j^h$,

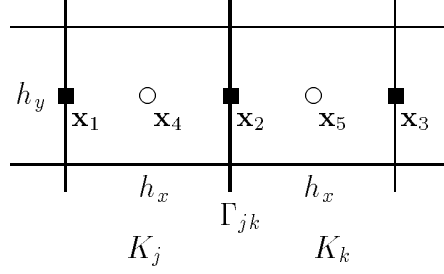


Figure 4: The quadrilateral RTN0 elements.

$j = 1, \dots, M$, such that

$$\begin{aligned}
 \text{(a)} \quad & (\alpha \mathbf{u}_j^h, \mathbf{v})_{\Omega_j} - (p_j^h, \nabla \cdot \mathbf{v})_{\Omega_j} + \sum_k \left\langle \frac{1}{2} (p_j^h + \hat{p}_j^h), \mathbf{v} \cdot \nu_j \right\rangle_{\Gamma_{jk}} \\
 & = 0, \quad \mathbf{v} \in \mathbf{V}_j^h, \\
 \text{(b)} \quad & \hat{p}_j^h = p_k^h; \quad \mathbf{u}_j^h \cdot \nu_j + \mathbf{u}_k^h \cdot \nu_k = 0, \quad \mathbf{x} \in \Gamma_{jk}, \\
 \text{(c)} \quad & (\nabla \cdot \mathbf{u}_j^h, w)_{\Omega_j} + (\eta p_j^h, w)_{\Omega_j} = (f, w)_{\Omega_j}, \quad w \in W_j^h.
 \end{aligned} \tag{4.12}$$

(Here we apply an exterior bordering of the subdomains Ω_j .) It is not difficult to verify that problem (4.12) is *equivalent* to the original discrete problem (4.11), i.e., their solutions are identical, when a mass-lumping quadrature is employed. The terms provided by the variational formulation are integrated by the quadrature rule in which the quadrature points coincide with the degrees of freedom (the Gauss-Lobatto points for \mathbf{u}^h and the Gauss points for p^h).

We will verify the equivalence. It suffices to check it for the basis functions (in \mathbf{V}^h) the support of which is not included in one subdomain. For a simple presentation, we will consider the 2D case; see Figure 4. Let $\mathbf{u} = (u^{(1)}, u^{(2)})$. The degrees of freedom for p^h (resp., $u^{(1),h}$) are denoted by \circ (resp., \blacksquare). Let $\Theta = (\theta, 0) \in \mathbf{V}^h$ be the basis function such that

$$\theta = \begin{cases} 1, & \mathbf{x} \in \partial K_j \cap \partial K_k, \\ 0, & \mathbf{x} = \mathbf{x}_i, \quad i = 1, 3, \end{cases} \tag{4.13}$$

where K_j and K_k are finite elements. When $\mathbf{v} = \Theta$, the equation (4.11a) becomes

$$-h_y p_4^h + \alpha_2 h_x h_y u_2^{(1),h} + h_y p_5^h = 0; \tag{4.14}$$

while the equations (4.12a)-(4.12b) yield

$$-\frac{h_y}{2} p_4^h + \alpha_2 \frac{h_x h_y}{2} u_2^{(1),h} + \frac{h_y}{2} p_5^h = 0, \tag{4.15}$$

where the numerals in the subscripts denote degrees of freedom. In a similar way the result can be verified on the interfaces orthogonal to the y -axis, which implies that (4.12) solves (4.11).

Now, we can define the iterative algorithm as follows: Select $\{\mathbf{u}_j^{h,0}, p_j^{h,0}\} \in \mathbf{V}_j^h \times W_j^h$, $j = 1, \dots, M$, arbitrarily, then recursively build the sequence $\{\mathbf{u}_j^{h,n}, p_j^{h,n}\} \in \mathbf{V}_j^h \times W_j^h$, $n \geq 1$, by solving

$$\begin{aligned}
\text{(a)} \quad & (\alpha \mathbf{u}_j^{h,n}, \mathbf{v})_{\Omega_j} - (p_j^{h,n}, \nabla \cdot \mathbf{v})_{\Omega_j} + \sum_k \left\langle \frac{1}{2} (p_j^{h,n} + \hat{p}_j^{h,n}), \mathbf{v} \cdot \nu_j \right\rangle_{\Gamma_{jk}} \\
& = 0, \quad \mathbf{v} \in \mathbf{V}_j^h, \\
\text{(b)} \quad & -\beta \mathbf{u}_j^{h,n} \cdot \nu_j + \hat{p}_j^{h,n} = \beta \mathbf{u}_k^{h,n-1} \cdot \nu_k + p_k^{h,n-1}, \quad \text{on } \Gamma_{jk}, \\
\text{(c)} \quad & (\nabla \cdot \mathbf{u}_j^{h,n}, w)_{\Omega_j} + (\eta p_j^{h,n}, w)_{\Omega_j} = (f, w)_{\Omega_j}, \quad w \in W_j^h,
\end{aligned} \tag{4.16}$$

or, equivalently,

$$\begin{aligned}
\text{(a)} \quad & (\alpha \mathbf{u}_j^{h,n}, \mathbf{v})_{\Omega_j} - (p_j^{h,n}, \nabla \cdot \mathbf{v})_{\Omega_j} + \sum_k \left\langle \frac{1}{2} (\beta \mathbf{u}_j^{h,n} \cdot \nu_j + p_j^{h,n}), \mathbf{v} \cdot \nu_j \right\rangle_{\Gamma_{jk}} \\
& = - \sum_k \left\langle \frac{1}{2} (\beta \mathbf{u}_k^{h,n-1} \cdot \nu_k + p_k^{h,n-1}), \mathbf{v} \cdot \nu_j \right\rangle_{\Gamma_{jk}}, \quad \mathbf{v} \in \mathbf{V}_j^h, \\
\text{(b)} \quad & (\nabla \cdot \mathbf{u}_j^{h,n}, w)_{\Omega_j} + (\eta p_j^{h,n}, w)_{\Omega_j} = (f, w)_{\Omega_j}. \quad w \in W_j^h,
\end{aligned} \tag{4.17}$$

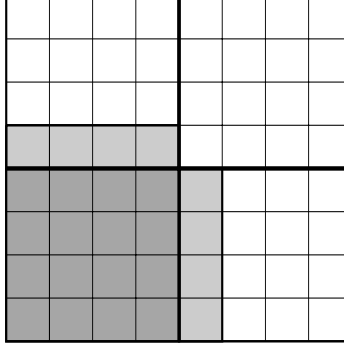
Here, $\beta > 0$ is a relaxation parameter introduced to accelerate the convergence and to make the subproblems well-posed. Note that, on the subdomain interfaces, the modified RIBC (4.16b) imposes continuity of the normal flux only.

It should be noted that our matching condition (4.16b) depends on the numerical quadrature. It is not easy to find the matching condition when we consider higher-order accurate quadrature rules or unstructured meshes. For such cases, we can employ overlapping techniques (expansion of nonoverlapping subdomains). Then, the resulting DD algorithm can be considered as a generalized Schwarz method [43]. We will discuss an algorithm of this type in the following section.

4.3. A minimum-overlapping DD method

In this section, we consider a DD method for solving (4.1)-(4.2) by RTN0. We assume that the mesh is quadrilateral and quasi-regular, but not necessarily uniform or structured. Since finding a matching condition for the RIBC is not a simple task for such a general case, as mentioned in the previous section, we will consider a DD method with a minimum overlap.

Let $\tilde{\Omega}_j$ be an expansion of Ω_j such that $\tilde{\Omega}_j$ includes Ω_j and the finite elements

Figure 5: The subdomain expansion $\tilde{\Omega}_j$ (shaded).

having a face common with $\partial\Omega_j$, and set $\tilde{\Gamma}_{jk} = \partial\tilde{\Omega}_j \cap \bar{\Omega}_k$. (See Figure 5.) Let

$$\begin{aligned} \mathbf{V}_j^h &= \mathbf{V}^h|_{\Omega_j}, & \tilde{\mathbf{V}}_j^h &= \mathbf{V}^h|_{\tilde{\Omega}_j}, \\ \mathbf{V}_{0,j}^h &= \{\mathbf{v} \in \mathbf{V}_j^h : \mathbf{v}|_{\partial\Omega_j} = 0\}, & \tilde{\mathbf{V}}_{0,j}^h &= \{\mathbf{v} \in \tilde{\mathbf{V}}_j^h : \mathbf{v}|_{\partial\tilde{\Omega}_j} = 0\}, \\ W_j^h &= W^h|_{\Omega_j}, & \tilde{W}_j^h &= W^h|_{\tilde{\Omega}_j}, \end{aligned}$$

and

$$\tilde{\mathbf{V}}_{I,j}^h = \{\mathbf{v} \in \tilde{\mathbf{V}}_j^h : \mathbf{v}|_{\Omega^h \setminus \cup \Gamma_{jk}} = 0\},$$

where Ω^h is the set of nodal points in $\bar{\Omega}$. For $\{\tilde{\mathbf{v}}_j^h, \tilde{w}_j^h\} \in \tilde{\mathbf{V}}_j^h \times \tilde{W}_j^h$, using the same subscripts as in the definitions of spaces, we set

$$\begin{aligned} \mathbf{v}_j^h &= \tilde{\mathbf{v}}_j^h|_{\mathbf{V}_j^h}, & \mathbf{v}_{0,j}^h &= \tilde{\mathbf{v}}_j^h|_{\mathbf{V}_{0,j}^h}, \\ \tilde{\mathbf{v}}_{0,j}^h &= \tilde{\mathbf{v}}_j^h|_{\tilde{\mathbf{V}}_{0,j}^h}, & w_j^h &= \tilde{w}_j^h|_{W_j^h}. \end{aligned} \quad (4.18)$$

Also, let $\hat{\mathbf{v}}_j^h$, $\tilde{\mathbf{v}}_{I,j}^h$, and \hat{w}_j^h be defined by the following equalities:

$$\tilde{\mathbf{v}}_j^h = \tilde{\mathbf{v}}_{0,j}^h + \hat{\mathbf{v}}_j^h = \mathbf{v}_{0,j}^h + \tilde{\mathbf{v}}_{I,j}^h + \hat{\mathbf{v}}_j^h, \quad \tilde{w}_j^h = w_j^h + \hat{w}_j^h. \quad (4.19)$$

Consider the following DD method: Find $\{\tilde{\mathbf{u}}_j^h, \tilde{p}_j^h\} \in \tilde{\mathbf{V}}_j^h \times \tilde{W}_j^h$, $j = 1, \dots, M$, such that

$$\begin{aligned} (a) \quad & (D^{-1}\tilde{\mathbf{u}}_j^h, \tilde{\mathbf{v}})_{\tilde{\Omega}_j} - (\tilde{p}_j^h, \nabla \cdot \tilde{\mathbf{v}})_{\tilde{\Omega}_j} = 0, \quad \tilde{\mathbf{v}} \in \tilde{\mathbf{V}}_{0,j}^h, \\ (b) \quad & (\nabla \cdot \tilde{\mathbf{u}}_j^h, w)_{\Omega_j} + (\eta\tilde{p}_j^h, w)_{\Omega_j} = (f, w)_{\Omega_j}, \quad w \in W_j^h, \\ (c) \quad & \tilde{\mathbf{u}}_j^h = \mathbf{u}_k^h \text{ on } \tilde{\Gamma}_{jk}; \quad \tilde{p}_j^h = p_k^h \text{ on } \tilde{\Omega}_j \cap \Omega_k. \end{aligned} \quad (4.20)$$

It is easy to verify that the subproblems in (4.20) are well-posed and that the problem (4.20) solves the original discrete problem (4.11). This is why overlapping DD

methods are easy to design and describe. Note that

$$\begin{aligned}
(D^{-1}\tilde{\mathbf{u}}_j^h, \tilde{\mathbf{v}})_{\tilde{\Omega}_j} &= (D^{-1}\tilde{\mathbf{u}}_{0,j}^h, \tilde{\mathbf{v}})_{\tilde{\Omega}_j} + (D^{-1}\hat{\mathbf{u}}_j^h, \tilde{\mathbf{v}})_{\tilde{\Omega}_j} \\
&= (D^{-1}\tilde{\mathbf{u}}_{0,j}^h, \tilde{\mathbf{v}})_{\tilde{\Omega}_j} + \sum (D^{-1}\hat{\mathbf{u}}_j^h, \tilde{\mathbf{v}})_{\Omega_k}, \\
(\hat{p}_j^h, \nabla \cdot \tilde{\mathbf{v}})_{\tilde{\Omega}_j} &= (p_j^h, \nabla \cdot \tilde{\mathbf{v}})_{\tilde{\Omega}_j} + (\hat{p}_j^h, \nabla \cdot \tilde{\mathbf{v}})_{\tilde{\Omega}_j} \\
&= (p_j^h, \nabla \cdot \mathbf{v})_{\Omega_j} + \sum_k (\hat{p}_j^h, \nabla \cdot \tilde{\mathbf{v}})_{\Omega_k}.
\end{aligned}$$

So, (4.20) can be rewritten as follows: Find $\{\tilde{\mathbf{u}}_j^h, \hat{p}_j^h\} \in \tilde{\mathbf{V}}_j^h \times \tilde{W}_j^h$, $j = 1, \dots, M$, such that

$$\begin{aligned}
\text{(a)} \quad & (D^{-1}\tilde{\mathbf{u}}_{0,j}^h, \tilde{\mathbf{v}})_{\tilde{\Omega}_j} - (p_j^h, \nabla \cdot \mathbf{v})_{\Omega_j} + \sum_k \left\{ (D^{-1}\hat{\mathbf{u}}_j^h, \tilde{\mathbf{v}})_{\Omega_k} - (\hat{p}_j^h, \nabla \cdot \tilde{\mathbf{v}})_{\Omega_k} \right\} \\
& = 0, \quad \tilde{\mathbf{v}} \in \tilde{\mathbf{V}}_{0,j}^h, \\
\text{(b)} \quad & (\nabla \cdot \mathbf{u}_j^h, w)_{\Omega_j} + (\eta p_j^h, w)_{\Omega_j} = (f, w)_{\Omega_j}, \quad w \in W_j^h, \\
\text{(c)} \quad & \hat{\mathbf{u}}_j^h = \mathbf{u}_{0,k}^h, \quad \hat{p}_j^h = p_k^h, \quad \text{on } \tilde{\Omega}_j \cap \Omega_k.
\end{aligned} \tag{4.21}$$

Now, let us define the mixed finite element DD iterative procedure for (4.21) as follows: Choose $\{\tilde{\mathbf{u}}_j^{h,0}, \hat{p}_j^{h,0}\} \in \tilde{\mathbf{V}}_j^h \times \tilde{W}_j^h$, $j = 1, \dots, M$, and recursively build the sequence $\{\tilde{\mathbf{u}}_j^{h,n}, \hat{p}_j^{h,n}\} \in \tilde{\mathbf{V}}_j^h \times \tilde{W}_j^h$, $n \geq 1$, by solving

$$\begin{aligned}
\text{(a)} \quad & (D^{-1}\tilde{\mathbf{u}}_{0,j}^{h,n}, \tilde{\mathbf{v}})_{\tilde{\Omega}_j} - (p_j^{h,n}, \nabla \cdot \mathbf{v})_{\Omega_j} \\
& + \sum_k \left\{ (D^{-1}\hat{\mathbf{u}}_j^{h,n}, \tilde{\mathbf{v}})_{\Omega_k} - (\hat{p}_j^{h,n}, \nabla \cdot \tilde{\mathbf{v}})_{\Omega_k} \right\} = 0, \quad \tilde{\mathbf{v}} \in \tilde{\mathbf{V}}_{0,j}^h, \\
\text{(b)} \quad & \sum_k \left\{ -\langle \beta \mathbf{u}_j^{h,n} \cdot \nu_j, \mathbf{v} \cdot \nu_j \rangle_{\Gamma_{jk}} + (D^{-1}\hat{\mathbf{u}}_j^{h,n}, \tilde{\mathbf{v}}_{I,j})_{\Omega_k} - (\hat{p}_j^{h,n}, \nabla \cdot \tilde{\mathbf{v}}_{I,j})_{\Omega_k} \right\} \\
& = \sum_k \left\{ \langle \beta \mathbf{u}_k^{h,n-1} \cdot \nu_k, \mathbf{v} \cdot \nu_j \rangle_{\Gamma_{jk}} \right. \\
& \quad \left. + (D^{-1}\mathbf{u}_{0,k}^{h,n-1}, \tilde{\mathbf{v}}_{I,j})_{\Omega_k} - (p_k^{h,n-1}, \nabla \cdot \tilde{\mathbf{v}}_{I,j})_{\Omega_k} \right\}, \quad \tilde{\mathbf{v}} \in \tilde{\mathbf{V}}_{0,j}^h, \\
\text{(c)} \quad & (\nabla \cdot \mathbf{u}_j^{h,n}, w)_{\Omega_j} + (\eta p_j^{h,n}, w)_{\Omega_j} = (f, w)_{\Omega_j}, \quad w \in W_j^h,
\end{aligned} \tag{4.22}$$

where $\beta \geq 0$ is a relaxation parameter and $\mathbf{v} = \tilde{\mathbf{v}}|_{\Omega_j}$. Equations (4.22a)-(4.22b) read

$$\begin{aligned}
& (D^{-1}\tilde{\mathbf{u}}_{0,j}^{h,n}, \tilde{\mathbf{v}})_{\tilde{\Omega}_j} - (p_j^{h,n}, \nabla \cdot \mathbf{v})_{\Omega_j} + \sum_k \langle \beta \mathbf{u}_j^{h,n} \cdot \nu_j, \mathbf{v} \cdot \nu_j \rangle_{\Gamma_{jk}} \\
& = - \sum_k \langle \beta \mathbf{u}_k^{h,n-1} \cdot \nu_k, \mathbf{v} \cdot \nu_j \rangle_{\Gamma_{jk}} \\
& \quad - \sum_k \left\{ (D^{-1}\mathbf{u}_{0,k}^{h,n-1}, \tilde{\mathbf{v}})_{\Omega_k} - (p_k^{h,n-1}, \nabla \cdot \tilde{\mathbf{v}})_{\Omega_k} \right\}, \quad \tilde{\mathbf{v}} \in \tilde{\mathbf{V}}_{0,j}^h.
\end{aligned}$$

Algorithm (4.22) overlaps the unknowns only on the interfaces of the original nonoverlapping subdomains, Γ_{jk} . The reader should notice that if $\beta = 0$ the algorithm can be considered as a variant of the standard Schwarz algorithm (Dirichlet data communication) and it is often accelerated by a CG-type exterior iterative algorithm and coarse grid corrections.

4.4. Convergence analysis

The convergence of algorithms (4.17) and (4.22) is proved in [33]. The analysis utilizes the energy norm estimate introduced in [18]. Here we present the result.

Theorem 4.1. *Let $\eta_* > 0$ and β be chosen as*

$$\beta = \sigma \frac{C_1}{4\eta_* h}, \quad \sigma = 1 + \left(1 + \frac{16\eta_* h^2}{C_2 D^*}\right)^{1/2}, \quad (4.23)$$

for some positive C_1 and C_2 independent of h , D , and η . Then the spectral radius of the iteration matrix of (4.17) and (4.22) is minimized and bounded as

$$\rho(\mathcal{A}) \leq 1 - C_3 \frac{\eta_* h^2}{\eta_* h^2 + D^*}, \quad (4.24)$$

for some $C_3 > 0$ independent of h , D , and η .

Remark. For (3.10), i.e., when $\eta \equiv 0$, algorithm (4.17) has not been proved to converge. However, (3.10) can be efficiently solved by a multigrid method of which the coarse grid problem is slightly damped by $\eta > 0$ and solved by the nonoverlapping DD algorithm (4.17) [36].

Remark. In geophysical applications, the magnitude of the diffusion-dispersion tensor is small, i.e., D^* is on the order of 10^{-2} to 10^{-8} , depending on the medium and the length scale [29]. One often chooses $\Delta t \equiv 1/\eta = \mathcal{O}(h)$. Algorithm (4.22) is expected to converge fast, even though acceleration techniques such as the CG iteration and the coarse subspace correction are not incorporated.

5. Numerical experiments

Algorithms (4.17) and (4.22) are implemented to solve (3.9) and (3.14) by RTN0. For the time stepping of the concentration equation, no coarse grid solver needs to be applied due to the possibility of choosing a quite accurate initial guess and small diffusion-dispersion tensor; see e.g. [10, 11].

5.1. Pressure equation

Let $\Omega = (0, 1)^3$. In this section, we present numerical results to show efficiency of the algorithm (4.17) for solving the following form of pressure equation:

$$\begin{aligned} \mathbf{v} &= -a(\mathbf{x})\nabla p, & \mathbf{x} \in \Omega, \\ \nabla \cdot \mathbf{v} &= f(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \mathbf{v} \cdot \boldsymbol{\nu} &= g(\mathbf{x}), & \mathbf{x} \in \Gamma. \end{aligned} \quad (5.1)$$

The diffusion coefficient a was chosen as

$$a(x, y, z) = \frac{1}{1 + 10(4x^2 + 2y^2 + z^2)}. \quad (5.2)$$

$1/h$	n	r_∞^n	e_1^n	e_2^n	e_3^n
32	121	1.1e-2	9.9e-3	9.4e-3	4.2e-3
64	127	2.7e-3	2.6e-3	2.3e-3	1.0e-3

Table 1: The DD method for the pressure equation. The domain is decomposed into pole-shaped subdomains.

The functions f and g were selected so that the true solution

$$p(x, y, z) = e^x \sin(5x) \cos(2\pi y) \cos(4\pi z). \quad (5.3)$$

Let the mesh size $\mathbf{h} = (h_x, h_y, h_z) = (1/N_x, 1/N_y, 1/N_z)$ and the domain be decomposed into $M_x \times M_y \times M_z$ subdomains. We choose $1/h = N_x = N_y = N_z$ and $M_x = 1$, $M_y = N_y$, and $M_x = N_z$, pole-shaped subdomains in the x -direction.

The error was estimated by the relative maximum norms r_∞^n and e_i^n , $i = 1, 2, 3$, and the stopping criterion was $s_\infty^n \leq 10^{-5}$, where

$$r_\infty^n = \frac{\|P^n - p\|_{L^\infty(\Omega)}}{\|p\|_{L^\infty(\Omega)}}, \quad e_i^n = \frac{\|U_i^n - u_i\|_{L^\infty(\Omega)}}{\|u_i\|_{L^\infty(\Omega)}}, \quad s_\infty^n = \frac{\|P^n - P^{n-1}\|_{L^\infty(\Omega)}}{\|P^n\|_{L^\infty(\Omega)}}.$$

Here P^n and $U^n = (U_1^n, U_2^n, U_3^n)$ are the approximate solutions of p and $u = (u_1, u_2, u_3)$ in the n -th iteration, respectively. Zero initial values, $P^0 = U_i^0 \equiv 0$, were assumed. Each processor was assigned to solve $M_y/8 \times M_z/8$ subdomains.

Table 1 contains the numerical results for the grid size $h = h_x = h_y = h_z$. The number of iterations is denoted by n . Since at least $(M_y + M_z - 1)$ iterations are required for a complete data communication, the convergence results (in particular, for the case $1/h = 64$) seem to be quite efficient. In the above table, we observe second-order convergence in the L^∞ -norm for both velocity and pressure approximations.

5.2. Contaminant transport

Now, we consider underwater contamination by toxic liquids such as PCBs, TCE, and various petroleum products. Such liquids can be leaking from toxic waste sites [28, and references therein]. It is of interest to track the movement of the contamination. When the water velocity is low, the contaminant transport is often gravity-dominated because of the higher density of toxic materials. The resulting problem in this case is (relatively) easier than convection-dominated flow problems. We assume that the toxic material is perfectly miscible with water so that the gravity effect can be ignored. Also we assume that the velocity field is independent of time.

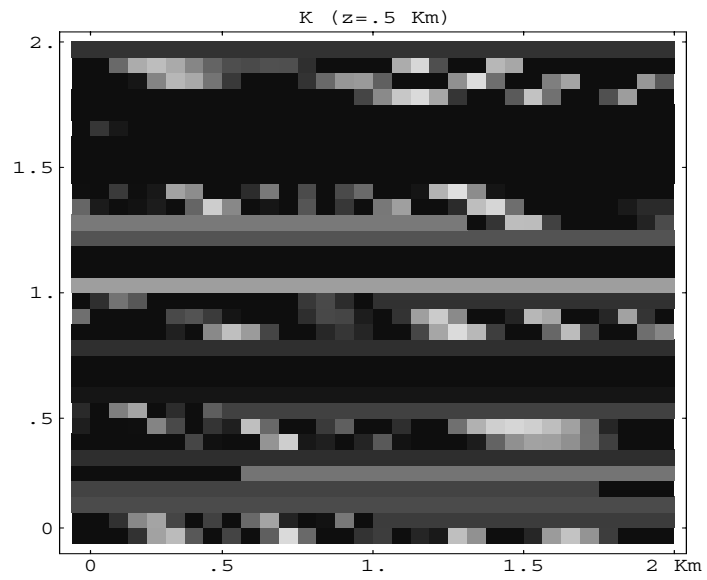
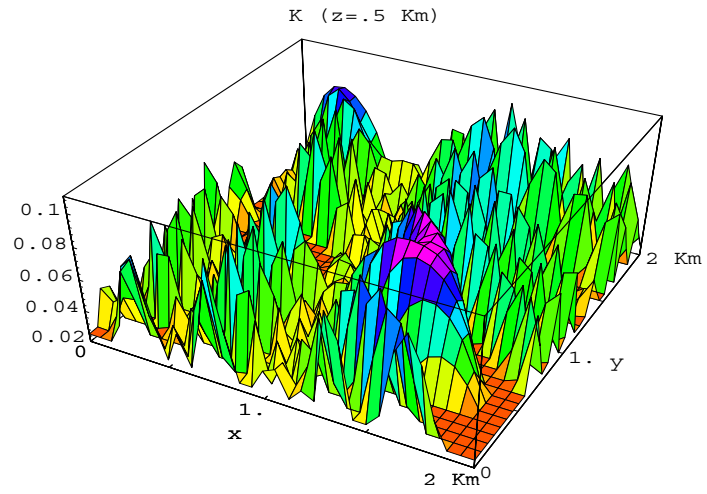


Figure 6: The fracture permeability field K on $Z_{0.5}$.

This section reports numerical results for the system of equations (3.3)-(3.7). The operator splitting scheme presented in §3.2 is employed for an efficient time-stepping. The algorithms (4.17) and (4.22) were applied to (3.9) and (3.14), respectively. Steps 1, 2, and 4 are solved by the point SOR. (Its relaxation parameter is chosen so that it is optimal for the problem with locally averaged coefficients.) The following numerical data were considered:

Domain	$\Omega = (0, 2\text{Km}) \times (0, 2\text{Km}) \times (0, 1\text{Km})$
Viscosity	$\mu = 1 \text{ cP}$
Porosity	$\phi = 0.2, \quad \Phi = 0.01$
Permeability	$k(\mathbf{x}) = K(\mathbf{x})/40$
Flow rate on boundary	$\mathbf{V} \cdot \nu_x _{x=0} = -\mathbf{V} \cdot \nu_x _{x=2} = -0.1 \text{ Km/yr};$ $\mathbf{V} \cdot \nu_y = \mathbf{V} \cdot \nu_z = 0$
Initial concentration	$c^0 = C^0 \equiv 0$
Injected concentration	$q_f = \begin{cases} \delta(\mathbf{x} - \mathbf{x}_s), & \text{if } t \leq 1 \text{ mth}, \\ 0, & \text{if } t > 1 \text{ mth}, \end{cases}$ where $\mathbf{x}_s = (0.5, 1.0, 0.5)$
Diff-Disp coefficients	$d_m = 1.0\text{e-}7; \quad d_\ell = 1.0\text{e-}5; \quad d_t = 1.0\text{e-}6$

The fracture permeability $K(\mathbf{x})$ is given as random values so that $0.005 \leq K(\mathbf{x}) \leq 0.2$; see Figure 6, which depicts the 3D-plot (top) and the density-plot (bottom) for the permeability field K on the cross section

$$Z_{0.5} := \{(x, y, z) \in \Omega : z = 0.5\}.$$

A review of data on diffusion-dispersion coefficients in realistic media can be found in [29]. The simulation parameters were chosen as follows:

Number of elements	$N_x \times N_y \times N_z = 32 \times 32 \times 16$
Size of Ω_x	$25 \text{ m} \times 25 \text{ m} \times 12.5 \text{ m}$
Elements in Ω_x	$m_x \times m_y \times m_z = 5 \times 5 \times 4$
Time interval	$J = [0, 10 \text{ yr}]$
Timestep	$\Delta t^m = 0.05 \text{ yr}$ (fixed)
Stopping criterion	$s_\infty^n \leq 10^{-5}$ (DD and SOR)

The resulting algebraic system of the above problem has more than 3 million unknowns in each time step. The computation is carried out in an IBM Power PC-AIX 4.1. The number of iterations and elapsed time are as follows:

<u>The pressure equation</u>	
Number of iteration	$n = 145$
Elapsed time	CPU = 7.1 seconds
<u>The concentration equation</u>	
Number of timesteps	$M = 200$
Max-DD-iter on a timestep	$n = 12$
Min-DD-iter on a timestep	$n = 2$
Total DD-iter	$n = 777$
Total SOR-iter for Step 2	$n = 676$
Total SOR-iter for Step 4	$n = 681$
Elapsed time	CPU = 2077.7 sec \simeq 34.6 min

From the above result, we can see that the average number of iterations per time step is about 4 for the pole-shaped DD method and 3.4 for SOR iterations. In most timesteps, the algorithm required three DD iterations for Step 3 and three SOR iterations for each of Steps 2 and 4. The costly timesteps are the very early ones and the several timesteps after the injection of contamination has stopped. The 10 year simulation for contaminant transport, which consists of 200 time steps with more than 3 million unknowns at each step, was performed within 35 minutes on an old personal computer.

We will report the numerical results obtained from the data presented in this section to show the features of problem (2.1)-(2.7) and efficiency of the algorithms (3.11)-(3.17), (4.17), and (4.22). For a technical reason, we report only the results on the cross section $Z_{0.5}$. The concentration of matrix blocks is averaged on each block. In the averaging process, only the interior nodal values of the matrix block are considered. (Due to the continuity condition (3.6), the nodal values on the matrix block boundaries vary rapidly with the fracture concentration; if the boundary nodal values are included, the resulting quantity may not represent the average of the matrix concentration.)

In Figures 7-10, the fracture concentration C (top) and the matrix concentration c (bottom) are depicted, when $t = 2, 4, 6$ and 10 years, respectively. From the figures, one can see the *tail* which follows the main body of the fracture contaminants and the *tongue* which invokes the contaminants in matrix blocks. The contaminants in fractures move fast and some of them are smeared into matrix blocks in the downstream. When concentration in the fractures are lower, the mixtures in matrix blocks would move out to the fractures. These are the main features of the dual-porosity model (2.1)-(2.7).

Figures 11 and 12 present the concentration on the center subdomain (i.e., the (16,8)-th subdomain on the yz -plane). The thick lines denote the concentration profile of the fractures and the thin lines are for that of matrix blocks. Note that the main

body of the fracture contaminant in Figure 12, when $t = 10$ years, is near $x = 1.35$. (It was expected to be near $x = 1.5$ for single-porosity models, since the average (fracture) velocity was given to be 0.1Km/year.) The dual-porosity model (2.1)-(2.7) can be characterized by such delay phenomena and hopefully express the physics of contaminant transport more accurately. It should be also noticed that the matrix-fracture fluid transfer is modeled by imposing the continuity condition between the fracture concentration and the matrix concentration on the matrix boundaries and by including the source/sink term into the fracture equation. The tongue on the matrix block concentration is caused by the continuity condition on the matrix boundaries, while the tail of the fracture concentration is due to the matrix source.

In Figures 13 and 14, we report the change of the concentration on the matrix block attached in the (12, 16, 8)-th fracture element. (The element contains the point $\mathbf{x}_o = (0.75, 1, 0.5)$ on one of its faces; we call the matrix block $\Omega_{\mathbf{x}_o}$.) When $t = 10$, the matrix block concentration c has a maximum value on the node near the boundary (but not on the boundary). This is not surprising since the values on $\partial\Omega_{\mathbf{x}_o}$ would vary rapidly with the fracture values.

In Figure 15, we depict change of the concentrations C and c on the point \mathbf{x}_o and the matrix block $\Omega_{\mathbf{x}_o}$, respectively. The matrix concentration grows almost linearly during the time interval: 0 to 10 years.

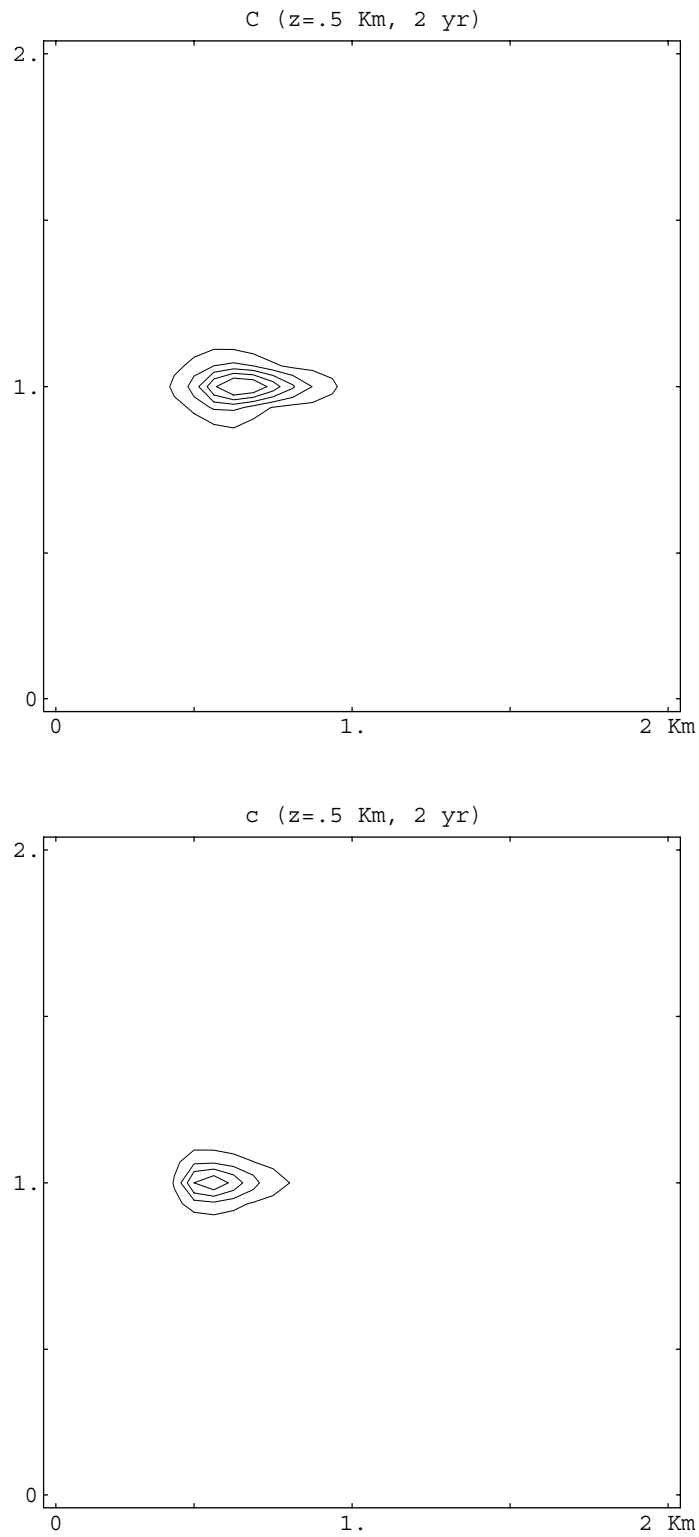


Figure 7: The concentrations for the fractures (top) and matrix blocks (bottom) on $Z_{0.5}$ when $t = 2$ yrs.

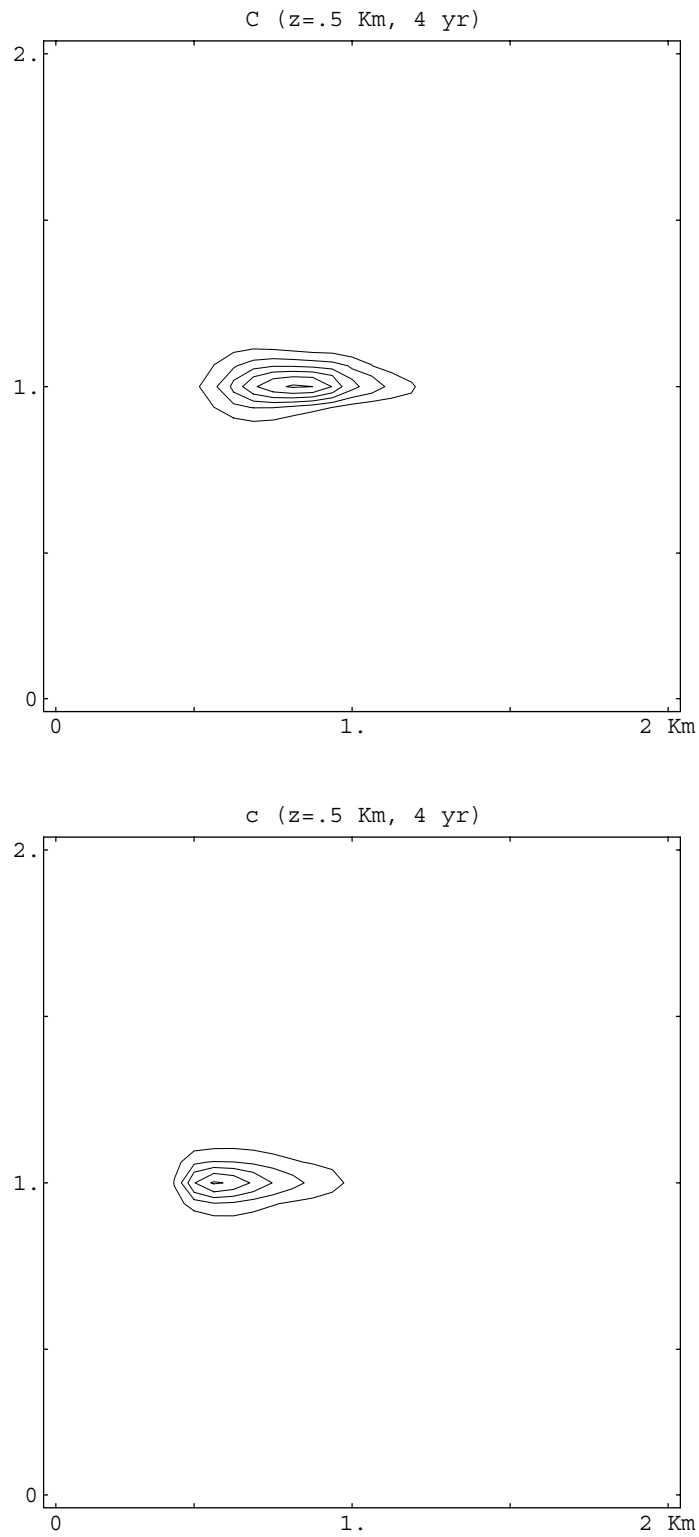


Figure 8: The concentrations for the fractures (top) and matrix blocks (bottom) on $Z_{0.5}$ when $t = 4$ yrs.

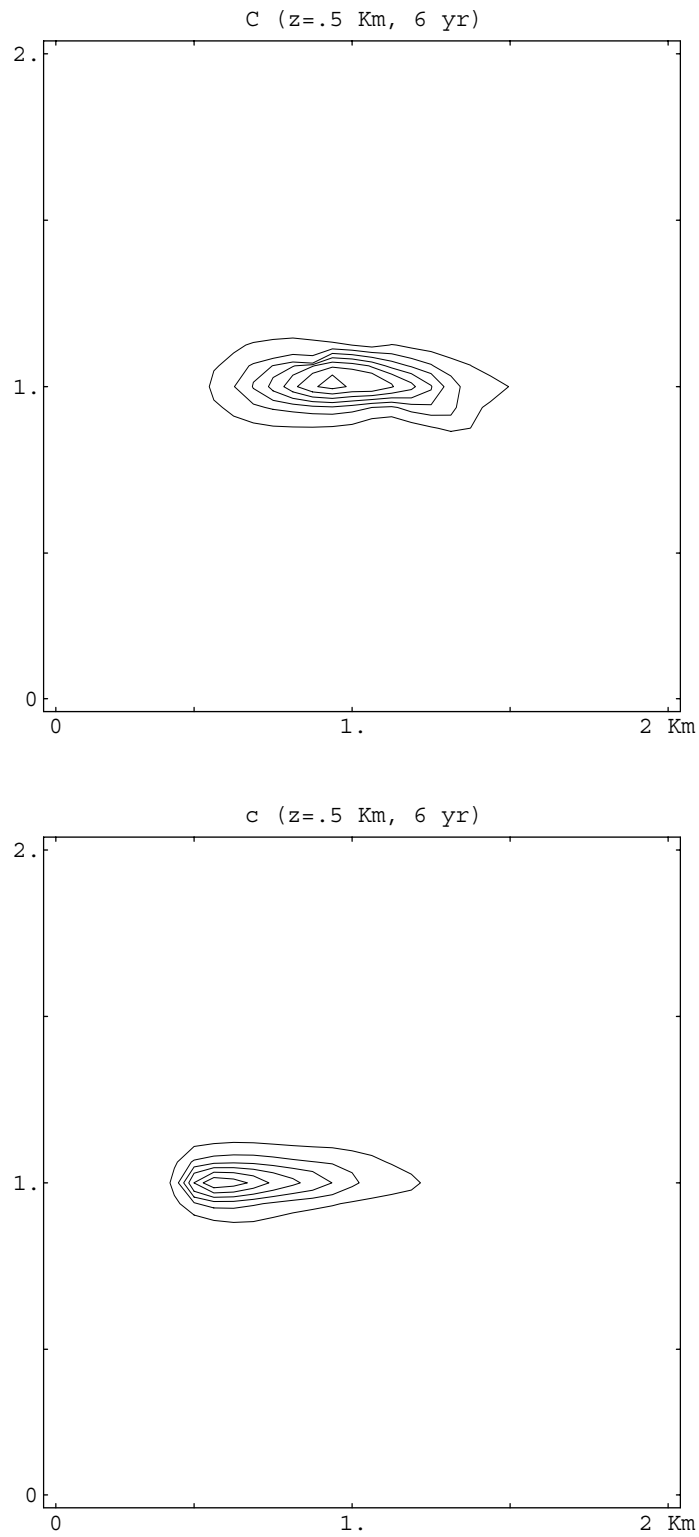


Figure 9: The concentrations for the fractures (top) and matrix blocks (bottom) on $Z_{0.5}$ when $t = 6$ yrs.

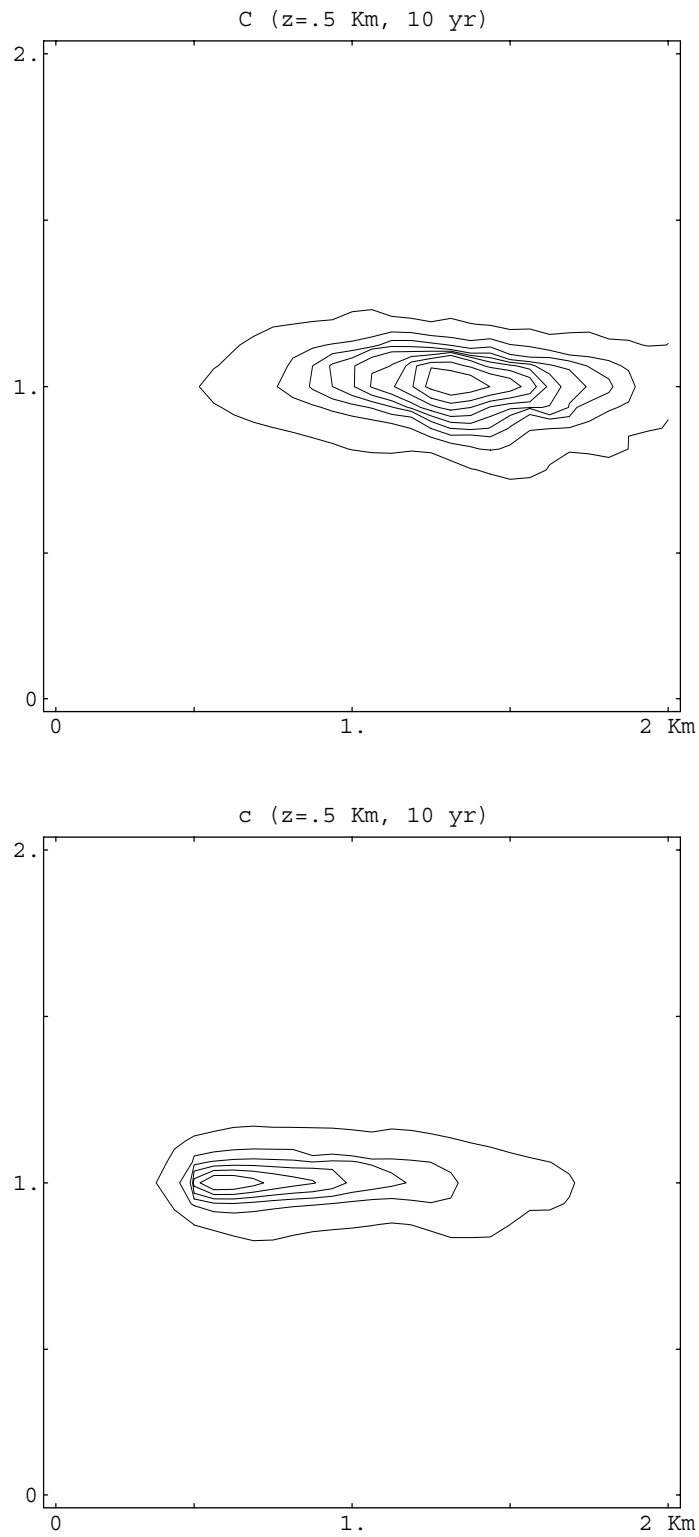


Figure 10: The concentrations for the fractures (top) and matrix blocks (bottom) on $Z_{0.5}$ when $t = 10$ yrs.

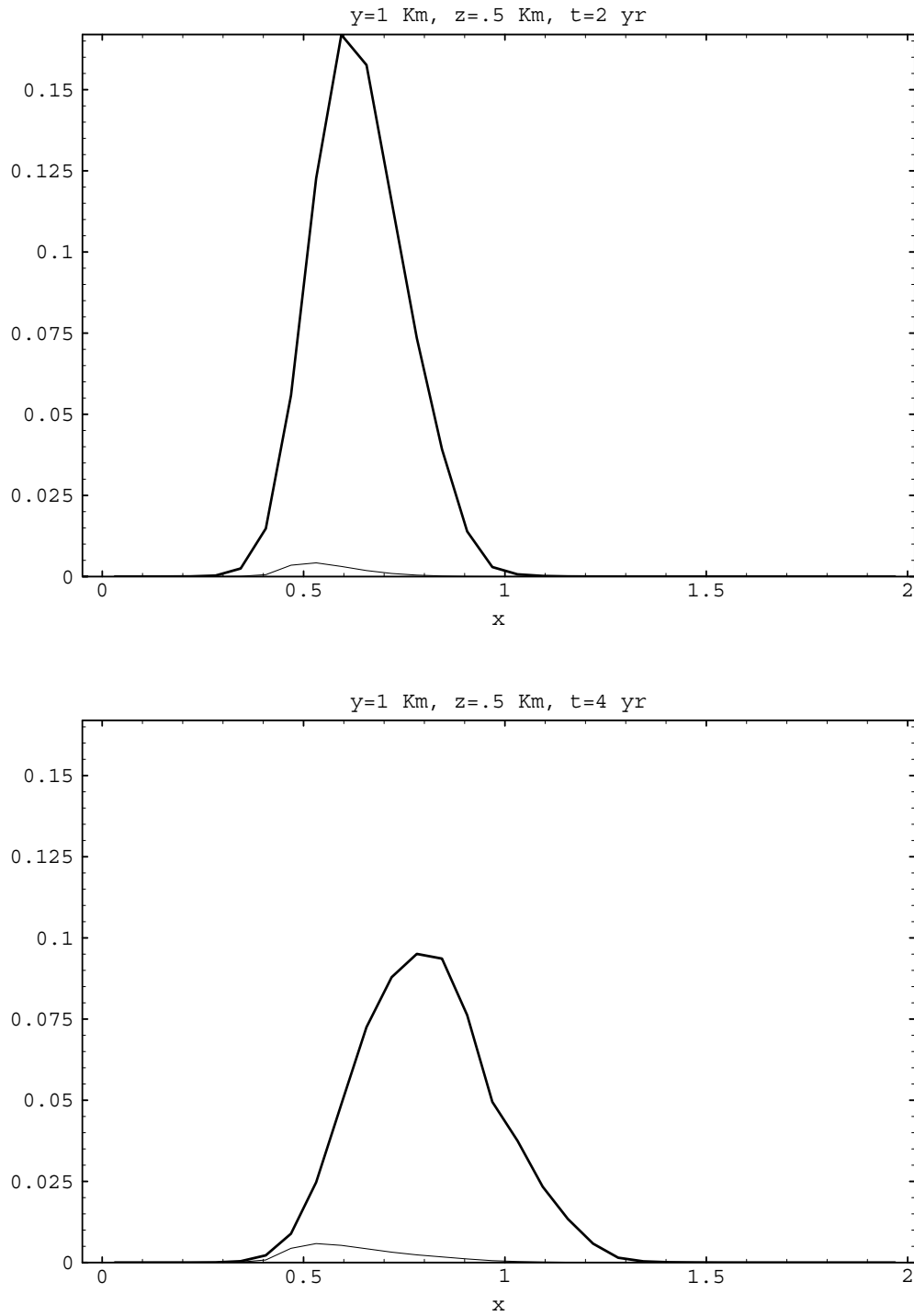


Figure 11: The concentration on the center subdomain, when $t = 2$ and 4 yrs.

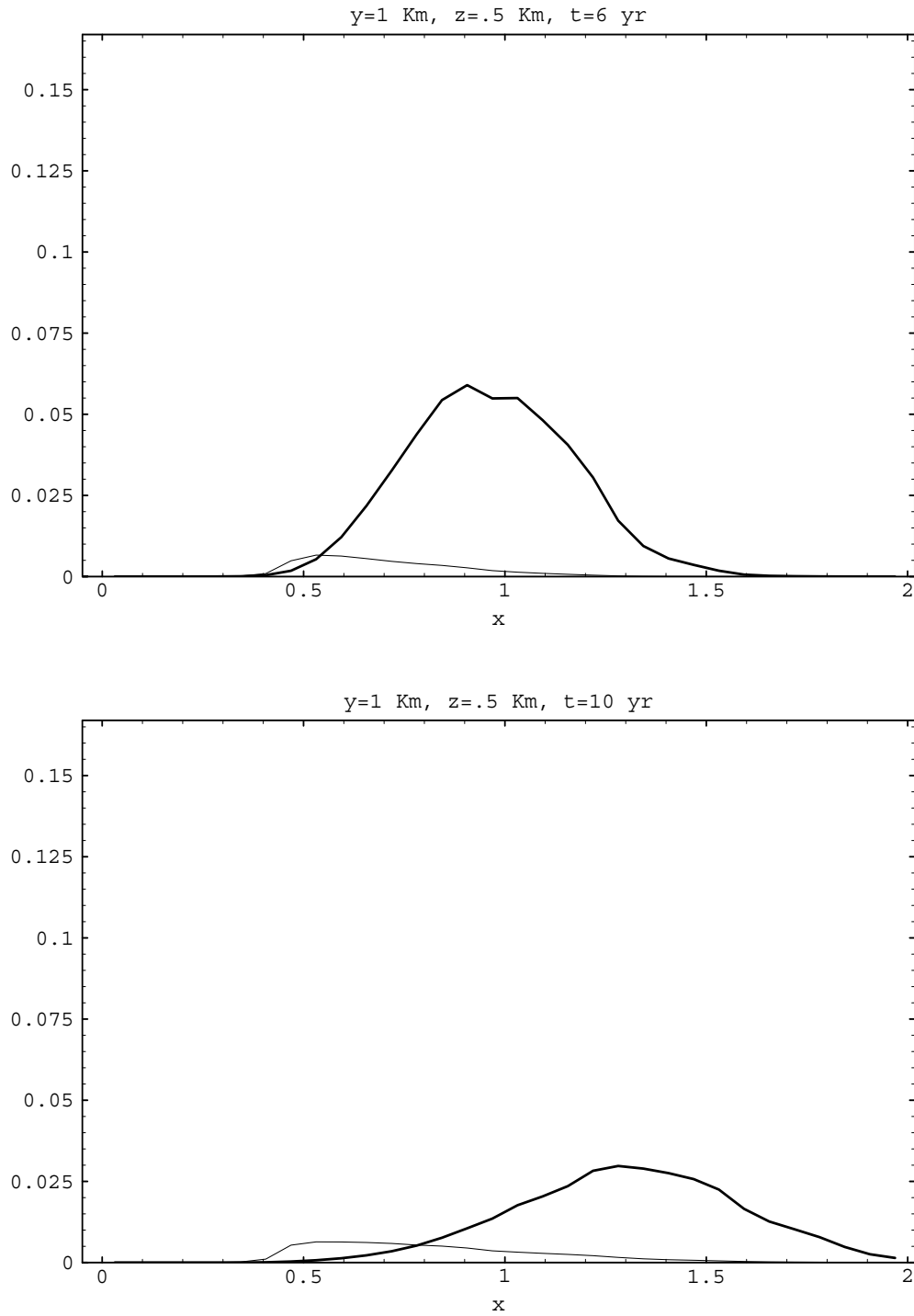


Figure 12: The concentration on the center subdomain, when $t = 6$ and 10 yrs.

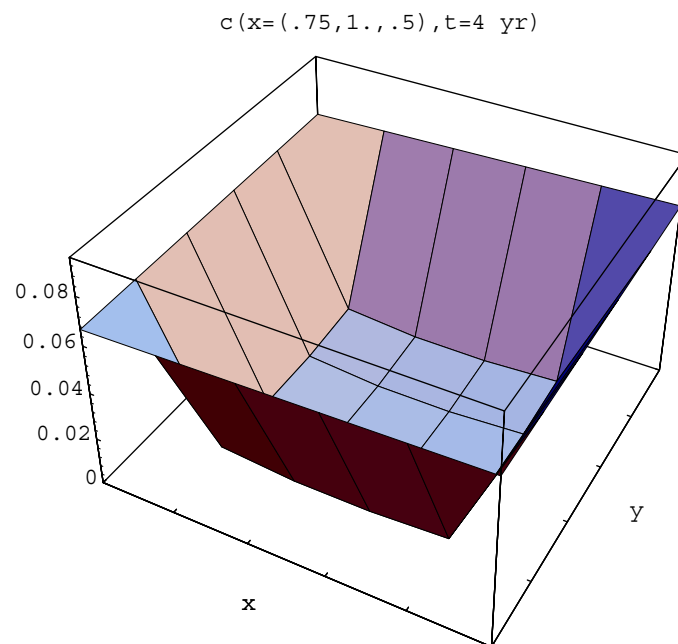
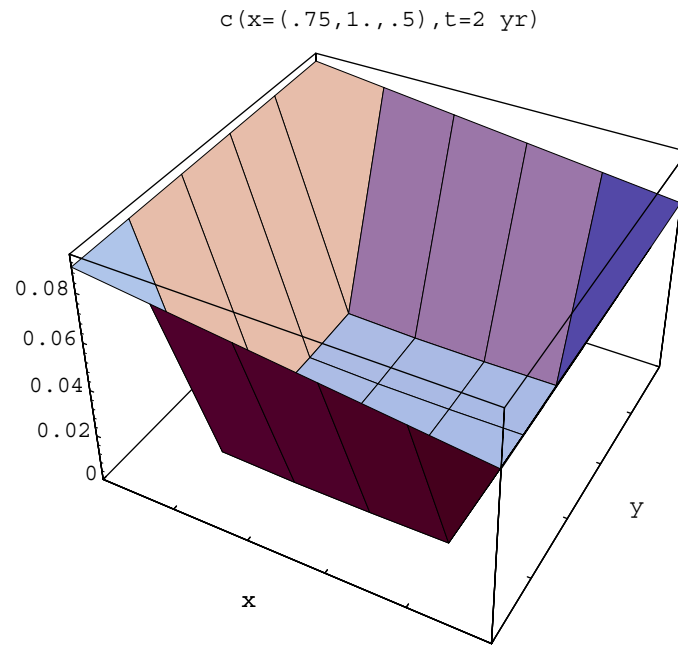


Figure 13: The matrix concentration on Ω_{x_o} , when $t = 2$ and 4 yrs.

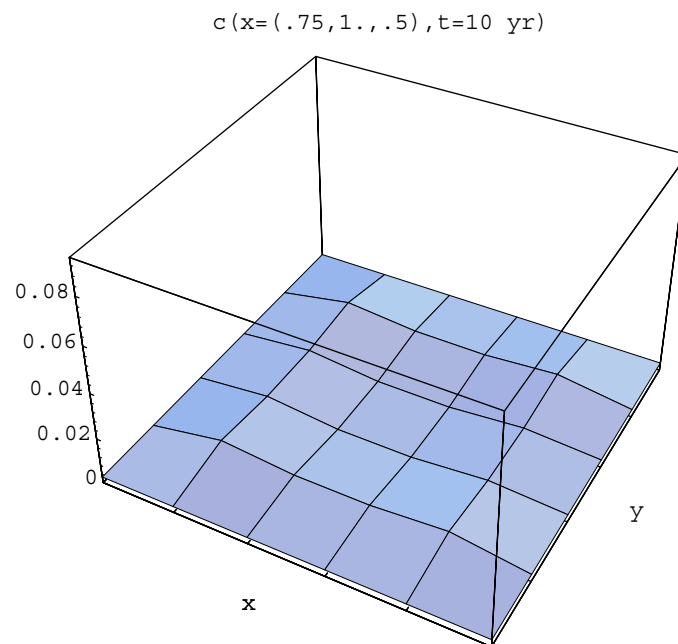
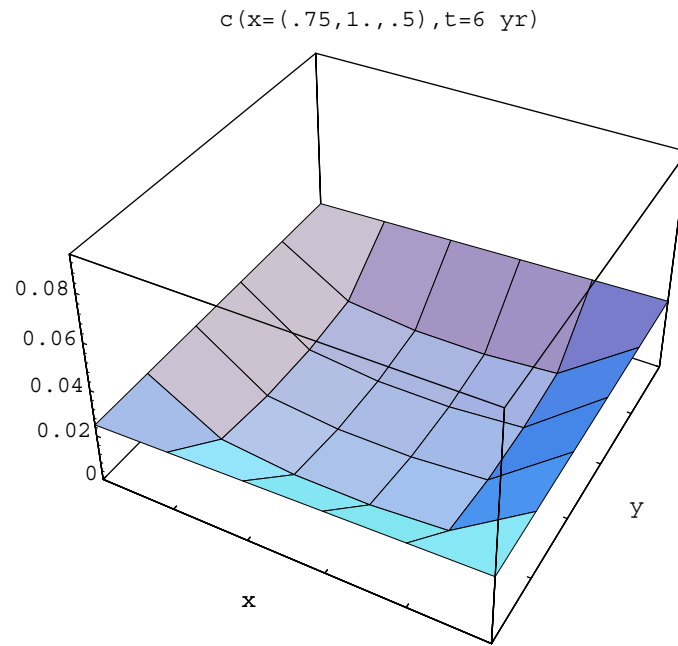


Figure 14: The matrix concentration on Ω_{x_0} , when $t = 6$ and 10 yrs.

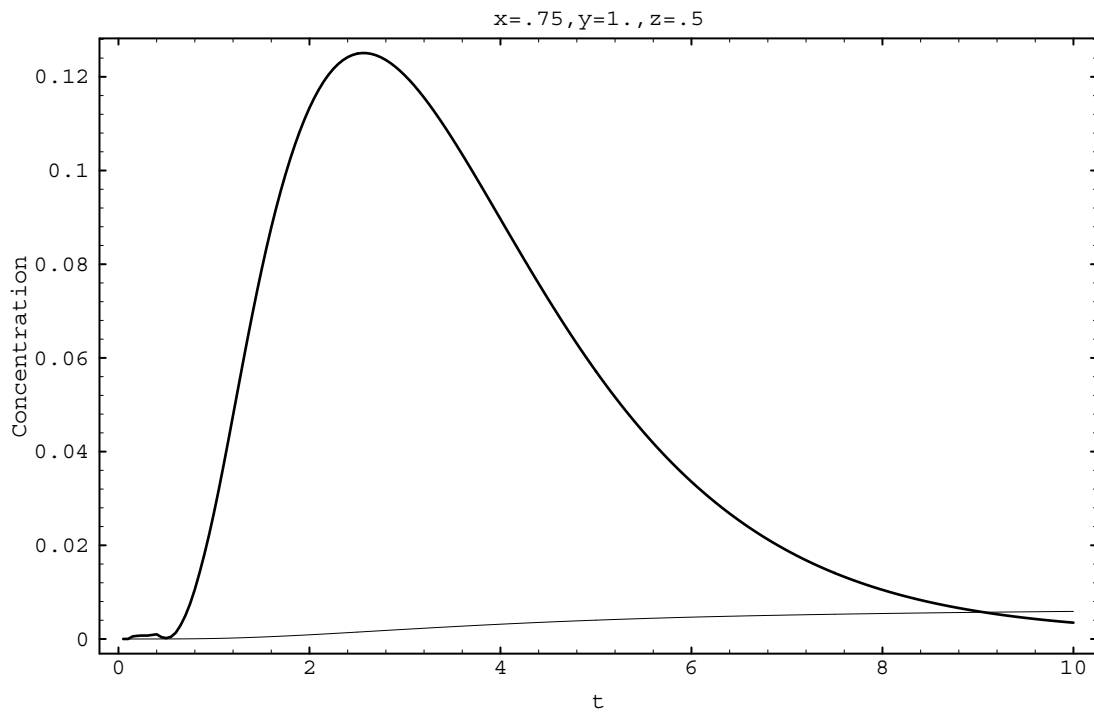


Figure 15: Change of the concentrations on Ω_{x_o} .

6. Conclusions and applications

We have considered an efficient numerical algorithm incorporating an operator splitting technique and DD methods for solving contaminant transport in 3D fractured porous media. The operator splitting scheme is a four-step direct algorithm in order for the coupled matrix-fracture system to be decoupled into easy subsystems. The matrix concentration has been approximated by the backward Euler scheme in time and by the standard trilinear finite element method in space. The MMOC has been employed to effectively handle the convection-dominated fracture fluid; nonoverlapping DD iterative procedures have been introduced to solve the global fracture problems by RTN0. To match the continuity of the normal flux on the subdomain interfaces, we have adopted a modified Robin interface boundary condition. Relaxation parameters have been introduced to accelerate the convergence of the DD iteration. The DD methods have been proved efficient from numerical experiments.

Acknowledgement

This paper is a part of my Ph.D. dissertation (August, 1995); for various reasons, I lost a chance to publish it earlier. I would like to express my sincere thanks to my advisor, Prof. Jim Douglas, Jr., Purdue University, for his valuable comments and encouragement, and for his continuous support on my research activities.

References

- [1] T. ARBOGAST, *On the simulation of incompressible, miscible displacement in a naturally fractured petroleum reservoir*, R.A.I.R.O. Modél. Math. Anal. Numér., 23 (1989), pp. 5–51.
- [2] T. ARBOGAST, A. CHILAKAPATI, AND M. WHEELER, *A characteristic-mixed finite element method for contaminant transport and miscible displacement*, in Computational Methods in Water Resources, T. Russell, R. Ewing, C. Brebbia, W. Gray, and G. Pindar, eds., Southampton, U.K., 1992, Computational Mechanics Publications, pp. 77–84.
- [3] T. ARBOGAST, J. DOUGLAS, JR., AND U. HORNUNG, *Derivation of the double porosity model of single phase flow via homogenization theory*, SIAM J. Math. Anal., 21 (1990), pp. 823–836.
- [4] ———, *Modeling of naturally fractured reservoirs by formal homogenization techniques*, in Frontiers in Pure and Applied Mathematics, R. Dautray, ed., Amsterdam, 1991, Elsevier, pp. 1–19.

- [5] K. AZIZ AND T. SETTARI, *Petroleum Reservoir Simulation*, Applied Science Publishers, London, 1979.
- [6] J. BEAR, *Dynamics of Fluids in Porous Media*, Dover Publication Inc., New York, 1972.
- [7] L. COWSAR AND M. WHEELER, *Parallel domain decomposition method for mixed finite elements for elliptic partial differential equations*, in Fourth International Symposium on Domain Decomposition Method for Partial Differential Equations, R. Glowinski, G. Meurant, J. Periaux, and O. Widlund, eds., Philadelphia, 1991, SIAM, pp. 358–372.
- [8] B. DESPRÉS, *Domain decomposition method and the Helmholtz problem*, in Mathematical and Numerical Aspects of Wave Propagation Phenomena, G. Cohen, L. Halpern, and P. Joly, eds., Philadelphia, 1991, SIAM, pp. 44–52.
- [9] J. DOUGLAS, JR. AND T. ARBOGAST, *Dual-porosity models for flow in naturally fractured reservoir*, in Dynamics of Fluids in Hierarchical Porous Media, J. Cushman, ed., Academic Press, 1990, pp. 177–221.
- [10] J. DOUGLAS JR., T. DUPONT, AND R. EWING, *Incomplete iteration for time-stepping a Galerkin method for a quasilinear parabolic problem*, SIAM J. Numer. Anal., 16 (1979), pp. 503–522.
- [11] J. DOUGLAS, JR., T. DUPONT, AND P. PERCELL, *A time-stepping method for Galerkin approximations for nonlinear parabolic equations*, in Numerical Analysis, Dundee 1977, vol. 630 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1978.
- [12] J. DOUGLAS, JR., F. FURTADO, AND F. PEREIRA, *On the numerical simulation of waterflooding of heterogeneous petroleum reservoirs*, Computational Geosciences, 1 (1997), pp. 155–190.
- [13] J. DOUGLAS, JR., J. HENSLEY, AND T. ARBOGAST, *A dual-porosity model for waterflooding in naturally fractured reservoir*, Comput. Meth. Appl. Mech. Eng., 87 (1991), pp. 157–174.
- [14] J. DOUGLAS, JR., J. HENSLEY, T. ARBOGAST, P. P. LEME, AND N. NUNES, *Medium and tall block models for immiscible displacement in naturally fractured reservoirs*, Technical Report #130, Center for Applied Mathematics, Purdue University, W. Lafayette, IN 47907, 1990.

- [15] J. DOUGLAS, JR., C.-S. HUANG, AND F. PEREIRA, *The modified method of characteristics with adjusted advection for an immiscible displacement problem*, in *Advances in Computational Mathematics*, vol. 202 of *Lecture Notes in Pure and Appl. Math.*, Dekker, New York, 1999, pp. 53–73.
- [16] J. DOUGLAS, JR. AND S. KIM, *Incomplete characteristic methods in reservoir simulation*. In preparation, 2000.
- [17] ———, *Mass-conservative characteristic methods in reservoir simulation*. Preprint, 2000.
- [18] J. DOUGLAS, JR., P. PAES LEME, J. ROBERTS, AND J. WANG, *A parallel iterative procedure applicable to the approximate solution of second order partial differential equations by mixed finite element methods*, *Numer. Math.*, 65 (1993), pp. 95–108.
- [19] J. DOUGLAS, JR., D. PEACEMAN, AND H. RACHFORD, JR., *A method for calculating multi-dimensional displacement*, *Trans. AIME*, 216 (1959), pp. 297–308.
- [20] J. DOUGLAS, JR., F. PEREIRA, AND L.-M. YEH, *A parallelizable method for two-phase flows in naturally fractured reservoirs*, *Computational Geosciences*, 1 (1997), pp. 333–368.
- [21] ———, *A locally conservative Eulerian-Lagrangian method and its application to nonlinear transport in porous media*. (preprint), 1999.
- [22] ———, *A locally conservative Eulerian-Lagrangian method for flow in a porous medium of a mixture of two components having different densities*. (preprint), 1999.
- [23] J. DOUGLAS, JR. AND J. E. ROBERTS, *Global estimates for mixed methods for second order elliptic equations*, *Math. Comp.*, 44 (1985), pp. 39–52.
- [24] J. DOUGLAS, JR. AND T. RUSSELL, *Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures*, *SIAM J. Numer. Anal.*, 19 (1982), pp. 871–885.
- [25] M. DRYJA AND O. WIDLUND, *Some recent results on Schwarz type domain decomposition algorithms*, in *Domain Decomposition Methods in Science and Engineering*, A. Quarteroni, J. Periaux, Y. Kuznetsov, and O. Widlund, eds., vol. 157 of *Contemporary Mathematics*, Philadelphia, 1994, SIAM, pp. 53–61.

- [26] R. EWING, T. RUSSELL, AND M. WHEELER, *Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics*, *Comput. Meth. Appl. Mech. Eng.*, 47 (1984), pp. 73–92.
- [27] R. EWING AND J. WANG, *Analysis of the schwarz algorithm for mixed finite element methods*, *R.A.I.R.O. Modélisation Mathématique et Analyse*, 26 (1992), pp. 739–756.
- [28] P. FORSYTH, *A control volume finite element approach to napl groundwater contamination*, *SIAM J. Sci. Stat. Comput.*, 12 (1991), pp. 1029–1057.
- [29] L. GELHAR, C. WELTY, AND K. REHFELDT, *A critical review of data on field-scale dispersion in aquifers*, *Water Resources Research*, 28 (1992), pp. 1955–1974.
- [30] R. GLOWINSKI, W. KINTON, AND M. WHEELER, *Acceleration of domain decomposition algorithms for mixed finite element methods by multi-level methods*, in *Third International Symposium on Domain Decomposition Method for Partial Differential Equations*, T. Chan, R. Glowinski, J. Periaux, and O. Widlund, eds., Philadelphia, 1989, SIAM, pp. 263–289.
- [31] R. GLOWINSKI AND M. WHEELER, *Domain decomposition and mixed finite element methods for elliptic problems*, in *Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. Golub, G. Meurant, and J. Periaux, eds., Philadelphia, 1988, SIAM, pp. 144–172.
- [32] S. KIM, *A parallelizable iterative procedure for the Helmholtz problem*, *Appl. Numer. Math.*, 14 (1994), pp. 435–449.
- [33] —, *Domain Decomposition Methods for Contaminant Transport in Fractured Porous Media*, Ph.D. Thesis, Center for Applied Mathematics, Purdue University, W. Lafayette, IN 47907, August 1995.
- [34] —, *Domain decomposition iterative procedures for solving scalar waves in the frequency domain*, *Numer. Math.*, 79 (1998), pp. 231–259.
- [35] —, *On the use of rational iterations and domain decomposition methods for solving the Helmholtz problem*, *Numer. Math.*, 79 (1998), pp. 529–552.
- [36] —, *Artificial damping in multigrid methods*, *Appl. Math. Letters*, (2000). Accepted and to appear.

- [37] P. LIONS, *On the Schwarz alternating method III: a variant for nonoverlapping subdomains*, in Domain Decomposition Methods for Partial Differential Equations, T. Chan, R. Glowinski, J. Periaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM, pp. 202–223.
- [38] J. C. NEDELEC, *Mixed finite elements in \mathbf{R}^3* , Numer. Math., 35 (1980), pp. 315–341.
- [39] D. PEACEMAN, *Fundamentals of Numerical Reservoir Simulation*, Elsevier, New York, 1977.
- [40] P.-A. RAVIART AND J. M. THOMAS, *A mixed finite element method for second order elliptic problems*, in Mathematical Aspects of the Finite Element Method, I. Galligani and E. Magenes, eds., vol. 606 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, New York, 1977, pp. 292–315.
- [41] A. SCHEIDEGGER, *The Physics of Flow Through Porous Media*, University of Toronto Press, Toronto, 1974.
- [42] A. M. SPAGNOULO, *Approximation of nuclear contaminant through porous media*, Ph.D. Thesis, Technical Report # 319, Center for Applied Mathematics, Purdue University, W. Lafayette, IN 47907, July 1998.
- [43] W. TANG, *Generalized Schwarz splittings*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 573–595.