Noname manuscript No.
(will be inserted by the editor)

An Adaptive Mesh Refinement Algorithm for Compressible Two-Phase Flow In Porous Media

George S. H. Pau · John B. Bell · Ann S. Almgren · Kirsten M. Fagnan · Michael J. Lijewski

the date of receipt and acceptance should be inserted later

Abstract We describe a second-order accurate sequential algorithm for solving two-phase multicomponent flow in porous media. The algorithm incorporates an unsplit second-order Godunov scheme that provides accurate resolution of sharp fronts. The method is implemented within a block structured mesh refinement framework that allows grids to dynamically adapt to features of the flow and enables efficient parallelization of the algorithm. We demonstrate the second-order convergence rate of the algorithm and the accuracy of the AMR solutions compared to uniform fine-grid solutions. The algorithm is then used to simulate the leakage of gas from a LPG storage cavern. The algorithm captures the complex behavior of the resulting flow. We further examine differences resulting from using different relative permeability functions.

Mathematics Subject Classification (2000) 76S05 · 65M08 · 65M50

1 Introduction

Modeling of subsurface flows that include a gaseous phase involves processes in which compressibility effects can be important. For example, in modeling the leakage of gas from a LPG storage cavern, the density of the gaseous phase decreases rapidly as it rises to the ground surface. Variation in pressure can also be sufficiently large that the incompressible assumption is no longer valid for the entire domain being considered. This usually happens when the size of the domain is very large, for example when modeling the long-term migration of supercritical CO_2 or aromatic contaminants. In the latter case, large pressure difference may lead to sublimation and condensation of the aromatic contaminants in different regions of the domain.

For an isothermal system, the formulation for a compressible multiphase system is usually a generalization of its incompressible counterpart, see for example [6, 8]. The difference lies in how the phase pressures are treated in the formulations, which ultimately affect the design of the numerical algorithms. The approach that we use solves for only one of the phase pressures since the other phase pressure can be determined based on the the capillary pressure function, which is usually modeled as a function of saturation. Chavent and Jaffré [6] and Amaziane et al. [2], on the other hand, use a global pressure formulation that removes the nonlinear capillary pressure gradient term from the flux terms. In non-isothermal system, conservation of energy must be taken into account, for which several formulations have been proposed in [13, 26].

The formulation we use is based on a total velocity splitting approach that follows the compositional formulation of Acs et al [1]. Analysis of this type of approach for isothermal systems is discussed in [23, 24] and for multiphase, compositional thermal systems in [26]. This formulation allows us to construct

George S. H. Pau
Earth Science Division, Lawrence Berkeley National Laboratory, Berkeley, CA.
E-mail: gpau@lbl.gov

John B. Bell, Ann S. Almgren and Michael J. Lijewski
Center of Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA.
E-mail: jbbell@lbl.gov, asalmgren@lbl.gov, mjlijewski@lbl.gov

Kirsten M. Fagnan
National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, Berkeley, CA.
E-mail: kmfagnan@lbl.gov

a sequential algorithm that can be accurately and efficiently parallelized. For an incompressible system, this leads to the IMPES class of methods where an elliptic pressure equation is solved implicitly and the component density equations are solved explicitly. In a compressible system, we need to solve a parabolic pressure equation. Note that the operator splitting approach is less common for highly compressible system and non isothermal system; implicit approaches are usually the norm [13].

We then use the total velocity formulation to construct an efficient algorithm that is implemented within a block-structured adaptive mesh refinement framework. The existence of localized phenomena such as steep concentration gradients and sharp fronts in multiphase flow motivates the need for dynamic gridding that allows these phenomena to be accurately and efficiently resolved. Dynamic local grid refinement approaches in subsurface flow are first considered by Heinemann [12] and Ewing *et al.* [11]. More recent papers [21, 9] discuss development of adaptive techniques in the context of unstructured grids. Related approaches were developed by Nilsson *et al.* [16] who use a spatially anisotropic refinement strategy with no temporal refinement and by Edwards [10] who uses a cell-by-cell refinement strategy. The approach to local refinement considered here is based on block structured-grid adaptive mesh refinement. This type of approach, based on the strategy introduced for gas dynamics by Berger and Colella [4],was first applied to porous media flow by Hornung and Trangenstein [15] and by Propp [19]. Additional developments are discussed by Trangenstein [22], Trangenstein and Bi [25], Hoang and Kleppe [14] and Pau et al. [18]. In this approach, regions to be refined are uniformly subdivided in both space and time.

In subsequent sections, we give the formulation for a two-phase multicomponent compressible system. We then describe a sequential algorithm on the AMR grid. Due to the parabolicity of the pressure equation, the AMR algorithm is significantly different than the algorithm described in [18]. In particular, we solve the system of equations on a composite grid instead of using the subcycling procedure described in [18]. As a consequence, there is no refinement in time. We conclude with some convergence studies and as well as a more complex example that considers 2D studies of a gas leakage problem.

2 Formulation

We consider two-phase multicomponent compressible flow in a porous medium under isothermal condition. The porous medium is assumed to be compressible with porosity, $\phi = \phi(p)$. We consider a system in which each component only appears in one of the phases with no mass transfer between the phases. We let n_k^α denote the mass of component k per unit pore volume in phase α. The assumptions above imply that n_k^α is only nonzero for one value of α. We then define the mass per unit pore volume for each phase by

$$n_T^\alpha = \sum_k n_k^\alpha, \quad \alpha = 1, 2. \tag{1}$$

and associated mass fractions $Y_k^\alpha = n_k^\alpha / n_T^\alpha$.

We assume a Darcy flow with Darcy velocity for phase α given by

$$v_\alpha = -\lambda_\alpha (\nabla p_\alpha - \rho_\alpha g), \tag{2}$$

where the phase mobility, $\lambda_\alpha = \frac{\kappa k_{r,\alpha}}{\mu_\alpha}$, $p_\alpha = p + p_{c,\alpha}$ is the pressure of phase α, and $p_{c,\alpha}$ is the capillary pressure.

For each phase we define a specific volume $\tau_\alpha(p_\alpha, Y_k^\alpha)$ that represents the pore volume per unit mass occupied by phase α at a given phase pressure and composition. Then $u_\alpha = n_T^\alpha \tau_\alpha$ is the fraction of the pore volume occupied by phase α. The mass conservation equations are then given by

$$\frac{\partial \phi n_k^\alpha}{\partial t} + \nabla \cdot \frac{n_k^\alpha v_\alpha}{u_\alpha} = S_k^\alpha, \qquad k = 1, \ldots, K. \tag{3}$$

To close the system we define an equation of state

$$U = \sum_\alpha n_T^\alpha \tau_\alpha = \sum_\alpha u_\alpha = 1 \tag{4}$$

We note that U depends on phase pressures and compositions; however, we will also view it implicitly as a function of time. We also note that analytically, the u_α correspond to saturations; however, we use a distinct notation because in the fractional step numerical scheme presented later we will not require that the u_α discretely sum to one.

Our basic discretization strategy is based on a fractional step scheme that uses a total velocity splitting approach [1,18]. In this approach we first derive an equation for a global reference pressure. From the solution of the pressure equation we then define a total velocity as the sum of the phase velocities. Finally we rewrite the component conservation equations in terms of the total velocity.

To derive the pressure equation, we first differentiate (4) with respect to time and multiply the resulting equation by ϕ to obtain

$$\phi\tau_\alpha \frac{\partial n_T^\alpha}{\partial t} + \phi n_T^\alpha \frac{\partial \tau_\alpha}{\partial p}\frac{\partial p}{\partial t} + \phi n_T^\alpha \frac{\partial \tau_\alpha}{\partial Y_k^\alpha}\frac{\partial Y_k^\alpha}{\partial t} = 0. \tag{5}$$

Here we have assumed $p_{c,\alpha}$ is not a function of p, allowing us to write $\frac{\partial\tau_\alpha}{\partial p} = \frac{\partial\tau_\alpha}{\partial p_\alpha}\frac{\partial p_\alpha}{\partial p} = \frac{\partial\tau_\alpha}{\partial p_\alpha}$. To obtain the desired equation for p, we first derive several identities. First we derive an equation for the evolution of the total mass in each phase by summing the component conservation equations over all the components in that phase to obtain

$$\frac{\partial \phi n_T^\alpha}{\partial t} + \nabla\cdot\frac{n_T^\alpha v_\alpha}{u_\alpha} = \sum_k S_k^\alpha = S^\alpha. \tag{6}$$

From (3) and the definition of Y_k^α, we then obtain

$$\begin{aligned}
\frac{\partial \phi n_T^\alpha Y_k^\alpha}{\partial t} + \nabla\cdot\frac{n_T^\alpha Y_k^\alpha v_\alpha}{u_\alpha} &= S_k^\alpha \\
Y_k^\alpha\frac{\partial \phi n_T^\alpha}{\partial t} + \phi n_T^\alpha\frac{\partial Y_k^\alpha}{\partial t} + \frac{n_T^\alpha}{u_\alpha}v_\alpha\cdot\nabla Y_k^\alpha + Y_k^\alpha\nabla\cdot\frac{n_T^\alpha}{u_\alpha}v_\alpha &= S_k^\alpha \\
Y_k^\alpha\left(\frac{\partial \phi n_T^\alpha}{\partial t} + \nabla\cdot\frac{n_T^\alpha}{u_\alpha}v_\alpha\right) + n_T^\alpha\left(\phi\frac{\partial Y_k^\alpha}{\partial t} + \frac{v_\alpha}{u_\alpha}\cdot\nabla Y_k^\alpha\right) &= S_k^\alpha \\
Y_k^\alpha(S^\alpha) + n_T^\alpha(\phi\frac{\partial Y_k^\alpha}{\partial t} + \frac{v_\alpha}{u_\alpha}\cdot\nabla Y_k^\alpha) &= S_k^\alpha \\
n_T^\alpha(\phi\frac{\partial Y_k^\alpha}{\partial t} + \frac{v_\alpha}{u_\alpha}\cdot\nabla Y_k^\alpha) &= S_k^\alpha - Y_k^\alpha S^\alpha.
\end{aligned} \tag{7}$$

In addition, we note that by definition

$$\tau_\alpha\frac{n_T^\alpha}{u_\alpha} = 1 \quad . \tag{8}$$

Also,

$$\nabla\tau_\alpha = \tau_\alpha\varepsilon_\alpha\nabla p + \tau_\alpha\bar\beta_k^\alpha\nabla Y_k^\alpha; \tag{9}$$

where

$$\varepsilon_\alpha = \frac{1}{\tau_\alpha}\frac{\partial\tau_\alpha}{\partial p}, \quad \bar\beta_k^\alpha = \frac{1}{\tau_\alpha}\frac{\partial\tau_\alpha}{\partial Y_k^\alpha} \tag{10}$$

are, respectively, the phase compressibility and compositional derivative of the specific volume. We also define the compressibility of the porous medium as

$$\varepsilon_r = \frac{\partial\phi}{\partial p}. \tag{11}$$

Each term in (5) can then be expressed in terms of (7)–(11):

$$\begin{aligned}
\phi\tau_\alpha\frac{\partial n_T^\alpha}{\partial t} &= \tau_\alpha\frac{\partial\phi n_T^\alpha}{\partial t} - \tau_\alpha n_T^\alpha\frac{\partial\phi}{\partial t} \\
&= \tau_\alpha\left(-\nabla\cdot\frac{v_\alpha}{\tau_\alpha} + S^\alpha\right) - \varepsilon_r\frac{\partial p}{\partial t},
\end{aligned} \tag{12}$$

$$\phi n_T^\alpha\frac{\partial\tau_\alpha}{\partial p}\frac{\partial p}{\partial t} = \phi\varepsilon_\alpha u_\alpha\frac{\partial p}{\partial t}, \tag{13}$$

$$\phi n_T^\alpha\frac{\partial\tau_\alpha}{\partial Y_k^\alpha}\frac{\partial Y_k^\alpha}{\partial t} = \tau_\alpha\bar\beta_k^\alpha\left(-\frac{n_T^\alpha v_\alpha}{u_\alpha}\nabla Y_k^\alpha + S_k^\alpha - Y_k^\alpha S^\alpha\right). \tag{14}$$

Substituting these expressions into 5 leads to a pressure equation given by

$$(\varepsilon_r - \phi\varepsilon_\alpha u_\alpha)\frac{\partial p}{\partial t} = -\tau_\alpha\nabla\cdot\frac{v_\alpha}{\tau_\alpha} - \bar\beta_k^\alpha v_\alpha\cdot\nabla Y_k^\alpha + \tau_\alpha S^\alpha + \tau_\alpha\bar\beta_k^\alpha(S_k^\alpha - Y_k^\alpha S^\alpha), \tag{15}$$

If we express

$$\frac{n_T^\alpha v_\alpha}{u_\alpha}\nabla Y_k^\alpha = \nabla\cdot Y_k^\alpha\frac{v_\alpha}{\tau_\alpha} - Y_k^\alpha\nabla\cdot\frac{v_\alpha}{\tau_\alpha}, \tag{16}$$

(15) can be rewritten as

$$
\begin{aligned}
(\varepsilon_r - \phi \varepsilon_\alpha u_\alpha)\frac{\partial p}{\partial t} &= -\tau_\alpha \nabla \cdot \frac{v_\alpha}{\tau_\alpha} + \tau_\alpha \bar{\beta}_k^\alpha (-\nabla \cdot (Y_k^\alpha \frac{v_\alpha}{\tau_\alpha}) + Y_k^\alpha \nabla \cdot \frac{v_\alpha}{\tau_\alpha}) + \tau_\alpha S^\alpha + \tau_\alpha \bar{\beta}_k^\alpha (S_k^\alpha - Y_k^\alpha S^\alpha) \\
&= -\tau_\alpha (1 - \bar{\beta}_k^\alpha Y_k^\alpha) \nabla \cdot \frac{v_\alpha}{\tau_\alpha} - \tau_\alpha \bar{\beta}_k^\alpha \nabla \cdot (Y_k^\alpha \frac{v_\alpha}{\tau_\alpha}) + \tau_\alpha S^\alpha + \tau_\alpha \bar{\beta}_k^\alpha (S_k^\alpha - Y_k^\alpha S^\alpha). \qquad (17)
\end{aligned}
$$

Substituting (2) into (17) then leads to a single equation for p that is parabolic.

We now rewrite the component density equations based on the total velocity splitting approach. Defining the total velocity as

$$v_T = \sum_\alpha v_\alpha, \qquad (18)$$

the phase velocities are given by

$$
\begin{aligned}
v_1 &= \frac{\lambda_1}{\lambda_T} v_T - \frac{\lambda_1 \lambda_2}{\lambda_T} \{(\rho_1 - \rho_2) g - \nabla p_{c,2}\}, \qquad (19) \\
v_2 &= \frac{\lambda_2}{\lambda_T} v_T - \frac{\lambda_1 \lambda_2}{\lambda_T} \{(\rho_2 - \rho_1) g + \nabla p_{c,2}\}, \qquad (20)
\end{aligned}
$$

where $p_2 = p_1 + p_{c,2}$, and $p = p_1$. The component density equations for a two-component two-phase problem can then be expressed as

$$
\begin{aligned}
\frac{\partial \phi n_1}{\partial t} + \nabla \rho_1 \left\{ \frac{\lambda_1}{\lambda_T} v_T - \frac{\lambda_1 \lambda_2}{\lambda_T} (\rho_2 - \rho_1) g \right\} &= -\nabla \rho_1 \frac{\lambda_1 \lambda_2}{\lambda_T} \nabla p_{c,2}, \qquad (21) \\
\frac{\partial \phi n_2}{\partial t} + \nabla \rho_2 \left\{ \frac{\lambda_2}{\lambda_T} v_T - \frac{\lambda_1 \lambda_2}{\lambda_T} (\rho_1 - \rho_2) g \right\} &= \nabla \rho_2 \frac{\lambda_1 \lambda_2}{\lambda_T} \nabla p_{c,2}. \qquad (22)
\end{aligned}
$$

For an incompressible system, (17) simplifies to

$$
\begin{aligned}
-\tau_\alpha \nabla \cdot \frac{v_\alpha}{\tau_\alpha} - \bar{\beta}_k^\alpha v_\alpha \cdot \nabla Y_k^\alpha + \tau_\alpha S^\alpha + \tau_\alpha \bar{\beta}_k^\alpha (S_k^\alpha - Y_k^\alpha S^\alpha) &= 0 \\
-\nabla \cdot v_T + (\tau_\alpha \nabla \frac{1}{\tau_\alpha}) \cdot v_\alpha - \bar{\beta}_k^\alpha \nabla Y_k^\alpha \cdot v_\alpha + \tau_\alpha S^\alpha + \tau_\alpha \bar{\beta}_k^\alpha (S_k^\alpha - Y_k^\alpha S^\alpha) &= 0 \\
-\nabla \cdot v_T + \tau_\alpha S^\alpha + \tau_\alpha \bar{\beta}_k^\alpha (S_k^\alpha - Y_k^\alpha S^\alpha) &= 0 \qquad (23)
\end{aligned}
$$

since $(1/\tau_\alpha)\nabla \tau_\alpha = \bar{\beta}_k^\alpha \nabla Y_k^\alpha$ for incompressible flow. Thus we arrive at the velocity divergence condition in the incompressible formulation.

For simplicity, we will primarily look at a two-component two-phase system. In this case, equation (15) simplifies to

$$(\varepsilon_r - \phi \varepsilon_\alpha u_\alpha)\frac{\partial p}{\partial t} = -\tau_\alpha \nabla \cdot \frac{v_\alpha}{\tau_\alpha} + \tau_\alpha S^\alpha. \qquad (24)$$

Note that

$$-\tau_\alpha \nabla \cdot \frac{v_\alpha}{\tau_\alpha} = -\nabla \cdot v_T - (\tau_\alpha \nabla \frac{1}{\tau_\alpha}) \cdot v_\alpha. \qquad (25)$$

In slightly compressible system, the magnitude of the second term $\frac{1}{\tau_\alpha} \nabla \tau_\alpha \cdot v_\alpha$ is usually small and can thus be neglected [6,7]. However, for highly compressible system, the magnitude of this term is significant and must be included in the formulation.

3 Numerical Approach

The numerical approach we propose is a modification of the algorithm we developed for incompressible flow [18]. Similar to that algorithm, the component equations (21) – (22) and the pressure equation (24) are solved using a sequential approach: the pressure equation is solved implicitly using a cell-centered finite difference method and the mass conservation equations are solved using an explicit treatment of advection based on a second-order Godunov method. In addition, the algorithm is constructed so that the overall splitting approach is second-order accurate in time. We also use the same adaptive mesh refinement framework we used in our previous work. This framework represents the solution on a nested hierarchy of logically rectangular grids [5] that dynamically change with time based on a set of user-defined refinement criteria. We use component density gradients of all the components as the refinement criterion in our simulations, resulting in finer grids in regions where large concentration gradients are present.

A major difference from [18] is that the pressure equation is now nonlinear and parabolic. This has direct consequences for the entire AMR algorithm, leading to an algorithm that differs significantly from [18]. In Section 3.1, we describe our AMR algorithm, focusing on the approximation of the pressure equation since the approach for solving component conservation equations is similar to [18]. We ignore capillary effects for notational convenience and describe the algorithm based on a spatial discretization of a 2D domain. We then describe the time-stepping scheme in Section 3.2 that leads to second-order accuracy in time. The spatial discretization uses a volume of fluid approach in which $\mathbf{n}_{ij}^k$ denotes the average value of the vector of component densities, $\mathbf{n}$, over cell (i,j) at time t^k; n and p are defined on cell centers while v_T are defined on cell edges.

3.1 AMR algorithms

The three main components of the AMR algorithm are:

1. *Averaging and interpolation.* Assuming the solutions at finer grid levels are more accurate, we average the finer-grid solution down to coarser grid levels. During a regrid, we interpolate solutions from the coarser levels to the finer levels when new fine grids are overlaid over coarser grids. We ensure the interpolation procedure is conservative for cell-centered quantities and preserves divergence condition for edge-centered quantities.
2. *Pressure equation.* The parabolic equation is solved implicitly on the composite grid.
3. *Component density equations.* A second order Godunov method is used to solve the hyperbolic part of the component density equations on the composite grid. The solution at each level of the grid hierarchy is solved independently and boundary conditions at finer levels are derived from the coarser levels. This leads to a flux mismatch along the coarse-fine boundaries; this discrepancy is corrected by replacing the coarse flux by the sum of the fine fluxes along those boundaries.

A common time step is used for all levels of grids, and it is limited by the CFL number computed based on the finest grid. Note that we have considered schemes that follow the level-solve and synchronization procedure of [18]. However, the associated computational overhead is not justifiable. We note that the parallelization of the code is straightforward within the BoxLib AMR framework [20] used for the implementation. Nevertheless, it is not the goal of this paper to examine the parallel performance of the resulting algorithm.

3.1.1 Averaging and interpolation

We define the averaging operator $\mathcal{A}^{\ell,c}$ for cell-centered quantities, such as n^ℓ and p^ℓ, as

$$\mathcal{A}^{\ell,c}\phi^{\ell+1} = \frac{1}{A^\ell}\sum_r A^{\ell+1}\phi_r^{\ell+1}, \tag{26}$$

where A^ℓ is the cell size at level ℓ. Similarly, the averaging operator $\mathcal{A}^{\ell,e}$ for edge-centered quantities, such as v_T^ℓ, is given by

$$\mathcal{A}^{\ell,e}v^{\ell+1} = \frac{1}{\Delta x^\ell}\sum_r \Delta x^{\ell+1}v_r^{\ell+1}. \tag{27}$$

Both averaging operators are conservative.

We define an interpolation operator $\mathcal{I}^{\ell,c}$ for the cell-centered quantities as

$$\phi^{\ell+1} = \mathcal{I}^{\ell,c}\phi^\ell. \tag{28}$$

Conservation interpolation will be used. Similarly, we define an interpolation operator $\mathcal{I}^{\ell,e}$ for the edge-centered quantities. For the velocity, we must ensure the divergence of the velocity is preserved to ensure mass is conserved when the velocity is used in the advection equation. To achieve this a two-step procedure is used. Along the common edge of fine and coarse cells, a simple linear interpolation is used: for example, along the left edge,

$$\begin{aligned} u^f_{2i-1/2,2j} &= u^c_{i-1/2,j} - \frac{\Delta x^f}{2}\frac{du^c_{i-1/2,j}}{dy}, \\ u^f_{2i-1/2,2j+1} &= u^c_{i-1/2,j} + \frac{\Delta x^f}{2}\frac{du^c_{i-1/2,j}}{dy}. \end{aligned}$$

Similar formulae can be derived for other edges. For internal edges, simple interpolation is first used to obtain an approximate $\tilde{u}^f$. A discrete projection is then used within each coarse cell to ensure u^f conserves the divergence. Let

$$u^f = \tilde{u}^f + \frac{\partial \phi}{\partial x}, \tag{29}$$

$$v^f = \tilde{v}^f + \frac{\partial \phi}{\partial y}. \tag{30}$$

Then, we solve

$$\nabla^2 \phi = \frac{\partial u^f}{\partial x} + \frac{\partial v^f}{\partial y} - d = r. \tag{31}$$

where $\int_{\Omega^c} d = D = \int_{\Omega^c} \frac{\partial u^c}{\partial x} + \frac{\partial v^c}{\partial y}$, and $\nabla \phi = 0$ along the boundaries. Since the problem has no unique answer and we are only interested in gradient of ϕ, we can arbitrarily fixed ϕ for a particular cell to zero. For a reference ratio of 2, $d = D/4$, and the solution ϕ is given by

$$\begin{aligned}\begin{pmatrix} \phi_{2i,2j} \\ \phi_{2i+1,2j} \\ \phi_{2i,2j+1} \end{pmatrix} &= \begin{pmatrix} -1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{2} & -\frac{1}{4} & -\frac{3}{4} \end{pmatrix} \begin{pmatrix} r_{2i,2j} \\ r_{2i+1,2j} \\ r_{2i,2j+1} \end{pmatrix}, \\ \phi_{2i+1,2j+1} &= 0.\end{aligned}$$

3.1.2 Pressure solve

The pressure equation is spatially discretized using a cell-centered finite difference method and temporally discretized based on the Crank-Nicolson scheme. We let $D^{E \to C}$ and $G^{C \to E}$ be the discretized forms of the divergence and gradient operators where $D^{E \to C}$ defines a cell-centered divergence in terms of normal components of a vector field on cell faces and $G^{C \to E}$ defines the normal component of the gradient on cell faces from the values at cell centers. Using the splitting given by (25), the discretized pressure equation at a particular AMR level is given by

$$\begin{aligned}\Big(\beta^{\ell,k+1/2} - \frac{\Delta t}{2}\Big(D^{E \to C} \lambda_\alpha^{\ell,k+1} G^{C \to E} + (\tau_\alpha^{\ell,k+1} G^{C \to E} \frac{1}{\tau_\alpha^{\ell,k+1}}) \cdot \lambda_\alpha^{\ell,k+1} G^{C \to E}\Big)\Big) p^{\ell,k+1} = \\ - \phi \varepsilon_\alpha^{\ell,k+1/2} u_\alpha^{\ell,k+1/2} p^{\ell,k} - \frac{\Delta t}{2} \left(D^{E \to C} v_T^{\ell,k} + \{(\tau_\alpha \nabla \frac{1}{\tau_\alpha}) \cdot v_\alpha\}^{\ell,k} \right) \\ - \frac{\Delta t}{2} \left(D^{E \to C} (\lambda^{\ell,k+1} \rho^{\ell,k+1} g) + (\tau_\alpha^{\ell,k+1} G^{C \to E} \frac{1}{\tau_\alpha^{\ell,k+1}}) \cdot \lambda_\alpha^{\ell,k+1} \rho_\alpha^{\ell,k+1} g \right),\end{aligned} \tag{32}$$

where

$$D^{E \to C}(\sum_\alpha \lambda_\alpha G^{C \to E} p) = \frac{1}{\Delta x}(F^1_{i+1/2,j} - F^1_{i-1/2,j}) + \frac{1}{\Delta y}(F^1_{i,j+1/2} - F^1_{i,j-1/2}); \tag{33}$$

$$\sum_\alpha (\tau_\alpha G^{C \to E} \frac{1}{\tau_\alpha}) \cdot (\lambda_\alpha G^{C \to E} p) = \frac{1}{\Delta x}(F^2_{i+1/2,j} + F^2_{i-1/2,j}) + \frac{1}{\Delta y}(F^2_{i,j+1/2} + F^2_{i,j-1/2}), \tag{34}$$

and

$$\begin{aligned} F^1_{i+1/2,j} &= \left(\sum_\alpha \lambda_{\alpha,i+1/2,j} \right) \frac{p_{i+1,j} - p_{i,j}}{\Delta x}, \\ F^1_{i,j+1/2} &= \left(\sum_\alpha \lambda_{\alpha,i,j+1/2} \right) \frac{p_{i,j+1} - p_{i,j}}{\Delta y}, \\ F^2_{i+1/2,j} &= \left(\frac{1}{2} \sum_\alpha \lambda_{\alpha,i+1/2,j} \tau_{\alpha,i+1/2,j} \frac{1/\tau_{\alpha,i+1,j} - 1/\tau_{\alpha,i,j}}{\Delta x} \right) \frac{p_{i+1,j} - p_{i,j}}{\Delta x}, \\ F^2_{i,j+1/2} &= \left(\frac{1}{2} \sum_\alpha \lambda_{\alpha,i,j+1/2} \tau_{\alpha,i,j+1/2} \frac{1/\tau_{\alpha,i,j+1} - 1/\tau_{\alpha,i,j}}{\Delta y} \right) \frac{p_{i,j+1} - p_{i,j}}{\Delta y}. \end{aligned}$$

We evaluate $\lambda_\alpha = \kappa k_r / \mu$ at edges by taking the harmonic average of κ and arithmetic average of k_r/μ. We note that (33) is a symmetric operator while (34) is a nonsymmetric operator. In addition, these operators can be expressed as differences and averages of fluxes. Thus, only minor modifications need to be made to the multilevel geometric multigrid solver used to implicitly solve the pressure equation on the composite AMR grid. Details of the solver can be found in [17].

In the second term on the RHS of (32), the operator is expressed in term of v_T and $(\tau_\alpha \nabla \frac{1}{\tau_\alpha}) \cdot v_\alpha$ at time k instead of p at time k. This choice implies that when grid changes, we interpolate and average v_T and $(\tau_\alpha \nabla \frac{1}{\tau_\alpha}) \cdot v_\alpha$ at time k to the new grid, and evaluate the operator based on these interpolated or averaged values. If we have expressed the operator in terms of p, the evaluation will involve taking the gradient of the interpolated p at time k, which introduces a first order error to the solutions. In addition, the reconstructed v_T at time k may no longer have the correct discrete divergence. Interpolating these gradient terms directly maintains their second-order accuracy, leading to an AMR discretization that is consistent with the cell-centered finite difference procedure that gives a 2nd-order accurate gradient [3].

3.1.3 Component Density Equations

We use Godunov method to determine the advective flux in the component density equation. For the AMR grid, we solve the equation level by level, with boundary conditions for finer levels derived from the coarser levels. After we have traversed from the coarsest to the finest level, there will be a mismatch in the flux along the coarse-fine boundaries; we replace the coarse fluxes by the sum of the fine fluxes. Note that we do not need to compute solution on grid cells that are covered by finer grid cells. However, for the data structure of the AMR framework we are using, it is algorithmically simpler to solve for the solutions of all grid points on a particular level, and this simplification has no significant impact on the parallel performance. Greater details can be found in [18].

3.2 Time-Stepping Scheme

The above spatial discretization is second-order accurate. To efficiently solve the above equations, we use a sequential approach in which the pressure and component densities equations are solved consecutively. This necessitates a multistep algorithm to achieve second-order accuracy in time. The algorithm can be summarized as follows:

1. Determine $p^{k+1,*}$ with coefficients at time $k+1$ evaluated using n and p at time k. We then determine the total velocity $v_T^{k+1,*}$ from $p^{k+1,*}$.
2. Evaluate $v_T^{k+1/2,*} = \frac{1}{2}(v_T^k + v_T^{k+1,*})$.
3. Evaluate $\rho_\alpha^{k+1,*} = \rho_\alpha(p^{k+1,*})$.
4. Based on $v_T^{k+1/2,*}$, determine $n^{k+1,*}$ based on second order Godunov method.
5. Recompute coefficients at time $k+1$ based on $n^{k+1,*}$ and $p^{k+1,*}$, and determine p^{k+1}. Recompute v_T^{k+1} .
6. Evaluate $v_T^{k+1/2} = \frac{1}{2}(v_T^k + v_T^{k+1})$.
7. Evaluate $\rho_\alpha^{k+1} = \rho_\alpha(p^{k+1})$.
8. Determine n^{k+1} based on $v_T^{k+1/2}$.

In cases in which the spatial discretization error dominates the temporal discretization error, step 1 to 4 is sufficient to get a second-order accurate solution for $\mathbf{n}^{k+1}$. This will half the computational time needed by the full algorithm.

In the algorithm presented here we advance the pressure and all of the component densities. Advancing all of the component density equations guarantees discrete conservation of mass for each component. However, the system is then overdetermined and the time stepping scheme does not explicitly enforce the constraint imposed by the equation of state. Thus, in this approach, we do not exactly satisfy the constraint that the fluid fill the pore volume; i.e., the u_α do not discretely sum to one. To ensure that the deviation does not grow with time, we note that

$$\phi \frac{\partial U}{\partial t} = \phi \frac{1 - U(t)}{\Delta t}; \tag{35}$$

the above is obtained from a first order Taylor expansion in time of the equation of states [1,24]. This introduces a relaxation term $-\phi \frac{1-U(t)}{\Delta t}$ to the right hand side of (32). The approach attempts to correct for the discrepancy at the end of the previous time step. To avoid the introduction of this term from excessively influence the solution, we multiply the term with a heuristic coefficient α_r, where $0 \leq \alpha_r \leq 1$.

Table 1 L^2 errors and convergence rates for n_2, p and v_T; $\alpha_r = 0$, $\Delta p = 1$ atm, and $\mu = 0.0178$.

h, m	$\lVert\delta n_2^h\rVert_{L_2}$	$r_{\delta n_2^h}$	$\lVert\delta p^h\rVert_{L_2}$	$r_{\delta p^h}$	$\lVert\delta v_T^h\rVert_{L_2}$	$r_{\delta v_T^h}$	$\lVert\varepsilon^h\rVert_{L_2}$	r_{ε^h}
1	1.34e-3	–	1.81e-3	–	7.36e-8	–	4.1e-7	–
0.5	3.35e-4	2.00	4.54e-4	2.00	1.78e-8	2.05	1.96e-7	1.06
0.25	8.37e-5	2.00	1.13e-4	2.01	4.42e-9	2.01	9,43e-8	1.06
0.125	2.09e-5	2.00	2.82e-5	2.00	1.11e-9	1.99	4.60e-8	1.04
0.0625	5.24e-6	2.00	6.99e-6	2.01	2.81e-10	1.98	2.27e-8	1.02

4 Results

4.1 Convergence studies

To demonstrate that the proposed algorithm is second-order accurate, we consider the displacement of water by a gaseous phase. The densities, ρ_i and viscosities, μ_i of water ($i = 1$) and gas ($i = 2$) are

$$\begin{aligned} \rho_1 &= 1000 \text{ kgm}^{-3}, & \mu_1 &= 1 \text{ cP}; \\ \rho_2 &= 2.33 + 10^{-5} p \text{ kgm}^{-3}, & \mu_2 &= 0.0178 \text{ cP}. \end{aligned}$$

Here water is incompressible but the gas is highly compressible. The 2D simulation domain is 16 m × 16 m, with no-flow boundary conditions on the top and bottom boundaries, an inflow boundary condition on the left boundary and an outflow boundary condition on the right boundary. The permeability of the porous medium is 1 D and the porosity is 0.03. Gravity is ignored. The initial distribution of the water saturation follows $0.5(1 + \cos(\pi x/16)$, which is independent of y. A pressure difference of 1 atm is applied in the x-direction. The problem then reduces to a 1D problem.

We first define the volume-averaged L_2 norm of a finite volume solution u^h as

$$\|u^h\|_{L_2} = \sqrt{\frac{\sum_i |u_i^h|^2 v_i^h}{\sum_i v_i^h}}, \tag{36}$$

where v_i^h is the volume of a single cell, and h is the mesh spacing, assuming the cells have unit aspect ratio. Given discrete solutions u^h at three grids with spacing h, $0.5h$ and $0.25h$, the convergence rate can then be approximated by

$$r_{\delta u^h} = \frac{\log(\|\delta u^h\|_{L_2} / \|\delta u^{0.5h}\|_{L_2})}{\log(2)}, \tag{37}$$

where

$$\delta u^h = u^{0.5h} - u_i^h. \tag{38}$$

The coarsest grid size used in this convergence study is $h = 1$ m, with a corresponding Δt of 0.05 s. When we reduce h by half, Δt is halved as well. Table 1 and 2 show the convergence results for a single grid with $\alpha_r = 0$ and 1 respectively, where α_r is defined in (35). We clearly see the second-order convergence in n, p and v_T in both cases.

However, the constraint (4) is only well-satisfied when $\alpha_r = 1$. Let

$$\varepsilon^h = 1 - U^h; \tag{39}$$

$\|\varepsilon^h\|_{L_2}$ provides a measure of how much our solutions deviate from (4). Tables 1 and 2 show that $\|\varepsilon^h\|_{L_2}$ for $\alpha_r = 1$ is 5 order of magnitude smaller than $\|\varepsilon^h\|_{L_2}$ for $\alpha_r = 0$. In addition, the error has second order convergence rate when $\alpha_r = 1$. When $\alpha_r = 0$, the convergence rate is only first order. The significance of the relaxation term is well-illustrated by Figure 1. When $\alpha_r = 0$, ε^h increases linearly with time. However, when $\alpha_r = 1$, ε^h decreases rapidly to machine precision.

Similar convergence behaviors are observed on the adaptive mesh. We introduce one level of refinement to the base grid we used in the single-grid convergence analysis; the finer grid covers 37.5% of the domain along the inflow boundary. Table 3 shows that we maintain the second-order convergence in $\mathbf{n}$, p and v_T. In addition, the error ε is small and converges to machine precision as $h \to 0$.

Table 2 L^2 errors and convergence rates for $\mathbf{n}$, p and v_T; $\alpha_r = 1$, $\Delta p = 1$atm, and $\mu = 0.178$.

h, m	$\|\delta n_2^h\|_{L_2}$	$r_{\delta n_2^h}$	$\|\delta p^h\|_{L_2}$	$r_{\delta p^h}$	$\|\delta v_T^h\|_{L_2}$	$r_{\delta v_T^h}$	$\|\varepsilon^h\|_{L_2}$	r_{ε^h}
1	1.34e-3	–	1.81e-3	–	7.34e-8	–	4.67e-11	–
0.5	3.35e-4	2.00	4.52e-4	2.00	1.77e-8	2.05	6.63e-12	2.81
0.25	8.37e-5	2.00	1.12e-4	2.01	4.38e-9	2.01	1.17e-12	2.50
0.125	2.09e-5	2.00	2.78e-5	2.01	1.09e-9	2.01	2.55e-13	2.20
0.0625	5.24e-6	2.00	6.79e-6	2.03	2.75e-10	1.99	6.07e-14	2.07

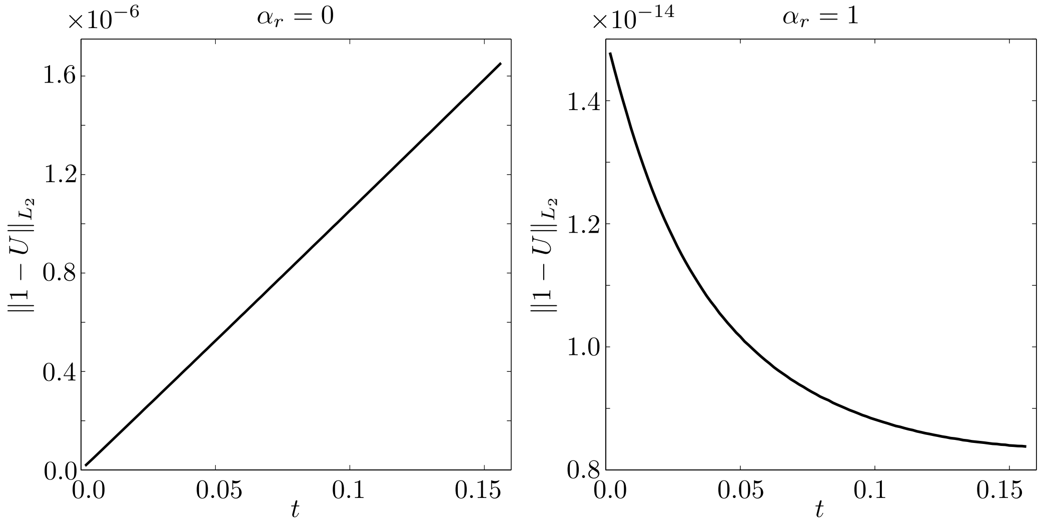


Fig. 1 When $\alpha_r = 0$, $\|1-U\|_{L_2}$ increases over time. When $\alpha_r = 1$, $\|1-U\|_{L_2}$ decreases rapidly to machine tolerance.

Table 3 L^2 errors and convergences rate for $\mathbf{n}$, p and v_T; $\alpha_r = 1$, 2-level of grids.

h, m	$\|\delta n_2^h\|_{L_2}$	$r_{\delta n_2^h}$	$\|\delta p^h\|_{L_2}$	$r_{\delta p^h}$	$\|\delta v_T^h\|_{L_2}$	$r_{\delta v_T^h}$	$\|\varepsilon^h\|_{L_2}$	r_{ε^h}
1	9.58e-4	–	1.87e-3	–	7.69e-8	–	4.67e-11	–
0.5	2.39e-4	2.00	4.67e-4	2.00	1.86e-8	2.05	6.63e-12	2.82
0.25	5.98e-5	2.00	1.16e-4	2.01	4.60e-9	2.02	1.17e-12	2.50
0.125	1.50e-5	2.00	2.87e-5	2.02	1.15e-9	2.00	2.57e-13	2.19
0.0625	3.75e-6	2.00	7.02e-6	2.03	2.93e-10	1.97	6.14e-14	2.07

In our second convergence study, we demonstrate that the proposed method is capable of resolving a discontinuity along the saturation front. For this purpose, we change the initial saturation profile. The domain is now saturated with water, and gas is injected from the left boundary. Figure 2 shows the solutions for n_1 and n_2 after 5000 s at different resolutions. The current discretization approach captures the discontinuity accurately with minimal numerical dispersion.

4.2 Uniform grid versus AMR grid

Here, we will compare results obtained on an uniform grid and an AMR grid. The setup in Section 4.1 is used here. The domain is initially saturated with water. In addition, we consider the effects of gravity and a heterogeneous permeability distribution shown in the left image of Figure 3. The mean of the permeability distribution is 1D. We note that due to the large mobility ratio, the flow is unstable. With uniform permeability, finite numerical errors in the geometric multigrid solver will induce the fingering behavior, resulting in finger patterns that depend on the grid structure. By using a common heterogeneous permeability field, the fingering patterns are then dictated by the prescribed permeability field and not by the numerical errors. This allows us to compare solutions obtained on different grid structures. In Figure 3, we compare the density of gas determined on a uniform grid with a grid size of 0.03125 m and an AMR grid with two levels of refinement where the finest grid size is also 0.03125 m. No discernible difference between the two solutions can be observed. The permeability is defined by cell-centered values at the finest level. At coarser

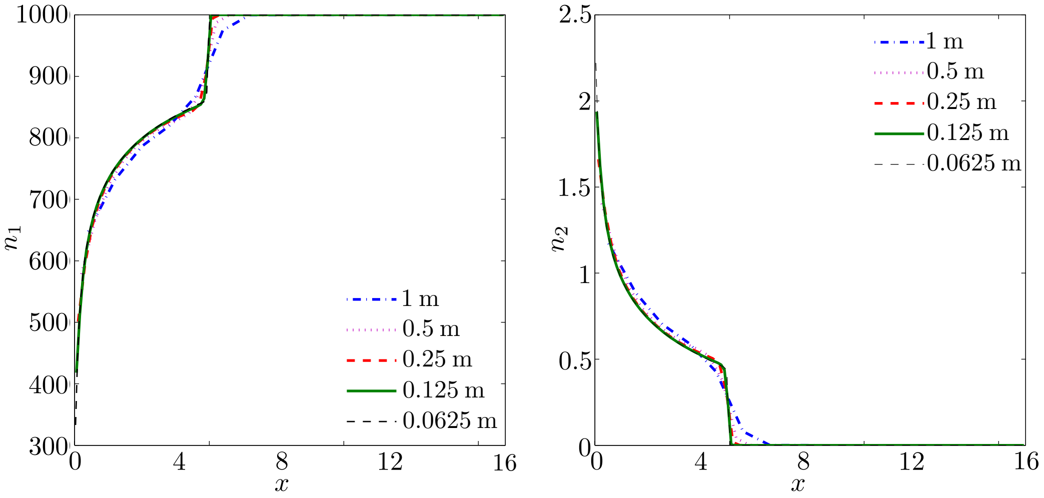


Fig. 2 Convergence of the solutions with discontinuities. We note that solutions for $h = 0.125$ m are very close to solutions for $h = 0.0625$ and are thus not discernible from the figure.

levels, the cell-centered values are given by the arithmetic mean of the values at the finest level and the edge-based values are obtained from the arithmetic mean of the edge-based values at the finest level, which are geometric means of cell-centered values adjacent to that edge.

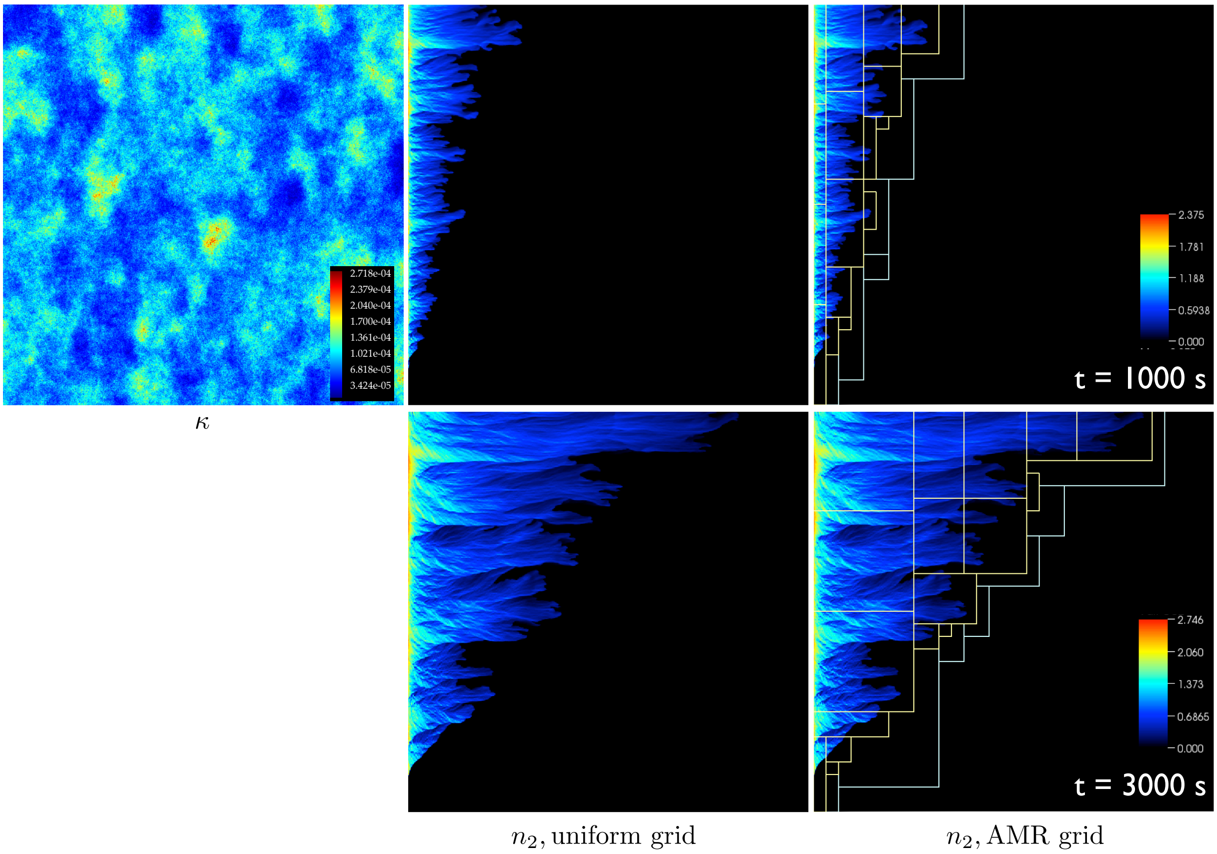
The computational savings achievable by AMR is problem-dependent. An ideal problem that takes full advantage of the AMR approach requires only high resolution in small region of the domain. The computational savings decreases when the fraction of the domain covered by the finest level of grid increases. In the current problem, the refinement criteria for the AMR grid is based on the gradient of $\mathbf{n}$, and thus the percentage of refined region grows over time as more water is being displaced by gas. At $t = 1000$ s, the finest level covers 15.1% of the domain. This grows to 39.1% at $t = 3000$ s. The percentage reduction in the computational cost, measured relative to computational cost of using the uniform grid, decreases from 79% to 62.3%. We note that the timing results are based on simulations performed on 2 CPUs. Admittedly, reduction in the computational cost is not large for this small problem, but we expect AMR techniques to be advantageous in more realistic applications in subsurface flow where the region of refinement is relatively small compared to the entire domain.

4.3 A Density-driven problem

For a more complex example that demonstrates the features in the algorithm, we consider a simplified model for the leakage of gas from an underground LPG storage cavern as described in [27]. LPG storage caverns are commonly used to store natural gas since they are considered safer and more economical than above-ground storage tanks. Theses caverns are usually located below the groundwater table. The natural gas in the caverns forms both liquid and vapor phases. Since these caverns are usually unlined, containment of gas is achieved by keeping water pressure outside the cavern higher than the liquid pressure inside the cavern. This can be achieved if the cavern is sufficiently deep or if a water curtain system is installed.

We model the leakage from the top of the storage cavern when the gas-containment criteria are not fulfilled. We limit the region of interest to a small domain surrounding the leak. The setup of the physical model is shown in Figure 4. The simulation domain has width W and height H, and is initially saturated with water. At the top boundary, we have an outflow boundary condition, and specify atmospheric pressure. At the center of the bottom boundary, there is an opening of length ℓ that allows gas to leak into the simulation domain and water to drain from the domain. The rest of the bottom boundary, and the left and right boundaries have no-flow boundary conditions.

In this example, we have $W = 64$ m, $H = 192$ m and $\ell = 10$ m. The porosity is $\phi = 0.03$ and the permeability is initially 1×10^{-14} m^2 (10 mD). Initially, quadratic relative permeability function is used and capillary effects are ignored. The properties of water and gas are given by (36). In the following sections, we examine how the solutions vary with grid resolutions and permeability distributions. We also examine the effects of using van Genuchten-Mualem relative permeability functions.



κ

n_2, uniform grid

n_2, AMR grid

Fig. 3 Top left image shows the permeability distribution used. Center and right images show the solutions to n_2 after 1000s and 3000s for a uniform grid of resolution 0.03125 m and an AMR grid with a finest resolution of 0.03125 m.

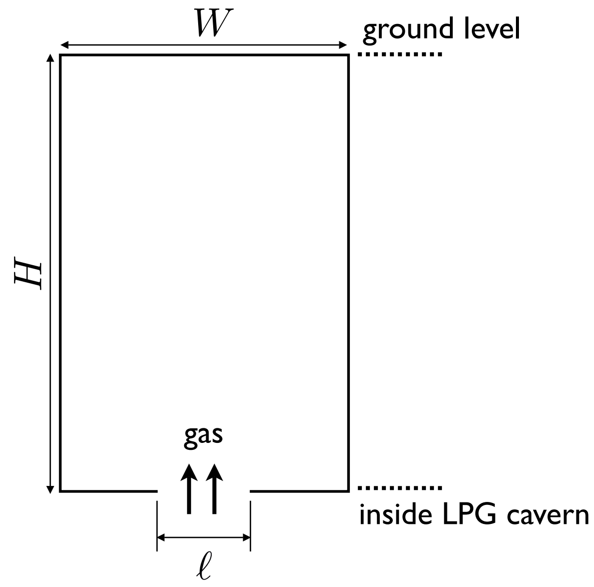


Fig. 4 Problem setup for the leakage of gas from an underground LPG storage cavern.

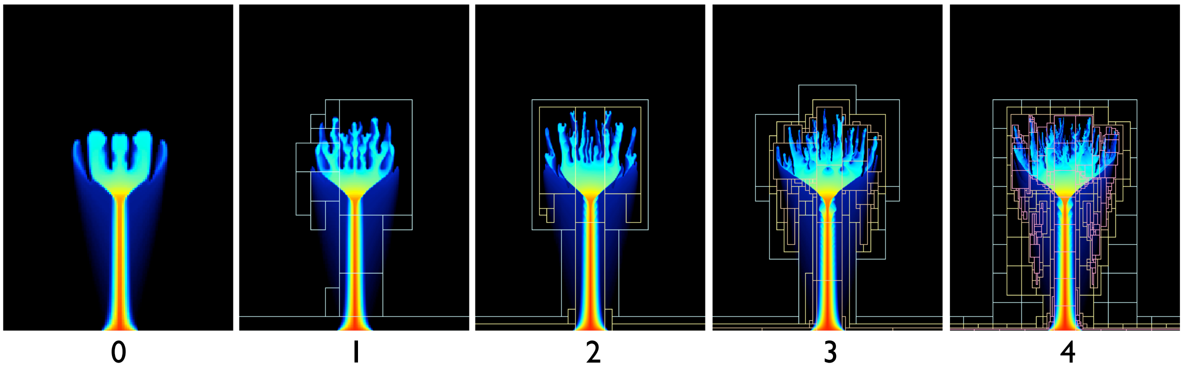


Fig. 5 Concentration of gas after 4e6 s for different level of refinements with a refinement ratio of 2. The effective resolutions are 0.5 m, 0.25 m, 0.125 m, 0.0625 m and 0.03125 m from left to right.

4.3.1 Grid resolution

Similar to the problem in Section 4.2, we have gas displacing the water from the simulation domain although the process is now driven by buoyancy forces. Fingering phenomena will occur even with a uniform permeability distribution, as explained in Section 4.2. In this figure, we start with a base grid of $h = 0.5$m and successively add layers of grid, up to 4 levels of grids; the refinement ratio between successive layers of grid is 2. Since the grid structure is different when different levels of refinement are used, the fingering structures differ from one another. However, the AMR grid dynamically adapts to ensure features of the flow are fully captured. Figure 5 also serves to show that the algorithm can handle multiple levels of refinement.

4.3.2 Solutions for different permeability distributions

The first set of permeability distributions we look at consists of an uniform distribution of 10 mD, and a heterogeneous distribution with a mean of 10 mD and a relative perturbation of 0.3. Figure 6(a) shows the solution at time 8×10^6 s. The overall behavior of the solutions are similar, indicating that the behavior is driven by macroscopic feature of the flow. However, the instabilities along the gas column is more pronounced in the heterogeneous case. Here, the AMR grids allow us to capture details of these instabilities.

In the second example, we assume the permeability abruptly changes from 10 mD to 1 D at $y = 24$m. We again compare the homogeneous case to the heterogeneous case (a random distribution with a relative perturbation of 0.3). As shown in Figure 6(b), our method captures the rapid change in the speed at which the gas moves upward. This is achieved by accurately predicting the eigenvalues of the hyperbolic system, allowing the algorithm to determine the timestep needed to capture this transition accurately without leading to instability. This leads to a well-resolved transition zone along the interface where the permeability changes. In this example, the heterogeneous permeability leads to a shorter breakthrough time due to existence of a higher permeability pathway that is accentuated by the high permeability layer.

4.3.3 Effects of relative permeability function

Instead of the quadratic relative permeability function used in previous sections, we look at the results of using the van Genuchten-Mualem relative permeability functions given by

$$k_{r,1} = \sqrt{s_1}\left(1 - \left(1 - s_1^{1/m}\right)^m\right)^2, \quad k_{r,2} = (1 - s_1)^{1/3}\left(1 - s_1^{1/m}\right)^{2m}. \tag{40}$$

The results obtained are significantly different from cases that use quadratic relative permeability function, as shown in Figure 7. In the case of uniform permeability of 10mD, the gas propagates upward at a faster rate; the gas in Figure 7(a) reaches almost the same point in space as the solution in Figure 6(a) in 1.5 $\times 10^6$ s, which is more than 5 times faster. In addition, a random relative perturbation of 0.3 in the permeability distribution leads to a even greater rate of upward propagation. The instability along the vertical column of gas is also less pronounced.

In the case of a layered permeability, results in Figure 6(b) and Figure 7(b) are more similar since the gas moves at a much faster rate once it reaches the layer with 1D permeability. Nevertheless, the use of van Genuchten-Mualem relative permeability functions leads to shorter breakthrough time.

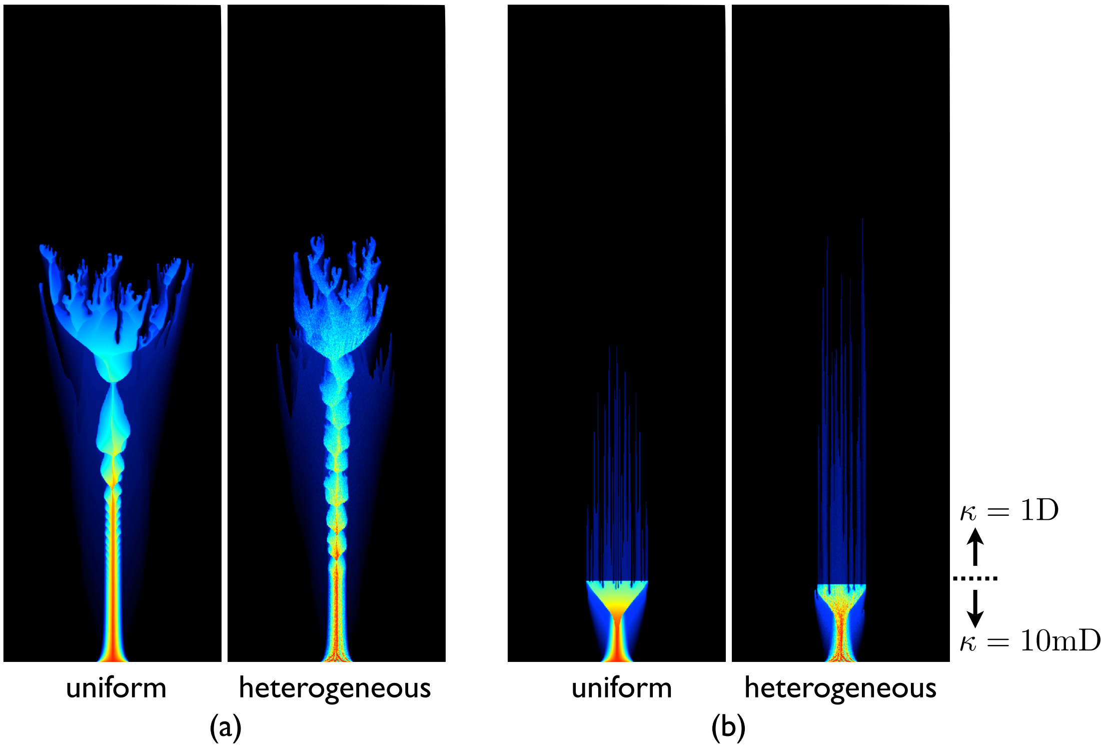


Fig. 6 Based on quadratic relative permeability function. Left: Concentration of gas after 8e6 s for a constant permeability and a permeability function with the same mean but with a relative perturbation of 0.3. Right: Concentration of gas after 2e6 s for a 2-layer permeability distribution with uniform values and one with a relative perturbation of 0.3.

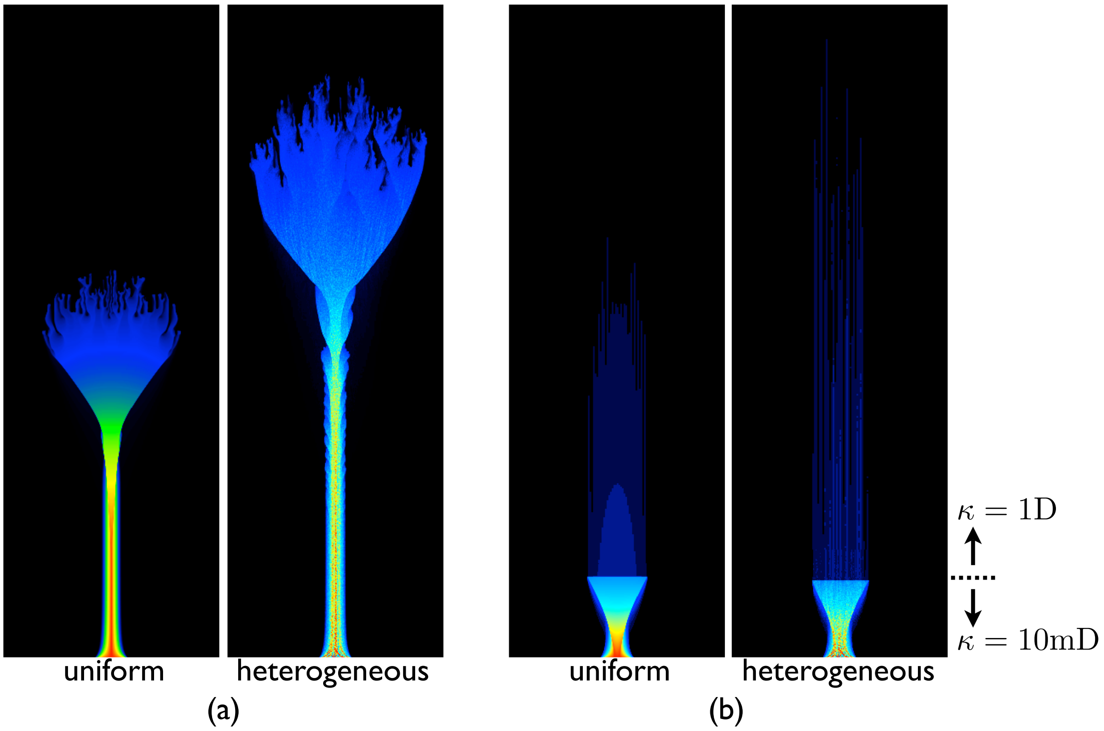


Fig. 7 Based on van Genuchten-Mualem relative permeability functions with $m = 0.5$. Left: Concentration of gas after 1.5e6 s for a constant permeability and a permeability function with the same mean but with a relative perturbation of 0.3. Right: Concentration of gas after 3.8e5 s for a 2-layer permeability distribution with uniform values and one with a relative perturbation of 0.3.

5 Conclusions and future work

In this paper, we have presented a structured AMR algorithm for compressible two-phase flows in porous media. The method is based on a total velocity splitting of the equations into a parabolic pressure equation and a system of hyperbolic conservation laws for the component densities. The resulting system is discretized using cell-centered differencing for the pressure equation and a second-order Godunov scheme for the component conservation equations. A global timestep is used and the pressure equation is solved on the composite grid. A relaxation term in the pressure equation ensures the equation of states is approximately satisfied for long time integration. The convergence results demonstrate the second-order accuracy of the approach. The AMR results also demonstrate the capability of the adaptive grid to capture complex flow behaviors of two-phase compressible flow. We intend to extend the approach to more general multiphase systems with mass transfer between phases and thermal effects in future work.

Acknowledgements We would like to thank Karsten Pruess for introducing us to the gas leakage problem. Support for this work was provided by the Applied Mathematics Research Program, the Office of Science, the Office of Basic Energy Sciences and the Advanced Simulation Capability for Environmental Management (ASCEM) program of the U.S. Department of Energy under contract DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center supported by the Office of Science of the U.S. Department of Energy under the same contract.

References

1. G. Acs, S. Doleschall, and E. Farkas. General purpose compositional model. *Society of Petroleum Engineers Journal*, 25:543–552, Aug. 1985.
2. B. Amaziane, M. Jurak, and A. Z. Keko. Modeling and numerical simulations of immiscible compressible two-phase flow in porous media by the concept of global pressure. *Transport in Porous Media*, 2009.
3. T. Arbogast, M. Wheeler, and I. Yotov. Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences. *SIAM Journal on Numerical Analysis*, 34(2):828–852, 1997.
4. M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comp. Phys.*, 82(1):64–84, May 1989.
5. M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, 53:484–512, Mar. 1984.
6. G. Chavent and J. Jaffré. *Mathematical models and finite elements for reservoir simulation: single phase, multiphase, and multicomponent flows through porous media.* North Holland, 1986.
7. Z. Chen, M. Espedal, and R. Ewing. Continuous-time finite element analysis of multiphase flow in groundwater hydrology. *Appl. Math*, 40:203–226, 1995.
8. Z. Chen, G. Huan, and Y. Ma. *Computational methods for multiphase flows in porous media.* Society for Industrial Mathematics, 2006.
9. J. Christensen, G. Darche, B. Dechelette, H. Ma, and P. Sammon. Applications of dynamic gridding to thermal simulations. Paper SPE 86969. In *SPE International Thermal Operations and Heavy Oil Symposium and Western Regional Meeting*, Bakersfield, California, March 2004.
10. M. G. Edwards. A higher-order Godunov scheme coupled with local grid refinement for flow in a porous medium. *Comput. Methods Appli. Mech. Engrg*, 131:287–308, 1996.
11. R. E. Ewing, B. A. Boyett, D. K. Babu, and R. F. Heinemann. Efficient use of refined grids for multiphase reservoir simulation. Paper SPE 18413. In *SPE Reservoir Simulation Symposium*, Houston, Texas, February 1989.
12. Z. E. Heinemann. Using local grid refinement in a multiple-application reservoir simulator. Paper SPE 12255. In *SPE Reservoir Simulation Symposium*, San Francisco, California, November 1983.
13. R. Helmig. *Multiphase Flow and Transport Processes in the Subsurface: A contribution to the modeling of hydrosystems.* Springer Berlin, Heidelberg, 1997.
14. H. M. Hoang and J. Kleppe. A parallel adaptive finite difference algorithm for reservoir simulation. Paper SPE 99572. In *SPE Europe/EAGE Annual Conference and Exhibition*, Vienna, Austria, June 2006.
15. R. D. Hornung and J. A. Trangenstein. Adaptive mesh refinement and multilevel iterations for flow in a porous media. *J. Comp. Phys.*, 136:522–545, 1997.
16. J. Nilsson, M. Gerritsen, and R. Younis. A novel adaptive anisotropic grid framework for efficient reservoir simulation. Paper SPE 93243. In *SPE Reservoir Simulation Symposium*, Woodlands, Texas, January 2005.
17. A. Nonaka, A. Almgren, J. Bell, M. Lijewski, C. Malone, and M. Zingale. MAESTRO: An Adaptive Low Mach Number Hydrodynamics Algorithm for Stellar Flows. *The Astrophysical Journal Supplement Series*, 188:358, 2010.
18. G. S. H. Pau, A. S. Almgren, J. B. Bell, and M. J. Lijewski. A parallel second-order adaptive mesh algorithm for incompressible flow in porous media. *Phil. Trans. R. Soc. A*, 367:4633–4654, 2009.
19. R. M. Propp. *Numerical Modeling of a Trickle Bed Reactor.* Ph.D., Dept. of Mechanical Engineering, Univ. of California, Berkeley, Sept. 1998.
20. C. A. Rendleman, V. E. Beckner, M. J. Lijewski, W. Y. Crutchfield, and J. B. Bell. Parallelization of structured, hierarchical adaptive mesh refinement algorithms. *Computing and Visualization in Science*, 3(3):147–157, 2000.
21. P. H. Sammon. Dynamic grid refinement and amalgamation for compositional simulation. Paper SPE 79683. In *SPE Reservoir Simulation Symposium*, Houston, Texas, February 2003.
22. J. A. Trangenstein. Multi-scale iterative techniques and adaptive mesh refinement for flow in porous media. *Advances in Water Resources*, 25:1175–1213, 2002.

23. J. A. Trangenstein and J. B. Bell. Mathematical structure of black-oil reservoir simulation. *SIAM J. Appl. Math.*, 49:749–783, 1989.
24. J. A. Trangenstein and J. B. Bell. Mathematical structure of compositional reservoir simulation. *SIAM J. Sci. Stat. Comput.*, 10:817–845, 1989.
25. J. A. Trangenstein and Z. Bi. Large multi-scale iterative techniques and adaptive mesh refinement for miscible displacement simulation. Paper SPE 75232. In *SPE/DOE Improved Oil Recovery Symposium*, Tulsa, Oklahoma, April 2002.
26. D. van Odyck, J. Bell, F. Monmont, and N. Nikiforakis. The mathematical structure of multiphase thermal models of flow in porous media. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 465(2102):523, 2009.
27. H. Yamamoto and K. Pruess. Numerical simulations of leakage from underground LPG storage caverns. Technical Report LBNL-56175, LBNL, 2004.