SIAM J. Sci. Comput.
Vol. 35, No. 1, pp. B149–B175

AN h-ADAPTIVE OPERATOR SPLITTING METHOD FOR TWO-PHASE FLOW IN 3D HETEROGENEOUS POROUS MEDIA[*]

CHIH-CHE CHUEH[†], NED DJILALI[‡], AND WOLFGANG BANGERTH[§]

Abstract. The simulation of multiphase flow in porous media is a ubiquitous problem in a wide variety of fields, such as fuel cell modeling, oil reservoir simulation, magma dynamics, and tumor modeling. However, it is computationally expensive. This paper presents an interconnected set of algorithms which we show can accelerate computations by more than two orders of magnitude compared to traditional techniques, yet retains the high accuracy necessary for practical applications. Specifically, we base our approach on a new adaptive operator splitting technique driven by an a posteriori criterion to separate the flow from the transport equations, adaptive meshing to reduce the size of the discretized problem, efficient block preconditioned solver techniques for fast solution of the discrete equations, and a recently developed artificial diffusion strategy to stabilize the numerical solution of the transport equation. We demonstrate the accuracy and efficiency of our approach using numerical experiments in one, two, and three dimensions using a program that is made available as part of a large open source library.

Key words. adaptive mesh refinement, stabilized finite element method, operator splitting, preconditioning, two-phase flow, heterogeneous porous media, fuel cells

AMS subject classifications. 65M22, 65M50, 65F08

DOI. 10.1137/120866208

1. Introduction.

Multiphase flow models in porous media are used in a wide variety of fields, such as oil reservoirs [4, 38, 39, 60], the flow of magma in the earth's crust [44, 58], and transport processes in fuel cells [53, 59]. To illustrate the importance of such models, let us consider the example of fuel cells. There, current interest is primarily focused on polymer electrolyte membrane (PEM) fuel cells in which the gas diffusion layer (GDL) provides pathways for gaseous fuel to reach the catalyst sites, for electrons produced electrochemically at the catalyst sites [27] to be conducted to the current collector, and, very importantly, for excess liquid water to exit the system. Excess water can curtail transport of reactant gases [11] as well as exacerbate degradation [77]. Improvements in PEM fuel cells are therefore contingent upon advancements in water management, which crucially relies on the simulation of two-phase transport phenomena in three-dimensional (3D) heterogeneous porous media using realistic, complex flow models.

[*] Submitted to the journal's Computational Methods in Science and Engineering section February 15, 2012; accepted for publication (in revised form) October 12, 2012; published electronically January 29, 2013. The work of the first and second authors was supported by the Canada Research Chairs Program and the MITACS Network of Centres of Excellence. Some computational resources in this work were supported by NSF award DMS-0922866.

http://www.siam.org/journals/sisc/35-1/86620.html

[†] Institute for Integrated Energy Systems and Department of Mechanical Engineering, University of Victoria, P.O. Box 3055 STN CSC, Victoria, BC, V8W 3P6, Canada. Current address: Queen's-RMC Fuel Cell Research Centre and Department of Mechanical and Materials Engineering, Queen's University, Kingston, ON, K7L 3N6, Canada (chih-che.chueh@queensu.ca).

[‡] Corresponding author. Institute for Integrated Energy Systems and Department of Mechanical Engineering, University of Victoria, P.O. Box 3055 STN CSC, Victoria, BC, V8W 3P6, Canada (ndjilali@uvic.ca).

[§] Corresponding author. Department of Mathematics, Texas A&M University, College Station, TX 77843-3368 (bangerth@math.tamu.edu). This author's work was supported by award KUS-C1-016-04, made by the King Abdullah University of Science and Technology, by the Computational Infrastructure in Geodynamics initiative through the NSF under award EAR-0949446 and The University of California–Davis, and through an Alfred P. Sloan Research Fellowship.

A number of recent publications have dealt with some of the modeling issues associated with multiphase flow in porous media. These include the development of improved numerical schemes for simulations of multidimensional wave-oriented upwind schemes [33], an exponential integrator for advection-dominated reactive flow [73], and an unconditionally convergent nonlinear solver for hyperbolic conservation laws [45]. An open-source MATLAB implementation offering a flexible discretization which can be used in more complex structures has also been developed [56]. A good review of recent efforts to develop either semianalytical solutions for one-dimensional (1D) verification or novel numerical schemes for solving the governing equations efficiently is provided by Geiger, Schmid, and Zaretskiy [38].

For PEM fuel cells as well as other applications involving more than one phase in porous media, the bottleneck for simulations is the speed with which complex 3D problems can be solved. In this regard, classical finite element and finite volume methods—while having the potential to accurately represent all the salient physics—do not always compare favorably against other methods such as pore network modeling [9, 12, 57, 62, 63] or the lattice Boltzmann method (LBM) [72, 84]. On the other hand, finite element methods are well understood and integrate easily into many other parts of the engineering workflow, unlike some of the other methods listed above. Increasing the computational efficiency of the finite element/volume numerical methods while retaining their accuracy therefore remains a central challenge.

There are at least five areas which a high performance simulator needs to address to improve computational speed for a given level of accuracy:

- Higher order spatial discretizations that can yield the same accuracy at smaller computational cost, but need to incorporate nontrivial stabilization mechanisms for hyperbolic problems (such as those representing multiphase flow) to benefit from the higher accuracy.
- Adaptive mesh refinement that can vastly reduce the number of cells required to resolve the flow field.
- Adaptive time stepping methods that allow the use of large time steps limited solely by the physical time scale rather than numerical stability.
- Operator splitting methods for coupled problems to transform a complex, coupled problem into a sequence of simpler problems for which more efficient solver techniques are available.
- Efficient solver and preconditioning methods that can accelerate the solution of the linear problems.

In this paper, we discuss a framework that addresses all of these issues and demonstrate that, with judicious choice of algorithms, we can accelerate the numerical simulation of multiphase flow problems by more than two orders of magnitude compared to widely used, traditional methods, making possible the simulation of previously inaccessible scenarios. At the core of these algorithms is the development of a new criterion for operator splitting and the adaptation of methods from other fields to accelerate the solution of linear systems and to stabilize the transport equations. The implementation of these methods is described in detail, and the associated code is made available as part of the widely used open source finite element library `deal.II` [6, 7] through the extensively documented tutorial program Step-43 [20]. We note that for more complex models than the ones considered here, an additional point that needs to be thoroughly addressed is the design of nonlinear solvers (see, for example, [45]), but we will not consider this here.

To place the proposed techniques into their proper context, in the remainder of this introduction, we summarize the main contributions of this paper and provide an overview of the related literature. The mathematical formulation and algorithm description are discussed in the subsequent sections.

Higher order discretizations and stabilization issues. Stability issues arising from the discretization of advection-dominated problems have long led to the belief that finite volume or upwinding schemes [79] (or their modern incarnation in the form of discontinuous Galerkin (DG) finite element methods [17]) combined with the usual set of flux/slope limiters are the only choice; in other words, upwinding schemes should be preferred over the introduction of artificial diffusion [49]. The primary reason for this belief is that diffusive stabilization often excessively smoothes out sharp fronts [42, 82].

On the other hand, it is not completely straightforward to derive upwind schemes of higher order (see, however, [13, 33]). Higher order methods are attractive because—up to a certain point—they typically provide the same accuracy with fewer degrees of freedom and less numerical work. Consequently, the development of stabilization methods for higher order methods applied to transport dominated problems is an important step to achieving higher computational efficiency. We have previously demonstrated an artificial diffusion method that, when used with a general continuous finite element discretization for the saturation transport (advection) equation in two-phase flow in porous media, ensures stability and accuracy of the solution [21]. The method uses an entropy-based diffusion term, proposed by Guermond and Pasquetti [41], and is able to efficiently damp unphysical oscillations while providing the same or better resolution for the saturation field as typical low-order DG methods without upwinding [55]. The trick is to ensure that the artificial viscosity term acts only in the vicinity of strong gradients in the saturation and other discontinuities, while disabling it in regions where the solution is smooth. The scheme offers higher order accuracy at least in smooth regions while providing stability where necessary [41].

Adaptive mesh refinement. In many engineering problems involving fluid dynamics, structural mechanics, etc., automated adaptive mesh refinement (AMR) has been used to obtain numerical solutions with higher accuracy and resolution, while requiring less memory and shorter computational times. Adaptivity, which can be traced back to the late 1970s [5], is based on the idea that in order to improve numerical accuracy, a fine mesh is not necessarily required everywhere but only in areas where the solution varies significantly. A variety of AMR methods have been proposed depending on the type of physical problem and associated partial differential equations, and a large body of literature exists for these methods (see, for example, [2, 8, 16, 80] for a general overview, and [34, 43] for overviews in multiphase flow). Adaptive methods can be grouped into a number of categories: h-adaptive methods change the local element size h in response to smoothness properties of the solution; p-adaptive methods vary the local polynomial order p of the finite element space; hp-adaptive methods combine the previous two methods; r-adaptive methods move mesh points [24, 25, 26, 54, 74, 75, 76, 87]; and subgrid methods superimpose finer meshes over the original grid where necessary [10]. Despite this variety, h-adaptive methods are most commonly used in engineering applications today. We note that the literature on the use of adaptive meshes remains relatively limited for *transient* multiphase flow in porous media.

Adaptivity is driven by refinement indicators that specify which cells of the mesh should be refined, which should be coarsened, and which should remain as they are. These indicators may be based on error estimates for the underlying partial differential equation [2, 8, 80]; however, for many applications, simpler error *indicators* based on the smoothness of the solution, such as those proposed in [37, 49], are often sufficient. In particular, for hyperbolic problems, most refinement criteria will yield very similar meshes that refine sharp fronts as they evolve in space and time, while reducing computational cost by keeping the mesh coarse in areas where fine meshes are not required. For the purpose of this work, we use a relatively simple but effective refinement criterion that is based on the gradient of the saturation.

Operator splitting techniques. Multiphase porous media flow is a coupled problem in which the flow and pressure fields affect the transport of individual phases, but the phase composition also nonlinearly affects the flow and pressure fields. Such couplings significantly complicate the numerical solution. Operator splitting methods, first introduced by Douglas, Peaceman, and Rachford [28, 29, 30, 31, 61] and in the form of fractional step methods by D'yakonov [32] and Yanenko [85], alleviate the problem by breaking up the coupled problem into a sequence of simpler problems whose solution can, if necessary, be iterated until sufficient accuracy is reached.

In the context of nonreactive systems of multiphase flow in porous media, two kinds of operator splitting are typically used: splitting the pressure-velocity calculation from the saturation calculation (see, for example, [39, 69, 71]), and splitting the saturation calculation into convection and diffusion components [23, 35, 36, 46]. We will here focus on the first variant, leading to time stepping schemes that are most commonly referred to as IMPES (implicit pressure, explicit saturation). In these schemes, one first solves the implicit, time independent pressure/velocity system with the current saturation values, and then uses an explicit time stepping scheme to advance the saturation by one time step.[1] While this effectively decouples the two equations, one is left with the important problem that solving the equations for flow velocity and pressure is much more expensive than solving that for the saturation, and vastly dominates the overall computational effort. On the other hand, the effect of the saturation on the flow field is typically weak; i.e., the flow field evolves only slowly, while the saturation changes significantly at each time step.[2] Consequently, we will here explore the possibility of not solving the computationally expensive flow equations at every time step but only when necessary, thereby significantly reducing computing time.

Such methods have previously been developed (see, for example, [1, 18, 19]) but with a fixed number of saturation time steps between each solution of the flow field. We will here make the timing of solving the flow equations adaptive using a new a posteriori criterion (see Theorem 3.1 below) relating the change in the velocity to the change in the saturation since the flow equations were solved last.

Solver and preconditioner techniques. The result of operator splitting methods is a sequence of linear systems—frequently ill-conditioned, nonsymmetric, and/or indefinite—that need to be solved efficiently. Their size typically leaves Krylov subspace methods as the only viable choice, hence requiring good preconditioners for efficient solution. In the context of the mixed formulation of porous media flow, the

[1] An alternative are IMPIS schemes, in which the saturation equation is solved implicitly as well, thereby avoiding the Courant–Friedrichs–Levy condition on the time step size. The observations of this article largely remain valid for this variation on the splitting algorithm.

[2] This is true because in the model we will introduce below, the saturation only enters the equation defining the velocity as a coefficient in an elliptic operator; see (2.1). On the other hand, for models that include capillary forces and gravity, the saturation enters the velocity equation as a source term, and one would need to expect a more immediate impact of changes in the saturation on the velocity field.

saddle point structure of the flow problem presents a particular complication. An overview of solvers can be found in [3, 64, 65, 67]. We will here adapt a method originally proposed for the Stokes system by Silvester and Wathen [70] that leads to very efficient preconditioners that can also be shown to easily scale to very large parallel computations [48].

Overview of the paper. The remainder of the paper presents the mathematical formulation of the methods outlined above and shows how they can be combined to form a method that achieves a speed-up of more than two orders of magnitude compared to a traditional two-phase porous medium solver. Section 2 presents the mathematical flow model, followed in section 3 by a discussion of the time stepping, discretization, operator splitting, and stabilization methods. We show numerical results in section 4. We conclude in section 5.

As mentioned above, the code in which our methods are implemented is available as part of the open source library `deal.II` [7] in the form of the extensively documented Step-43 tutorial program [20]. It is meant as the basis for further experiments by others. We note that while we only show examples below that use rectangular grids, nothing in the design of Step-43 prevents the use of `deal.II`'s facilities for unstructured meshes approximating complex geometries.

2. Mathematical model of two-phase porous media flow.

We consider the flow of a two-phase immiscible, incompressible fluid. Capillary and gravity effects are neglected, and viscous effects are assumed dominant. The governing equations for such a flow are then [39]

$$\mathbf{u}_t = -\mathbf{K}\lambda_t\left(S\right)\nabla p, \tag{2.1}$$

$$\nabla \cdot \mathbf{u}_t = q, \tag{2.2}$$

$$\epsilon\frac{\partial S}{\partial t} + \nabla \cdot \left(\mathbf{u}_t F\left(S\right)\right) = 0, \tag{2.3}$$

where S is the saturation (volume fraction) of the second (wetting) phase, p is the pressure, $\mathbf{K}$ is the permeability tensor, $\lambda_t(S)$ is the total mobility, ϵ is the porosity, $F(S)$ is the fractional flow of the wetting phase, q is the source term, and $\mathbf{u}_t$ is the total velocity. The exact dependency of F, λ_t on S rests on the chosen media parameterization; we will give concrete examples in section 4 when presenting numerical results. If the porosity ϵ in (2.3) is constant, it can be considered a scaling factor for the time variable, and could be omitted. However, we will continue to carry this coefficient for generality.

Equations (2.1)–(2.3) are augmented by initial conditions for the saturation and boundary conditions for the pressure. Since the flow equations do not contain time derivatives, initial conditions for the velocity and pressure variables are not required. The flow field separates the boundary into inflow or outflow parts. Specifically,

$$\mathbf{\Gamma}_{in}(t) = \left\{\mathbf{x} \in \partial\Omega : \mathbf{n} \cdot \mathbf{u}_t < 0\right\}, \tag{2.4}$$

and we arrive at a complete model by also imposing boundary values for the saturation variable on the inflow boundary $\mathbf{\Gamma}_{in}$.

3. Numerical methods.

As noted in the introduction, the numerical solution of (2.1)–(2.3) is computationally costly. In the following subsections, we will discuss the numerical methods we use to solve them efficiently and accurately.

3.1. Adaptive operator splitting and time stepping. The time stepping schemes most commonly used to solve equations of the kind (2.1)–(2.3) are of IMPES type in which one first solves the implicit pressure/velocity system (2.1)–(2.2) with the current saturation values, and then uses an explicit time stepping scheme to advance the saturation by one time step using (2.3). In IMPES schemes, the vast majority of computing time is spent in the implicit solver for the pressure and velocity variables. Computing efforts may be significantly reduced by noting that the pressure and velocity fields depend only weakly on the saturation, and therefore do not change significantly between time steps, whereas saturation fronts typically move by one cell in each step. We here propose a scheme that solves for the saturation at every time step, and only updates the velocity and pressure whenever necessary (we will call the intervals between such updates "macro time steps"). During saturation time steps in which we do not solve for the velocity, we find a velocity field by extrapolation from the previous two available velocity solutions. A similar method is described by Abreu et al. in [1] where the pressure system is only solved once per fixed number of saturation time steps. A better approach would let the length of the macro time steps depend *adaptively* on the changes incurred in the saturation since the last update.

To derive such a scheme, let superscripts in parentheses denote the number of the time step at which a quantity is defined. Furthermore, let $n^{(k)}$ be the number of the time step in which we computed the velocity and pressure variables for the kth time, and consider a time step n so that $n^{(k)} < n < n^{(k+1)}$. For the coupling in the saturation equation, we only need the velocity $\mathbf{u}_t^{(n)}$; however, we do not want to compute $\mathbf{u}_t^{(n)}$ but rather use a quantity already computed. Consequently, we need a criterion that tells us when this approximation becomes inadequate; i.e., we need to estimate $\|\mathbf{u}_t^{(n)} - \mathbf{u}_t^{(n^{(k)})}\|$ without actually computing $\mathbf{u}_t^{(n)}$.

We will not attempt to derive such an estimate at the level of the original partial differential equation. Rather, let us assume (as discussed further down below) that we have discretized the velocity-pressure equations with one of the usual finite element methods. Then, if we were to solve for the discretized pressure and velocity in time step n, they would have to satisfy the system of equations

$$\begin{pmatrix} \mathbf{M}^{\mathbf{u}}(S^{(n-1)}) & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}^{(n)} \\ \mathbf{P}^{(n)} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{F}_2^{(n)} \end{pmatrix}, \tag{3.1}$$

where $\mathbf{B}, \mathbf{B}^T$ correspond to the operators $-\mathrm{div}$ and ∇, respectively, and $\mathbf{M}^u(S)$ corresponds to $\mathbf{K}^{-1}\lambda_t(S)^{-1}$. Obviously, $\mathbf{U}^{(n^{(k)})}, \mathbf{P}^{(n^{(k)})}$ have to satisfy a corresponding set of equations. Then we have the following theorem.

THEOREM 3.1. *Let $\mathbf{U}^{(n^{(k)})}, \mathbf{P}^{(n^{(k)})}$ and $\mathbf{U}^{(n)}, \mathbf{P}^{(n)}$ be solutions to* (3.1) *at time steps $n^{(k)}$ and n, respectively, and assume that the source terms and pressure boundary conditions are constant in time, i.e., $\mathbf{F}_2^{(n^{(k)})} = \mathbf{F}_2^{(n)} = \mathbf{F}_2$. Furthermore, assume* (i) *that the mesh $\mathbb{T}$ consisting of cells κ is shape regular,* (ii) *that the finite element space results in a stable discretization,* (iii) *that $[\mathbf{K}\lambda_t(S)]^{-1}$ is bounded for the range of saturations S that appear in the solution, and* (iv) *that the source term $q(\mathbf{x})$ is bounded and consequently $\|\mathbf{F}\| < \infty$.*

Then there exist constants C_1, C_∞ such that

$$\left\|\mathbf{U}^{(n)} - \mathbf{U}^{(n^{(k)})}\right\|_1 \le C_1 \max_{\kappa\in\mathbb{T}} |\kappa| \left\| \frac{1}{\lambda_t\left(S^{(n-1)}\right)} - \frac{1}{\lambda_t\left(S^{(n^{(k)}-1)}\right)} \right\|_{L^\infty(\kappa)} \left\| \|\mathbf{K}^{-1}\|_1 \right\|_{L^\infty(\kappa)}, \tag{3.2}$$

$$\left\|\mathbf{U}^{(n)} - \mathbf{U}^{(n^{(k)})}\right\|_\infty \le C_\infty \max_{\kappa\in\mathbb{T}} |\kappa| \left\| \frac{1}{\lambda_t\left(S^{(n-1)}\right)} - \frac{1}{\lambda_t\left(S^{(n^{(k)}-1)}\right)} \right\|_{L^\infty(\kappa)} \left\| \|\mathbf{K}^{-1}\|_\infty \right\|_{L^\infty(\kappa)}. \tag{3.3}$$

In other words, the theorem provides a way to estimate, up to a constant, how far the unknown velocity $\mathbf{u}_t^{(n)}$ has deviated from the previously computed $\mathbf{u}_t^{(n^{(k)})}$, using only the saturations computed at the two time steps involved. We will construct a concrete indicator from this below.

Proof. Let

$$A(S) = \begin{pmatrix} \mathbf{M}^{\mathbf{u}}(S) & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix}, \qquad \mathbf{V} = \begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix}, \qquad \mathbf{G} = \begin{pmatrix} 0 \\ \mathbf{F}_2 \end{pmatrix}.$$

Under the assumptions of the theorem, $A(S)$ is an invertible matrix and the norm of $\mathbf{G}$ is bounded. Consequently, $\mathbf{V}^{(n)} = A(S^{(n-1)})^{-1}\mathbf{G}$ and $\mathbf{V}^{(n^{(k)})} = A(S^{(n^{(k)}-1)})^{-1}\mathbf{G}$ and we have for any vector norm and associated matrix norm

$$\begin{aligned}
\left\|\mathbf{U}^{(n)} - \mathbf{U}^{(n^{(k)})}\right\| &\le \left\|\mathbf{V}^{(n)} - \mathbf{V}^{(n^{(k)})}\right\| \\
&= \left\| \left[A(S^{(n-1)})^{-1} - A(S^{(n^{(k)}-1)})^{-1}\right]\mathbf{G}\right\| \\
&\le \left\| A(S^{(n-1)})^{-1} - A(S^{(n^{(k)}-1)})^{-1}\right\| \|\mathbf{G}\| \\
&= \left\| A(S^{(n-1)})^{-1}\left[A(S^{(n^{(k)}-1)}) - A(S^{(n-1)})\right] A(S^{(n^{(k)}-1)})^{-1}\right\| \|\mathbf{G}\| \\
&\le \left\| A(S^{(n-1)})^{-1}\right\| \left\| A(S^{(n^{(k)}-1)})^{-1}\right\| \left\| A(S^{(n^{(k)}-1)}) - A(S^{(n-1)})\right\| \|\mathbf{G}\|.
\end{aligned}$$

Next observe that among the components of A only M^u depends on the saturation, i.e., $\|A(S^{(n^{(k)}-1)}) - A(S^{(n-1)})\| = \|M^u(S^{(n^{(k)}-1)}) - M^u(S^{(n-1)})\|$. Thus, we get

$$\left\|\mathbf{U}^{(n)} - \mathbf{U}^{(n^{(k)})}\right\| \le C \left\|\mathbf{M}^{\mathbf{u}}(S^{(n-1)}) - \mathbf{M}^{\mathbf{u}}(S^{(n^{(k)}-1)})\right\| \tag{3.4}$$

by setting $C = \sup_S \|A(S)^{-1}\|^2 \|\mathbf{F}_2\|$ and using the assumed boundedness of $A(S)$ as a function of saturation S.

Since this holds for any vector and associated matrix norm, we may choose one that allows for the convenient evaluation of (3.4). To this end, recall that $\mathbf{M}^{\mathbf{u}}(S)_{ij} = \left((\mathbf{K}\lambda_t(S))^{-1}\mathbf{v}_i, \mathbf{v}_j\right)_\Omega$. Furthermore, for all common finite element shape functions, we have $\sum_j \mathbf{v}_j = \mathbf{e}$ with $\mathbf{e} = (1,1)^T$ in two dimensions and similar in three dimensions. Then

$$\begin{aligned}
&\left\|\mathbf{M}^{\mathbf{u}}(S^{(n-1)}) - \mathbf{M}^{\mathbf{u}}(S^{(n^{(k)}-1)})\right\|_\infty \\
&\qquad = \max_i \sum_j \left| \left(\left[\left(\mathbf{K}\lambda_t\left(S^{(n-1)}\right)\right)^{-1} - \left(\mathbf{K}\lambda_t\left(S^{(n^{(k)}-1)}\right)\right)^{-1} \right] \mathbf{v}_i, \mathbf{v}_j \right)_\Omega \right| \\
&\qquad = \max_i \int_\Omega \left| \left[\frac{1}{\lambda_t\left(S^{(n-1)}\right)} - \frac{1}{\lambda_t\left(S^{(n^{(k)}-1)}\right)} \right] \mathbf{e}^T \mathbf{K}^{-1} \mathbf{v}_i \right| \, d\Omega.
\end{aligned}$$

By our choice of finite element space, each of the shape functions $\mathbf{v}_i$ is nonzero in only one vector component. Furthermore, it is nonzero only on a patch $\omega_i \subset \Omega$ that is the union of a number of cells. Let n_v be the maximal number of cells meeting at any one vertex of the triangulation $\mathbb{T}$; then

$$\begin{aligned} &\left\|\mathbf{M}^{\mathbf{u}}(S^{(n-1)}) - \mathbf{M}^{\mathbf{u}}(S^{(n^{(k)}-1)})\right\|_{\infty} \\ &\qquad \le \max_i \int_{\omega_i} \left| \frac{1}{\lambda_t\left(S^{(n-1)}\right)} - \frac{1}{\lambda_t\left(S^{(n^{(k)}-1)}\right)} \right| \|\mathbf{K}^{-1}\|_1 \, d\omega_i \\ &\qquad \le n_v \max_{\kappa\in\mathbb{T}} |\kappa| \left\| \frac{1}{\lambda_t\left(S^{(n-1)}\right)} - \frac{1}{\lambda_t\left(S^{(n^{(k)}-1)}\right)} \right\|_{L^\infty(\kappa)} \left\| \|\mathbf{K}^{-1}\|_1 \right\|_{L^\infty(\kappa)} . \end{aligned} \tag{3.5}$$

The first statement of the theorem follows from this by using (3.4), combining the norm-dependent constant C and n_v into C_1. The second claim of the theorem is proven in exactly the same way. □

Remark 3.2. Intuitively, one would expect $\|\mathbf{U}^{(n)} - \mathbf{U}^{(n^{(k)})}\|_1$ to grow with the distance of $S^{(n-1)}$ from $S^{(n^{(k)}-1)}$ but be more or less independent of the spatial discretization. On the other hand, the previous theorem seems to indicate that the difference in velocities decreases with the size of mesh cells $\kappa \in \mathbb{T}$ since $h^d = |\kappa|$ appears on the right-hand side (with $d = \dim \Omega$). However, this latter conclusion is a fallacy; the constant C in the previous lemma grows like h^{-d}, canceling this factor. Consequently, it is indeed the change in inverse mobilities $\lambda_t^{-1}(S)$ along with the permeability that determines the change in velocities.

Following this reasoning, let us define the following indicator function derived from the first estimate of the theorem above:

$$\theta(n, n^{(k)}) = \max_{\kappa\in\mathbb{T}} \left(\left\| \frac{1}{\lambda_t\left(S^{(n-1)}\right)} - \frac{1}{\lambda_t\left(S^{(n^{(k)}-1)}\right)} \right\|_{L^\infty(\kappa)} \left\| \|\mathbf{K}^{-1}\|_1 \right\|_{L^\infty(\kappa)} \right) . \tag{3.6}$$

This quantity is easily and cheaply evaluated in each time step n. We then update the velocities and pressures by solving (3.1) whenever we find that $\theta(n, n^{(k)}) \ge \theta^*$, with θ^* a user-provided threshold.

Remark 3.3. In the common case of a diagonal permeability tensor $\mathbf{K}(\mathbf{x}) = k(\mathbf{x})\mathbf{1}$ the evaluation of (3.6) can be simplified using $\left\| \|\mathbf{K}^{-1}\|_1 \right\|_{L^\infty(\kappa)} = \left\| k^{-1} \right\|_{L^\infty(\kappa)}$.

Remark 3.4. The criterion above recomputes velocity and pressure whenever the current velocity would deviate more than a certain amount from the last computed one. On the other hand, at time steps where we do not recompute it, we use a velocity field extrapolated from the previous two times it was computed. A more appropriate criterion would therefore compare the (unknown) $U^{(n)}$ against the extrapolated velocity (i.e., a second order change) rather than against $U^{(n^{(k)})}$ (a first order change).

A proof of such a statement is immediately apparent. That said, we suspect that the resulting criterion replacing (3.6) would likely not make a significant difference other than that we would have to choose a different threshold θ^*. The reason is that we essentially only consider *significant* changes in the saturation, not infinitesimal ones. Furthermore, because the saturation is contained in a rather small interval, any large first order change must necessarily be accompanied by a significant second order change as well (linear change simply can't go on for very long if the function has to be bounded in $[0, 1]$). Thus, we conjecture that a second order version of the theorem would result in an indicator that, as far as practical consequences are concerned, would yield similar results to the first order version described above.

In summary, the algorithm outlined above allows us to perform a number of saturation time steps of length $\Delta t_c^{(n)}$ until criterion (3.6) tells us to recompute velocity and pressure variables, leading to a macro time step of length

$$\Delta t_p^{(n)} = \sum_{i=n^{(k)}+1}^{n} \Delta t_c^{(i)}.$$

We choose the length of (micro) steps subject to the Courant–Friedrichs–Lewy condition according to the criterion

$$\Delta t_c = \frac{\varepsilon \, \min_{\kappa\in\mathbb{T}} h_\kappa}{20\, p_S \, \|\mathbf{u}_t F'(S)\|_{L^\infty(\Omega)}}, \tag{3.7}$$

which we have confirmed through numerical experimentation to be stable for the choice of finite element and time stepping scheme for the saturation equation discussed below. Here, h_κ denotes the diameter of cell κ and p_S the polynomial degree of the shape functions used to discretize the saturation variable (where we choose $p_S = 1$ in all examples below). The result is a scheme where neither micro nor macro time steps are of uniform length, and both are chosen adaptively.

As an implementation detail, we note that we always solve the pressure-velocity part in the first three micro time steps to ensure accuracy at the beginning of computation, and to provide starting data to linearly extrapolate previously computed velocities to the current time step. A detailed description of the overall algorithm is shown in Algorithm 1.

Algorithm 1. Adaptive operator splitting for time steps $n \geq 3$.

```
$n^{(0)} \leftarrow 1$;
$n^{(1)} \leftarrow 2$;
$k = 1$;
for $n \leftarrow 3$ to $l$ do
    Compute the indicator $\theta(n, n^{(k)})$ for recomputing $\mathbf{u}_t, p$ from (3.6);
    if $\theta > \theta^*$ then
        Solve the pressure-velocity system for $\mathbf{u}_t^{(n)}, p^{(n)}$;
        $\mathbf{u}_t^* \leftarrow \mathbf{u}_t^{(n)}$;
        $n^{(k+1)} \leftarrow n$;
        $k = k + 1$;
    else
        Compute $\mathbf{u}_t^*$ by linear extrapolation to $t^{(n)}$ of $\mathbf{u}_t^{(n^{(k)})}, \mathbf{u}_t^{(n^{(k-1)})}$ defined at
        $t^{(n^{(k)})}, t^{(n^{(k-1)})}$;
    end
    Solve the saturation transport equation for $S^{(n)}$ using $\mathbf{u}_t^*$;
end
```

3.2. Spatial discretization and AMR strategy. The IMPES scheme described above requires the separate solution of velocity/pressure and saturation equations. We discretize these on the same mesh composed of quadrilaterals or hexahedra, using continuous Q_2 elements for the velocity and Q_1 elements for the pressure; this choice satisfies the usual Ladyzhenskaya–Babuska–Brezzi (LBB) conditions [14, 17]. We use continuous Q_1 finite elements to discretize the saturation equation.

As mentioned in the introduction, choosing meshes adaptively to resolve sharp saturation fronts is an essential ingredient to achieving efficiency in our algorithm. Here, we use the same shock-type refinement approach used in [21] to select those cells that should be refined or coarsened. The refinement indicator for each cell κ of the triangulation is computed by

$$\eta_\kappa = |\nabla S_h(\mathbf{x}_\kappa)|, \tag{3.8}$$

where $S_h(\mathbf{x}_\kappa)$ is the discrete saturation variable evaluated at the center of cell κ. This approach is analogous to ones frequently used in compressible flow problems, where density gradients are used to indicate refinement. Cells are coarsened if $\eta_\kappa < \theta_c$ and refined if $\eta_\kappa > \theta_r$ for given thresholds θ_c, θ_r; see also [21]. Meshes resulting from this choice can be seen, for example, in Figures 4.4 and 4.10 below.

3.3. Artificial diffusion stabilization of the saturation equation. The chosen Q_1 elements for the saturation equation do not lead to a stable discretization without upwinding or other kinds of stabilization, and spurious oscillations will appear in the numerical solution. Adding an artificial diffusion term is one approach to eliminating these oscillations [17]. On the other hand, adding too much diffusion smears sharp fronts in the solution and suffers from grid-orientation difficulties [17]. To avoid these effects, we adapt an artificial diffusion term originally proposed by Guermond and Pasquetti [41].

This method modifies the (discrete) weak form of the saturation equation (2.3) to read

$$\left(\epsilon\frac{\partial S_h}{\partial t}, \varphi_h\right) - \left(\mathbf{u}_t F\left(S_h\right), \nabla\varphi\right) + \left(\mathbf{n}\cdot\mathbf{u}_t \hat{F}\left(S_h\right), \varphi\right)_{\partial\Omega} + \left(\nu(S_h)\nabla S_h, \nabla\varphi_h\right) = 0, \tag{3.9}$$

where ν is the artificial diffusion parameter and $\hat{F}$ is an appropriately chosen numerical flux on the boundary of the domain (we use the obvious full upwind flux for this).

Following Guermond and Pasquetti [41], we choose $\nu(S_h)$ as a piecewise constant function, set on each cell κ with diameter h_κ as

$$\nu(S)|_\kappa = \beta \left\|\mathbf{u}_t \max\{F'(S), 1\}\right\|_{L^\infty(K)} \min\left\{h_\kappa, h_\kappa^\alpha \frac{\|\mathrm{Res}(S)\|_{L^\infty(\kappa)}}{c(\mathbf{u}_t, S)}\right\}, \tag{3.10}$$

where α is a stabilization exponent and β is a dimensionless user-defined stabilization constant. We choose $\mathbf{u}_t \max\{F'(S), 1\}$ instead of the true fluid velocity $\mathbf{u}_t F'(S)$ to avoid difficulties at so-called sonic points where $F'(S) = 0$.[3] The velocity and saturation global normalization constant, $c(\mathbf{u}_t, S)$, and the residual $\mathrm{Res}(S)$ are, respectively, given by

$$c(\mathbf{u}_t, S) = c_R \left\|\mathbf{u}_t F'(S)\right\|_{L^\infty(\Omega)} \mathrm{var}(S)^\alpha \ |\mathrm{diam}(\Omega)|^{\alpha-2} \tag{3.11}$$

[3] Taking the maximum with the constant 1 rather than any other number is motivated by the observation that for physical reasons we need to have $F(0) = 0, F(1) = 1$, and so on average over the interval $[0, 1]$ we have $F' = 1$. In models such as those used in section 4.1, the saturation remains within a narrower interval $[S_{wr}, 1-S_{nwr}]$ and one typically has $F(S_{wr}) = 0, F(1-S_{nwr}) = 1$. In that case, one may want to define the artificial viscosity by using the maximum with $1/(1 - S_{wr} - S_{nwr})$ since this is the average value of $F'(S)$ over the physically relevant interval. We believe that either choice is appropriate—the purpose of the modification is simply to have an artificial viscosity that is nonzero to ensure convergence to the viscosity limit even in places where $F'(S)(\mathbf{x}) = 0$.

and

$$\mathrm{Res}(S) = \left(\epsilon \frac{\partial S}{\partial t} + \mathbf{u}_t \cdot \nabla F(S) + F(S)q\right) \cdot S^{\alpha-1}, \tag{3.12}$$

where c_R is a second dimensionless user-defined constant, $\mathrm{diam}(\Omega)$ is the diameter of the domain, and $\mathrm{var}(S) = \max_\Omega S - \min_\Omega S$ is the range of the present saturation values in the entire computational domain Ω.

This stabilization scheme has a number of advantages over simpler schemes such as finite volume (or DG) methods or streamline upwind Petrov Galerkin (SUPG) discretizations. In particular, the artificial diffusion term acts primarily in the vicinity of discontinuities since the residual is small in areas where the saturation is smooth. It therefore provides for a higher degree of accuracy. On the other hand, it is nonlinear since ν depends on the saturation S. We avoid this difficulty by using an explicit Euler time stepping method, which leads to the following fully discrete problem at time step n:

$$\begin{aligned}(3.13)\quad \left(\epsilon S_h^{(n)}, \varphi_h\right) &= \left(\epsilon S_h^{(n-1)}, \varphi_h\right) + \Delta t_c^{(n)} \left(\mathbf{u}_t^* F\left(S_h^{(n-1)}\right), \nabla\varphi\right) \\ &- \Delta t_c^{(n)} \left(\mathbf{n} \cdot \mathbf{u}_t^* \hat{F}\left(S_h^{(n-1)}\right), \varphi\right)_{\partial\Omega} - \Delta t_c^{(n)} (\nu(S_h^{(n-1)}) \nabla S_h^{(n-1)}, \nabla\varphi_h) \qquad \forall \varphi_h,\end{aligned}$$

where $\mathbf{u}_t^*$ is the velocity linearly extrapolated from $\mathbf{u}_t^{(n^{(k)})}$ and $\mathbf{u}_t^{(n^{(k-1)})}$ to the current time $t^{(n)}$ (see Algorithm 1). Consequently, the equation is linear in $S_h^{(n)}$ and all that is required is to solve with a mass matrix on the saturation space.

3.4. Linear solvers and preconditioning. Following the discretization of the governing (2.1)–(2.3) discussed above, we are left with solving linear systems first for the velocity and pressure (if necessary, as determined by the criterion (3.6)) and then for the saturation. The first of these two has the form

$$\begin{pmatrix} \mathbf{M}^{\mathbf{u}} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}^{(n)} \\ \mathbf{P}^{(n)} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{F}_2 \end{pmatrix}, \tag{3.14}$$

where the individual matrices and vectors are defined as follows using shape functions $\mathbf{v}_i$ for velocity and ϕ_i for pressure:

$$\mathbf{M}^{\mathbf{u}}_{ij} = \left(\left(\mathbf{K}\lambda_t\left(S^{(n-1)}\right)\right)^{-1} \mathbf{v}_i, \mathbf{v}_j\right)_\Omega, \qquad \mathbf{B}_{ij} = -\left(\nabla \cdot \mathbf{v}_j, \phi_i\right)_\Omega, \tag{3.15}$$

$$(\mathbf{F}_2)_i = -\left(F\left(S^{(n-1)}\right) q, \phi_i\right)_\Omega. \tag{3.16}$$

We solve this linear system using the generalized minimal residual (GMRES) method [68]. Ideally, one would want to use the following preconditioner:

$$\mathbf{P} = \begin{pmatrix} \mathbf{M}^{\mathbf{u}} & \mathbf{0} \\ \mathbf{B} & -\mathbf{S} \end{pmatrix}, \qquad \mathbf{P}^{-1} = \begin{pmatrix} \left(\mathbf{M}^{\mathbf{u}}\right)^{-1} & \mathbf{0} \\ \mathbf{S}^{-1}\mathbf{B}\left(\mathbf{M}^{\mathbf{u}}\right)^{-1} & -\mathbf{S}^{-1} \end{pmatrix}, \tag{3.17}$$

where $\mathbf{S} = \mathbf{B}\left(\mathbf{M}^{\mathbf{u}}\right)^{-1}\mathbf{B}^T$ is the Schur complement [86] of the system. With this preconditioner, we have

$$\mathbf{P}^{-1} \begin{pmatrix} \mathbf{M}^{\mathbf{u}} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \left(\mathbf{M}^{\mathbf{u}}\right)^{-1}\mathbf{B}^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}. \tag{3.18}$$

It has been shown that GMRES converges in two iterations for this matrix [70]. On the other hand, we can of course not expect to use exact inverses of the velocity mass matrix and the Schur complement. We follow an idea by Silvester and Wathen [70] originally proposed for the Stokes system. Adapting it to the current set of equations yields the preconditioner

$$\tilde{\mathbf{P}}^{-1} = \begin{pmatrix} \widetilde{(\mathbf{M}^{\mathbf{u}})^{-1}} & \mathbf{0} \\ \widetilde{\mathbf{S}^{-1}}\mathbf{B}\widetilde{(\mathbf{M}^{\mathbf{u}})^{-1}} & -\widetilde{\mathbf{S}^{-1}} \end{pmatrix}, \tag{3.19}$$

where a tilde indicates an approximation of the exact inverse matrix. Since $\left(\mathbf{M}^{\mathbf{u}}\right)^{-1} = \left((\mathbf{K}\lambda_t)^{-1}\mathbf{v}_i, \mathbf{v}_j\right)_\Omega$ is a sparse symmetric and positive definite matrix, we choose for $\widetilde{(\mathbf{M}^{\mathbf{u}})^{-1}}$ a single application of a sparse incomplete Cholesky decomposition of this matrix [40]. On the other hand, the exact Schur complement $\mathbf{S}$ corresponds to the porous media flow operator in nonmixed form, i.e., the operator $-\nabla \cdot [\mathbf{K}\lambda_t(S)]\nabla$ applied to the pressure space; consequently, the matrix $\tilde{\mathbf{S}} = ((\mathbf{K}\lambda_t)\nabla\phi_i, \nabla\phi_j)_\Omega$ should be a good approximation of the actual Schur complement matrix $\mathbf{S}$ when applied to the pressure. $\tilde{\mathbf{S}}$ is again symmetric and positive definite, and we use an incomplete Cholesky decomposition of $\tilde{\mathbf{S}}$ for $\widetilde{\mathbf{S}^{-1}}$. It is important to note that $\tilde{\mathbf{S}}$ needs to be built with Dirichlet boundary conditions to ensure its invertibility. An alternative to the left preconditioner (3.19) would be to use a right-preconditioner; for ill-conditioned problems, this can make determining a stopping criterion for GMRES simpler (see [48]), though we have not found this to be necessary here.

Once the velocity $\mathbf{u}_t^*$ is available (either by direct computation or by extrapolation from time steps $n^{(k-1)}, n^{(k)}$; see Algorithm 1), we can assemble and solve the saturation equation using

$$\mathbf{M}^S\mathbf{S}^{(n)} = \mathbf{F}_3, \tag{3.20}$$

where $\mathbf{M}^S_{ij} = (\epsilon\phi_i, \phi_j)_\Omega$ and $\mathbf{F}_3$ as given in (3.13). The mass matrix $\mathbf{M}^S$ is solved by the conjugate gradient method, using once more an incomplete Cholesky decomposition as preconditioner.

4. Results and discussion.

In this section, we show numerical results that illustrate the efficiency and accuracy of our combined methods in solving (2.1)–(2.3). We first verify the accuracy of the methods by solving a 1D test case and evaluating the numerical solution against an analytical one. We then compare two-dimensional (2D) simulations with those previously obtained with a simpler finite element method and then consider a simulation with an unfavorably high mobility ratio that has in the past been used to demonstrate grid alignment effects. We finally utilize the approach for 3D computations and in particular show the speed-up obtained to test its efficiency. In the problems considered here, there will be no internal source term (i.e., $q = 0$) and flow will only be driven through pressure and saturation boundary conditions.

4.1. 1D verification.

In order to verify the reliability of the proposed methods, we choose the standard Buckley–Leverett problem [15] as a benchmark test. This set of equations describes the transient displacement of oil by injected water in a homogeneous, 1D horizontal system in the absence of capillary effects. We parameterize the flow using constant total mobility λ_t and permeability $\mathbf{K}$, and choose the relative permeability models for both phases as

$$k_{rw} = S_e^4, \qquad k_{rnw} = (1 - S_e)^2(1 - S_e^2),$$

Table 4.1
Physical parameters used in the 1D *verification (Buckley–Leverett) case.*

Parameter	Symbol	Value	Units
Computational domain	Ω	$[0, 300]$	m
Absolute permeability	k	1.0×10^{-7}	m^2
Porosity	ϵ	0.2	-
Total mobility	λ_t	2.0	$\mathrm{kg}^{-1} \cdot \mathrm{m} \cdot \mathrm{s}$
Residual saturation (wetting)	S_{wr}	0.2	-
Residual saturation (nonwetting)	S_{nwr}	0.2	-
Viscosity (wetting)	μ_w	0.001	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$
Viscosity (nonwetting)	μ_{nw}	0.001	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$
Stabilization exponent	α	1.0	-
Stabilization constant	β	0.35	-
Normalization constant	c_R	0.0003	-

where subscripts w, nw represent the wetting and nonwetting phases, respectively, and where $S_e = (S_w - S_{wr})/(1 - S_{wr} - S_{nwr})$ is the effective saturation and S_{*r} is the residual saturation for each phase. It is important to note that physical saturation levels cannot fall below the residual (or immobile) saturation levels; we therefore set $S_e = 0$ if $S < S_{wr}$ and $S_e = 1$ if $S > (1 - S_{nwr})$, thus ensuring the fractional flow of water remains bounded between 0 and 1.

With the two relative permeability models, the fractional flow of water with $\mu_w = \mu_{nw}$ (see [83]) is then

$$F(S) = \frac{\lambda_w}{\lambda_t} = \frac{\lambda_w}{\lambda_w + \lambda_{nw}} = \frac{k_{rw}(S)/\mu_w}{k_{rw}(S)/\mu_w + k_{rnw}(S)/\mu_{nw}} = \frac{S_e^4}{S_e^4 + (1 - S_e)^2 \cdot (1 - S_e^2)}.$$

The values of the physical parameters are summarized in Table 4.1. In addition, we choose initial and boundary conditions as follows:

$$\begin{aligned} S(\mathbf{x}, 0) &= 0.2, && \mathbf{x} \in \Omega, \\ S(0, t) &= 0.795, && t \in [0, T], \\ p(0, t) &= 2 \times 10^5, && t \in [0, T], \\ \mathbf{u}(300, t) \cdot \mathbf{n} &= 1.5 \times 10^{-7}, && t \in [0, T]. \end{aligned}$$

This model, describing the invasion of a water front from left to right, allows for a semianalytic solution against which we can compare our numerical solution at $t = 1500$ days (see [4, 83]). In particular, it is possible to compute the location of the discontinuity at this time accurately to 198.8. We note that in this test case, the incompressibility of the fluid leads to a constant velocity and the constant mobility results in a linear pressure profile. Neither of these two variables changes as a result of changes in the saturation and can therefore be solved only once at the beginning of the simulation (this is reflected by the fact that the adaptive operator splitting indicator (3.6) will always be zero for subsequent time steps).

Figure 4.1 shows convergence of the solution and that the artificial viscosity formulation allows us to resolve shocks with a width of some 5–6 cells and no oscillations. Figure 4.2 investigates whether using the entropy-based artificial viscosity (3.10) is indeed better than using a simple, first order artificial viscosity of the form

$$\nu(S)|_\kappa = \beta h_\kappa \left\| \mathbf{u}_t \max\{F'(S), 1\} \right\|_{L^\infty(K)}$$

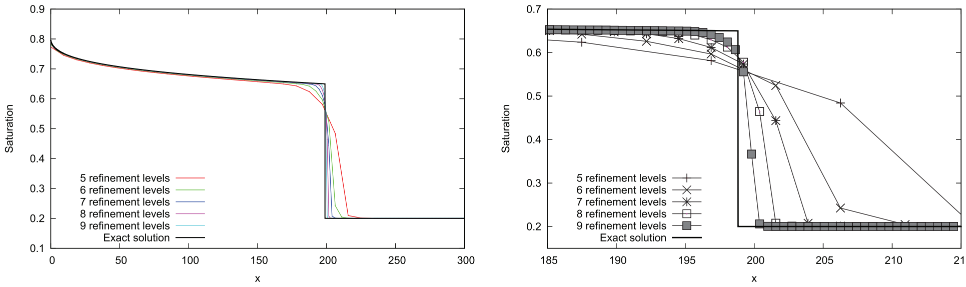

FIG. 4.1. *1D test case: Convergence of saturation profiles at $t = 1500$ days for different levels of mesh refinement. Left: Global view of the solution. Right: Zoom of the location of the discontinuity.*

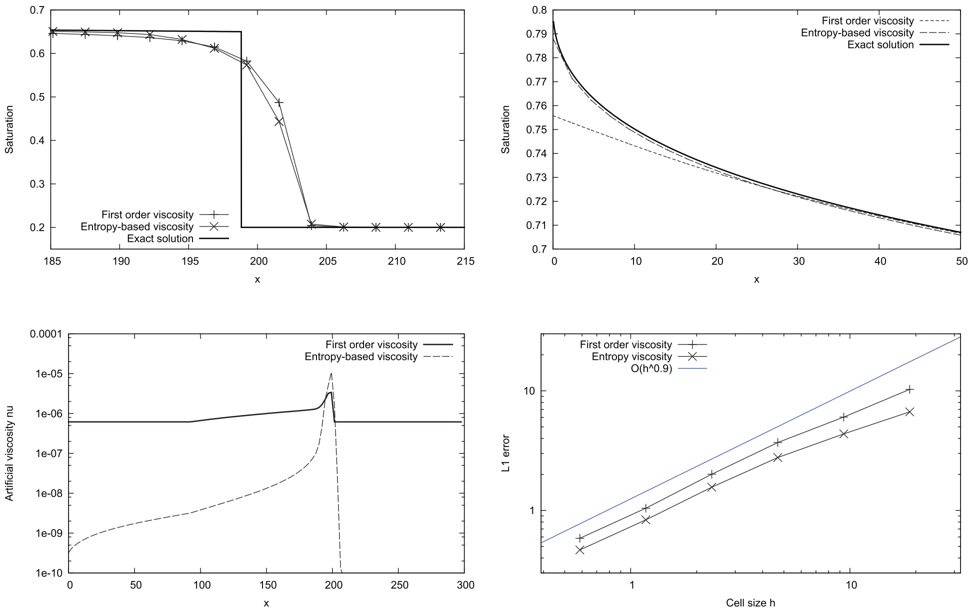

FIG. 4.2. *1D test case: Comparison in convergence between the first order artificial viscosity and the entropy-based artificial viscosity using solutions at time $t = 1500$ days. Top row: Solutions after 7 refinement cycles in the vicinity of the shock (left) and in the leftmost part of the domain (right). Bottom row: Sizes of the first order and entropy-based artificial viscosity terms in* (3.10) *from which the actual artificial viscosities are chosen by taking the smaller one (left), and error measured in the L_1 norm (right).*

as is frequently done. The figure shows that the shock is moderately better resolved; the primary difference, however, is in the resolution of the rarefaction wave at the left end of the domain, which is accurately represented using the entropy-based formulation, but not so with the first order viscosity that is too diffusive in this area (as shown in the bottom left panel of the figure, the first order viscosity is orders of magnitude larger than the entropy-based one). The figure also shows convergence in the L_1 norm that is appropriate for hyperbolic problems, at a rate of approximately 0.9 (the theoretical convergence rate approaches 1 as the polynomial degree of the finite element space increases) and that the entropy-based artificial viscosity leads to errors smaller by about a factor of 2 compared to the first order viscosity. We con-

TABLE 4.2
Physical parameters used in the 2D single-crack and 3D simulations.

PARAMETER	SYMBOL	VALUE	UNITS
Computational domain	Ω	$[0,1]^d$	m
Porosity	ϵ	1.0	-
Viscosity (wetting)	μ_w	0.2	$\mathrm{kg}\cdot\mathrm{m}^{-1}\cdot\mathrm{s}^{-1}$
Viscosity (nonwetting)	μ_{nw}	1.0	$\mathrm{kg}\cdot\mathrm{m}^{-1}\cdot\mathrm{s}^{-1}$
Stabilization exponent	α	1.0	-
Stabilization constant	β	0.3 in 2D; 0.27 in 3D	-
Normalization constant	c_R	1.0	-
Operator splitting threshold	θ^*	5.0	-

clude that even for the relatively simple 1D Buckley–Leverett problem, the benefits of the entropy-based artificial viscosity are substantial.

4.2. 2D comparison against existing methods. The results of the previous section show that the method converges to the exact solution as expected. In this section, we will investigate the level of accuracy we can expect from our method by comparing it with a widely used method for these equations, namely, using uniform grids together with discretization by Raviart–Thomas finite elements [66] for velocity and piecewise constant, discontinuous DG elements for pressure and saturation. This discretization is equivalent to a standard finite volume method for the saturation in conjunction with a staggered mesh discretization for the flow variables and satisfies the usual stability condition [14]; it is the basis for the program used in [55] against which we will compare results obtained with our algorithms.

To do so, let us consider a 2D test case where we parameterize total mobility λ_t and fraction flow F as follows:

$$\lambda_t(S) = \lambda_w(S) + \lambda_{nw}(S) = \frac{k_{rw}(S)}{\mu_w} + \frac{k_{rnw}(S)}{\mu_{nw}}, \tag{4.1}$$

$$F(S) = \frac{\lambda_w(S)}{\lambda_t(S)} = \frac{\lambda_w(S)}{\lambda_w(S) + \lambda_{nw}(S)} = \frac{k_{rw}(S)/\mu_w}{k_{rw}(S)/\mu_w + k_{rnw}(S)/\mu_{nw}}. \tag{4.2}$$

Coefficients are chosen as in Table 4.2. Following a commonly used prescription for the dependence of the relative permeabilities k_{rw} and k_{rnw} on saturation, we use

$$k_{rw}(S) = S^2, \qquad k_{rnw}(S) = (1-S)^2\,. \tag{4.3}$$

Initial conditions are required only for the saturation variable, and we choose $S(\mathbf{x},0) = 0$; i.e., the porous medium is initially filled by the nonwetting phase. We prescribe a linear pressure on the boundaries:

$$p(\mathbf{x},t) = 1 - x \qquad \text{on} \quad \partial\Omega \times [0,T]. \tag{4.4}$$

As discussed in section 2, pressure and saturation uniquely determine a velocity, and the velocity determines whether a boundary segment is an inflow or outflow boundary. On the inflow part of the boundary, $\mathbf{\Gamma}_{in}(t)$, we impose

$$S(\mathbf{x},t) = 1 \qquad \text{on} \quad \mathbf{\Gamma}_{in}(t) \cap \{x=0\}\,, \tag{4.5}$$

$$S(\mathbf{x},t) = 0 \qquad \text{on} \quad \mathbf{\Gamma}_{in}(t) \backslash \{x=0\}\,. \tag{4.6}$$

In other words, the domain is flooded by the wetting phase from the left. No boundary conditions for the saturation are required for the outflow parts of the boundary.

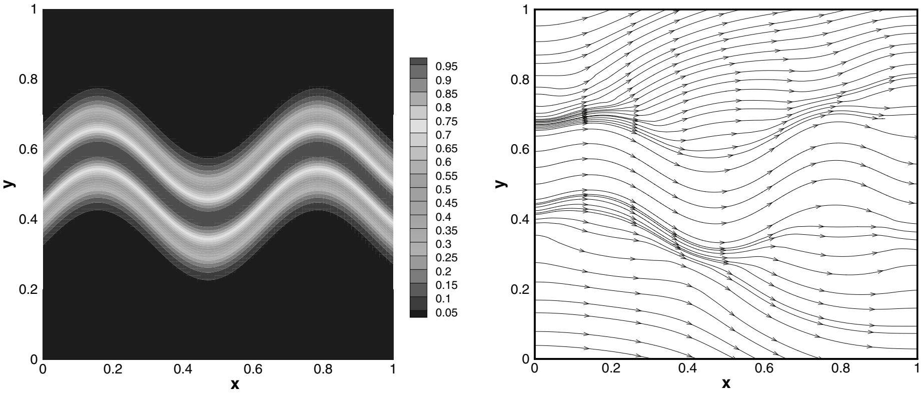

FIG. 4.3. *2D Single crack porous medium. Left: Permeability contour. Right: Streamline field.*

We consider a heterogeneous but isotropic porous media; i.e., we choose the permeability tensor as $\mathbf{K}(\mathbf{x}) = k(\mathbf{x}) \cdot \mathbf{1}$, and choose $k(\mathbf{x})$ as in [55] in the shape of a high permeability region that meanders from the left to the right (see the shape of the solution in Figure 4.4):

$$k(\mathbf{x}) = k_{sm}(\mathbf{x}) = \max\left\{\exp\left(-\left(\frac{y - 0.5 - 0.1 \cdot \sin(10x)}{0.1}\right)^2\right), 0.01\right\}. \tag{4.7}$$

This permeability field and streamlines for the resulting flow are shown in Figure 4.3.

In order to compare solutions, consider the case when the pressure-velocity system is solved once every 10 saturation time steps (fixed operator splitting). Figure 4.4 compares saturation contours between our discretization and the reference discretization outlined above at $t = 0.589$. The results are essentially identical, but it is clear that the adaptive mesh has far fewer degrees of freedom. Figure 4.5 shows the same saturation profiles but where the pressure-velocity system is solved only once every 30, 60, and 90 saturation time steps; it is obvious that the errors are getting more noticeable as the frequency of solving the pressure-velocity part decreases.

In contrast to this, when we drive the operator splitting by criterion (3.6) with $\theta^* = 5$, the difference between the two methods vanishes (see Figure 4.6), justifying our proposed indicator for the operator splitting algorithm. In particular, the solution so obtained is less diffusive than the one obtained by the reference approach using DG elements despite the use of a computationally far cheaper approach.

4.3. 2D quarter five-spot problem. The accurate simulation of miscible displacement plays an important role in the prediction of physical representation. Many such simulations involve adverse mobility ratio displacement; that is, the displacing fluid flows more easily through the porous medium than does the displaced fluid. For such cases, it has been demonstrated that numerical methods can suffer from fronts that are excessively smeared or that suffer from grid-orientation effects depending on whether flow is parallel or diagonal to mesh lines (see, for example, [47]). The grid-orientation phenomenon was first observed for immiscible displacement by Todd, O'Dell, and Hirasaki [78] in 1972. A more dramatic demonstration of the issue was

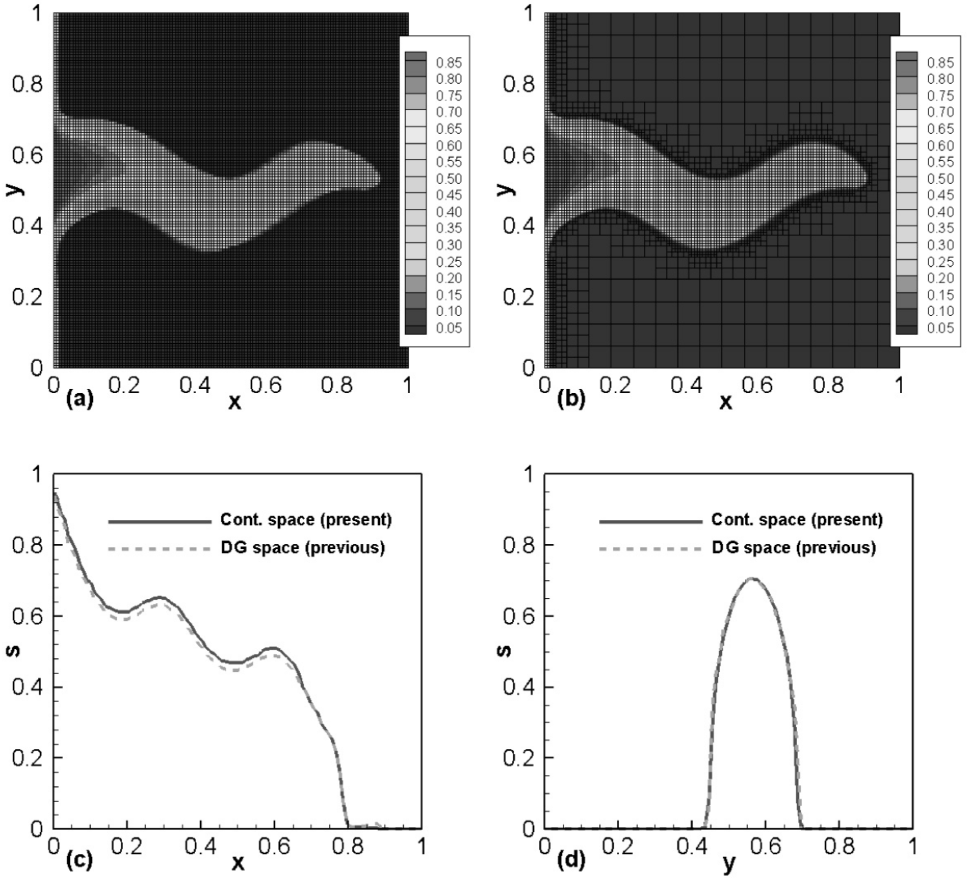

FIG. 4.4. *2D test case: Numerical results for our method in comparison with previous work of Li and Bangerth* [55] *at* $t = 0.589$*. The pressure-velocity system is solved once every* 10 *saturation time steps (fixed operator splitting).* (a) *Saturation field using DG space (previous work).* (b) *Saturation field using continuous space and adaptive meshes with the stabilizing term and operator splitting (present work).* (c) *Saturation profile along* $y = 0.5$*.* (d) *Saturation profile along* $x = 0.2$*.*

given by Coats et al. [22] in 1974. For a more complete discussion see Vinsome and Au [81].

Kozdon, Gerritsen, and Christie [47] noted that for some numerical schemes, with favorable mobility ratios, numerical solutions converge with grid refinement on both aligned and diagonal grids, while for unfavorable mobility ratios, convergence to the same solutions is not achieved with grid refinement. In order to verify that our numerical combination with adaptive grids is able to be used to obtain grid-independent solutions, let us consider results in which the mobility ratio is 30, i.e., the viscosity ratio $\mu_w/\mu_{nw} = 1/30$. As the geometry for this test case, we use a quarter five-spot. Here an initially nonwetting phase is drained through water injection happening at the lower left corner. The nonwetting fluid flows out of the domain at the upper right corner as shown in Figure 4.7. The relative permeability models and fractional flow of wetting phase used here are the same as described in the previous section.

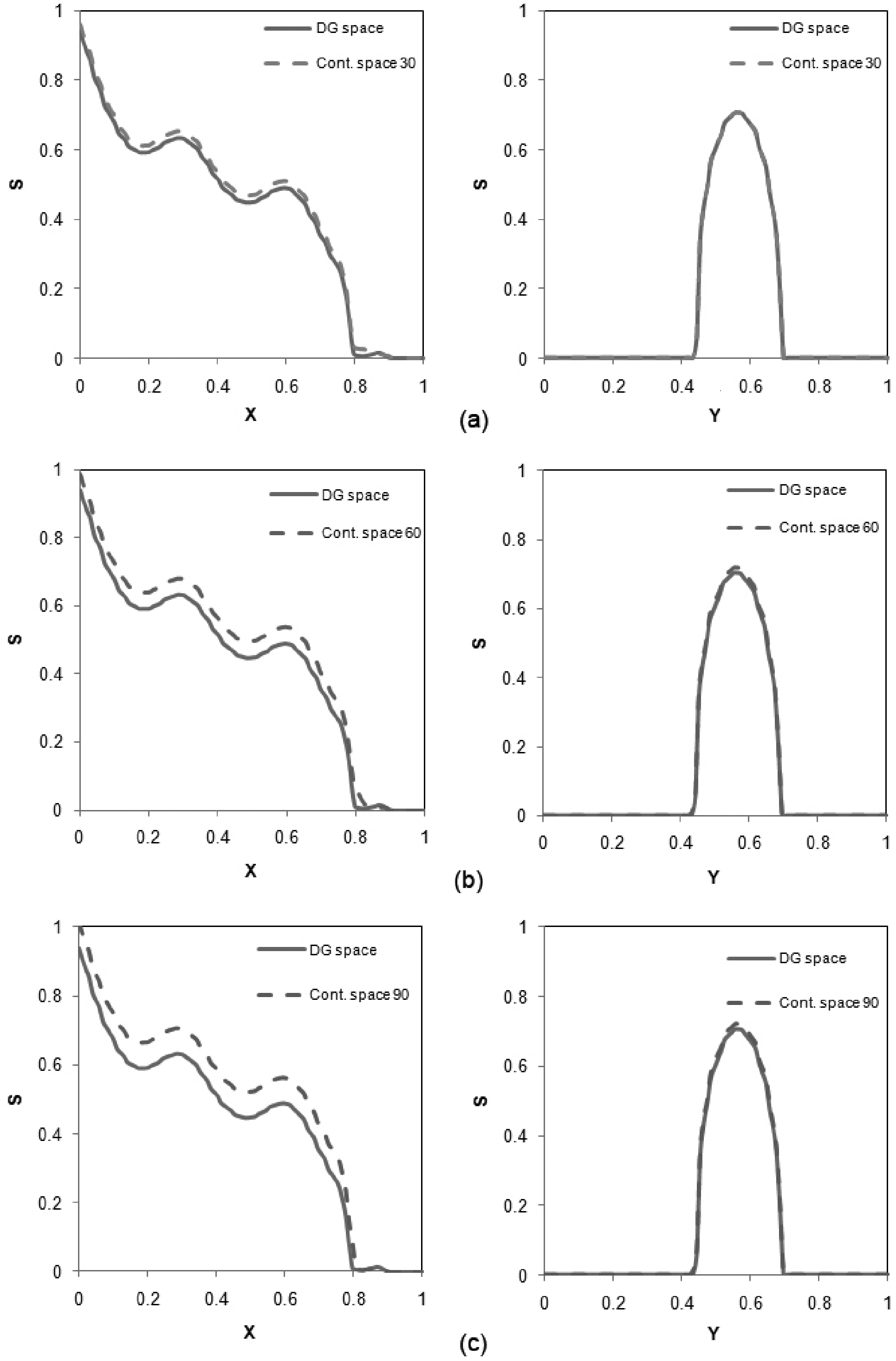

FIG. 4.5. *2D test case: Saturation profiles along $y = 0.5$ (left) and along $x = 0.2$ (right) in comparison with previous work of Li and Bangerth* [55] *at $t = 0.589$ when the pressure-velocity system is solved once every* (a) 30, (b) 60, *or* (c) 90 *saturation time steps (fixed operator splitting).*

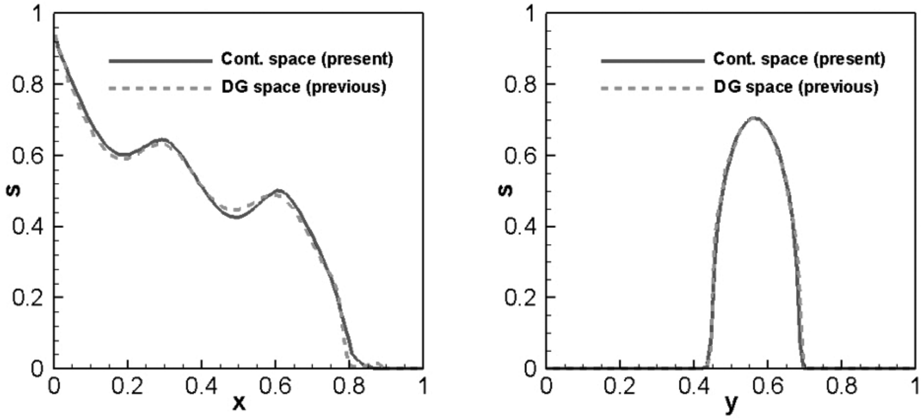

FIG. 4.6. *2D test case: Numerical results for the same situation as in Figure 4.4 but when the decision to solve the pressure-velocity system is made adaptively using criterion (3.6) with $\theta^* = 5$.*

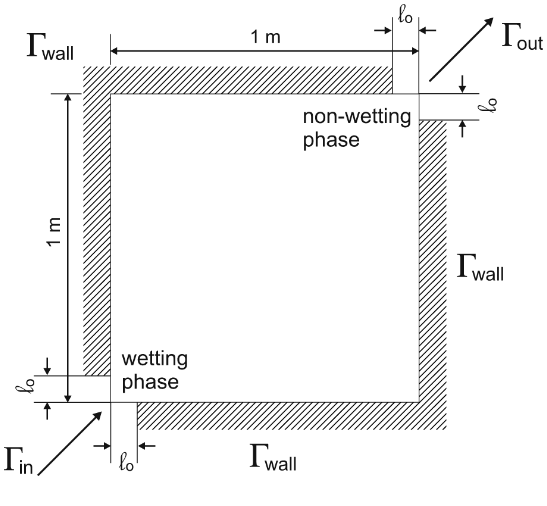

FIG. 4.7. *Setup for the quarter five-spot benchmark.*

The initial and boundary conditions are defined as

$$\begin{aligned} S(\mathbf{x},0) &= 0.0, && \mathbf{x} \in \Omega, \\ S &= 1.0, && \mathbf{x} \in \Gamma_{in}, \\ \mathbf{u}\cdot\mathbf{n} &= 0.0 && \mathbf{x} \in \Gamma_{wall}, \\ \mathbf{u}\cdot\mathbf{n} &= 0.1 && \mathbf{x} \in \Gamma_{in}, \\ p &= 0.0 && \mathbf{x} \in \Gamma_{out}. \end{aligned}$$

Material parameters for this setup are summarized in Table 4.3. The corresponding saturation field is shown in Figure 4.8 at $t = 7.773$, corresponding to roughly 0.19 pore volumes injected (PVI). This flow field is in good agreement with the results previously obtained by Kozdon, Gerritsen, and Christie [47] and Edwards [33].

TABLE 4.3
Physical parameters used in the 2D quarter five-spot problem.

PARAMETER	SYMBOL	VALUE	UNITS
Computational domain	Ω	$[0,1]^2$	m
Inlet or outlet side length	l_o	0.1	m
Absolute permeability	k	0.1	m^2
Porosity	ϵ	0.8	-
Residual saturation (wetting)	S_{wr}	0.0	-
Residual saturation (nonwetting)	S_{nwr}	0.0	-
Viscosity (wetting)	μ_w	1/30	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$
Viscosity (nonwetting)	μ_{nw}	1.0	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$
Stabilization exponent	α	1.0	-
Stabilization constant	β	0.8	-
Normalization constant	c_R	0.001	-

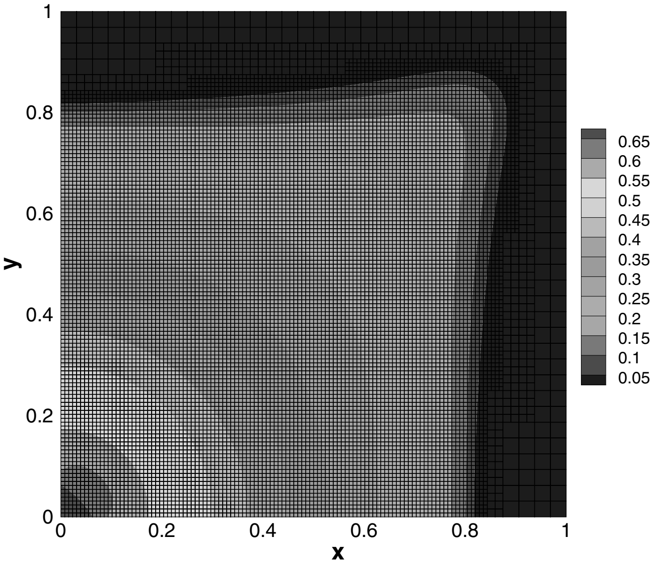

FIG. 4.8. *Quarter five-spot problem: Saturation contour with $\mu_w/\mu_{nw} = 1/30$ and adaptive grids at $t = 7.773$.*

4.4. 3D simulations. To evaluate the solution approach in three dimensions, we choose initial and boundary conditions in the same way as in the 2D simulations in section 4.2. Furthermore, we consider a heterogeneous but isotropic porous medium with a pseudo-random permeability described by

$$\begin{aligned} k(\mathbf{x}) = k_{rm}(\mathbf{x}) &= \min\left\{\max\left\{\sum_{l=1}^{N}\Psi_l(\mathbf{x}), 0.01\right\}, 4\right\}, \\ \Psi_l(\mathbf{x}) &= \exp\left(-\left(\frac{|\mathbf{x}-\mathbf{x}_l|}{0.05}\right)^2\right). \end{aligned} \tag{4.8}$$

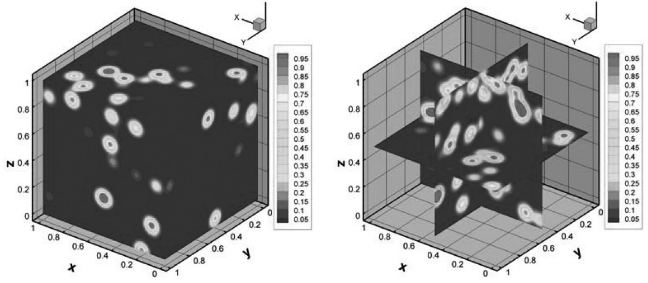

FIG. 4.9. *3D test case: Permeability fields of random porous media ($N = 200$). Left: Values on the surface of the domain. Right: Values along cutplanes through the center of the domain.*

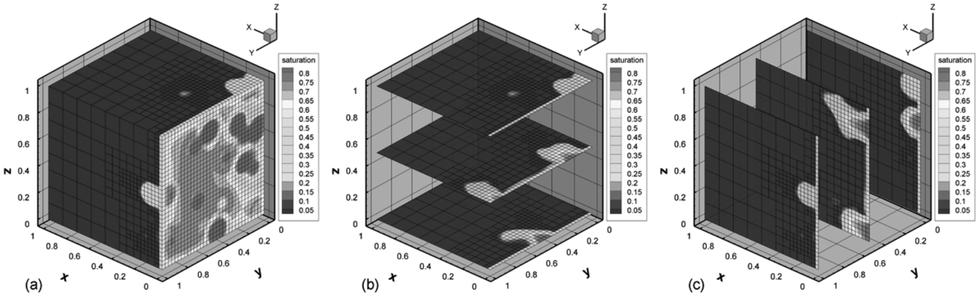

FIG. 4.10. *3D test case: Saturation distribution in a random porous medium with $N = 200$ at $t = 0.75$ (8,464 elements and 262,949 degrees of freedom).*

The N centers $\mathbf{x}_l$ of high permeability regions are randomly chosen in Ω. The permeability field is truncated from above and below to ensure a reasonable limit on the heterogeneous contrast. We will present results for $N = 200$, for which $k(\mathbf{x})$ is depicted in Figure 4.9.

Figure 4.10 shows the saturation distribution and corresponding adaptive grids as simulations progress, while Figure 4.11 shows a volume rendering of the invading wetting phase. The results of the inhomogeneity on the shape of the invading front are clearly visible.

In order to evaluate the adaptive operator splitting method, let us consider the number of saturation time steps between successive solves for the velocity and pressure variables, as determined by criterion (3.6). This is shown in Figure 4.12 for two different values of the threshold θ^*. Not surprisingly, choosing a larger value for the threshold relaxes the constraints and leads to more saturation time steps between macro solves. In both cases these numbers increase as the simulation progresses. This can be explained by observing that flow initially happens along only a few fingers but then spreads out across higher permeability areas; the average velocity of the front therefore declines, leading to smaller saturation changes per time step. From this figure, it is obvious that even with the tighter of the two criteria, the savings in the number of flow solves are at least a factor of 10 when averaged over the entire simulation; given that the flow solve is far more expensive than that for the saturation equation, the savings in computing time are corresponding. In addition, the variability of the time between macro solves shown in the figure illustrates the need for an

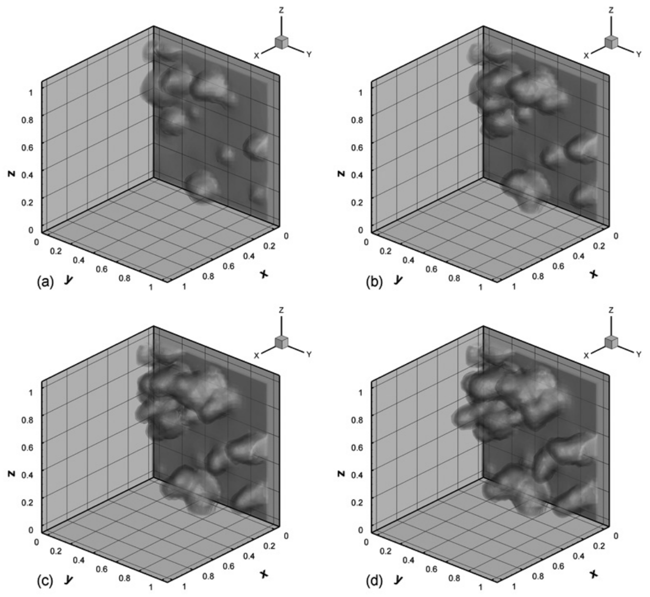

FIG. 4.11. *3D test case: Time evolution of the invading wetting phase front in the same situation as in Figure* 4.10. (a) $t = 0.19$. (b) $t = 0.38$. (c) $t = 0.57$. (d) $t = 0.74$.

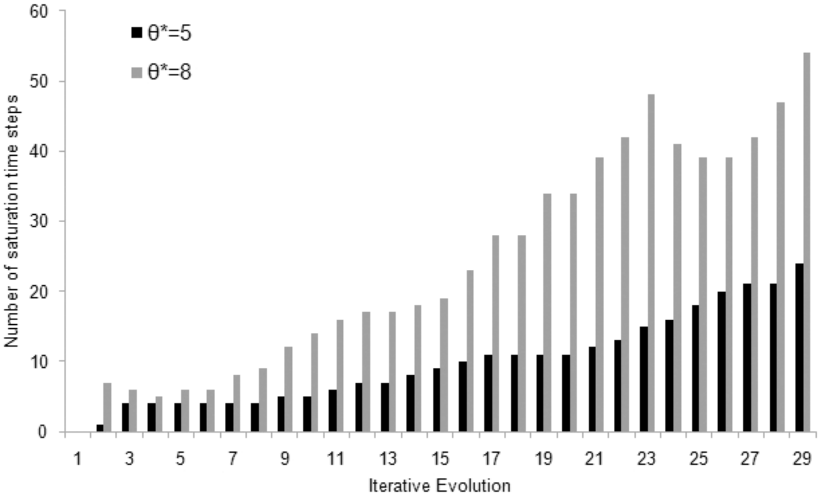

FIG. 4.12. *Numbers of saturation time steps between two solutions of the velocity-pressure system (i.e., between $n^{(k)}$ and $n^{(k-1)}$) for thresholds $\theta^* = 5$ and $\theta^* = 8$ in criterion* (3.6).

adaptive criterion, rather than simply solving the flow equations every fixed number of time steps.

4.5. Performance evaluation. In our final set of numerical experiments, we will compare the performance of our approach with algorithms that use simpler meshes, solvers, or operator splitting methods. We will do so both in two as well as in three dimensions, where computations are typically very expensive due to the

Table 4.4

Comparison of performance (total wall clock time in days from $t = 0$ to $t = 2$ with $N = 200$ and $\theta^* = 5$) for four combinations of our algorithms: uniform mesh ("UM"), adaptive mesh ("AM"), adaptive mesh plus adaptive operator splitting ("AM+AOS"), and adaptive mesh plus adaptive operator splitting plus block preconditioning ("AM+AOS+BP").

	UM	AM	AM+AOS	AM+AOS+BP
CPU time (days), 2D single crack	7.2	4.2	0.2	0.04
CPU time (days), 3D random porous	96.5	55.9	7.5	0.58

number of degrees of freedom required to solve sharp fronts. In particular, we compare the run time of different combinations of the methods discussed herein (adaptive mesh, adaptive mesh plus adaptive operating splitting, and adaptive mesh plus adaptive operating splitting plus block matrix preconditioning). The number of degrees of freedom ranges from 2,176 to 41,732 in two dimensions, and from 63,490 to 449,999 in three dimensions. All other conditions, including the use of the same CPU (AMD Opteron 2.2 GHz), are identical between simulations, and we use the same parameters as well as the same minimum element size h. The base line code uses a uniform mesh, solves the velocity/pressure system in each saturation time step, and uses the Schur complement solver previously used in [55].

Table 4.4 shows run times for the various combinations. Mesh adaptivity alone improves performance in the 2D/3D cases by about 43%. The adaptive operator splitting plus adaptive grids (third column) already provides up to 92% reduction in computing time compared with using uniform grids and solving the velocity-pressure system in each time step. Finally, the combination of all methods considered in this paper reduces compute time by a factor of 160–180, making computations possible that would otherwise be unachievable. The compute times could be further accelerated by parallelizing operations across multiple cores of a single computer, or across the nodes of a cluster. For example, we report in [48] that all of the techniques outlined here can be scaled to hundreds or even thousands of processors in the context of solving a problem described by the Boussinesq approximation that is structurally similar to the equations used here.

5. Conclusions and future work. Motivated by the computational expense of the methods that are traditionally used for this task, we have presented efficient integrated numerical methods allowing high-resolution simulation of two-phase flow in porous media. The methods are based on a finite element discretization using continuous elements and consist of (i) an adaptive operator splitting method that only recomputes the velocity and pressure variables whenever necessary, as determined by a new objective indicator; (ii) block matrix preconditioning methods that greatly reduce the compute time needs; (iii) an entropy-based stabilization term that preserves accuracy and ensures stability; and (iv) locally adaptive mesh refinement allowing highly resolved time-dependent simulations. The robustness and effectiveness of the numerical methods were demonstrated through a number of simulations and compared with regard to both their accuracy as well as the required CPU time. In addition, the methods described here provide no significant bottleneck to parallelization and can be made to run on clusters of computers using the techniques discussed in [48] for even greater computational efficiency. As also shown there, the cost of refining the mesh occasionally is negligible, and thus even small reductions in the number of cells due to adaptivity pay off in terms of overall run time.

More complex multiphase flows involve additional physical processes, such as multiple interacting phases or capillary effects with a capillary pressure relationship that is fundamentally dependent on material properties and operating conditions, saturation, wettability, and other material properties [50, 51, 52]. The addition of such terms obviously complicates the computation of solutions, but we believe that the techniques described herein can readily be extended to such problems. In particular, following a similar mathematical procedure to that described in section 3.1, it should be possible to extend our operator splitting indicator to these more complex situations.

A natural starting point for experiments in this direction would be the freely available and extensively documented source code for the experiments discussed herein [20].

Acknowledgments. CCC thanks Dr. Zhangxin Chen (University of Calgary) for his invaluable insights on finite element theory, Dr. Sergey Mashchenko (McMaster University) for installing the relevant libraries, and Mrs. Hsiao-Yun Janette Cheng (University of Sheffield) for help with the graphics. WB thanks Timo Heister and Martin Kronbichler for many discussions on flow solvers.

Access to the Shared Hierarchical Academic Research Computing Network (SHARCNET, http://www.sharcnet.ca) and Compute/Calcul Canada is gratefully acknowledged.

REFERENCES

[1] E. Abreu, J. Douglas, F. Furtado, and F. Pereira, *Operating splitting for three-phase flow in heterogeneous porous media*, Commun. Comput. Phys., 6 (2009), pp. 72–84.

[2] M. Ainsworth and J. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, John Wiley, New York, 2000.

[3] D. Arnold, R. Falk, and R. Winther, *Preconditioning H(div) and applications*, Math. Comp., 66 (1997), pp. 957–984.

[4] K. Aziz and A. Settari, *Petroleum Reservoir Simulation*, Applied Science Publishers, London, 1979.

[5] I. Babuška and W. C. Rheinboldt, *Error estimates for adaptive finite element computations*, SIAM J. Numer. Anal., 15 (1978), pp. 736–754.

[6] W. Bangerth et al., deal.II. *Differential Equations Analysis Library*, Technical Reference, 2012; available online from http://www.dealii.org/.

[7] W. Bangerth, R. Hartmann, and G. Kanschat, deal.II*—a general purpose object oriented finite element library*, ACM Trans. Math. Softw., 33 (2007), 24.

[8] W. Bangerth and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser Verlag, Basel, 2003.

[9] A. Bazylak, V. Berejnov, B. Markicevic, D. Sinton, and N. Djilali, *Numerical and microfluidic pore networks: Towards designs for directed water transport in GDLs*, Electrochimica Acta, 53 (2008), pp. 7630–7637.

[10] M. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys., 82 (1989), pp. 64–84.

[11] T. Berning and N. Djilali, *A 3D, multiphase, multicomponent model of the cathode and anode of a PEM fuel cell*, J. Electrochem. Soc., 150 (2003), pp. A1589–A1598.

[12] M. Blunt, *Flow in porous media—pore-network models and multiphase flow*, Current Opinion Colloid Interface Sci., 6 (2001), pp. 197–207.

[13] M. Blunt and B. Rubin, *Implicit flux limiting schemes for petroleum reservoir simulation*, J. Comput. Phys., 102 (1990), pp. 194–210.

[14] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.

[15] S. Buckley and M. Leverett, *Mechanism of fluid displacements in sands*, AIME Trans., 146 (1942), pp. 107–116.

[16] G. Carey, *Computational Grids: Generation, Adaptation and Solution Strategies*, Taylor & Francis, Washington, DC, 1997.

[17] Z. Chen, *Finite Element Methods and Their Applications*, Springer-Verlag, New York, 2005.

[18] Z. Chen, G. Huan, and B. Li, *An improved IMPES method for two-phase flow in porous media*, Transp. Porous Media, 54 (2004), pp. 361–376.

[19] Z. Chen, G. Huan, and Y. Ma, *Computational Methods for Multiphase Flows in Porous Media*, SIAM, Philadelphia, 2006.

[20] C. Chueh and W. Bangerth, *The* deal.II *Tutorial Manual: Step*-43, available online from http://dealii.org/developer/doxygen/deal.II/step_43.html.

[21] C. Chueh, M. Secanell, W. Bangerth, and N. Djilali, *Multi-level adaptive simulation of transient two-phase flow in heterogeneous porous media*, Comput. & Fluids, 39 (2010), pp. 1585–1596.

[22] K. Coats, W. George, C. Chu, and B. Marcum, *Three dimensional simulation of streamflooding*, AIME Trans., 257 (1974), pp. 573–592.

[23] H. Dahle, *Operator Splitting Techniques for Convection-Dominated Diffusion Problems in One and Two Space Dimensions*, Ph.D. Thesis, Department of Mathematics, University of Bergen, Norway, 1988.

[24] Y. Di, R. Li, and T. Tang, *A general moving mesh framework in* 3*D and its application for simulating the mixture of multi-phase flows*, Commun. Comput. Phys., 3 (2008), pp. 582–602.

[25] Y. Di, R. Li, T. Tang, and P. Zhang, *Moving mesh finite element methods for the incompressible Navier–Stokes equations*, SIAM J. Sci. Comput., 26 (2005), pp. 1036–1056.

[26] Y. Di, R. Li, T. Tang, and P. Zhang, *Moving mesh methods for singular problems on a sphere using perturbed harmonic mappings*, SIAM J. Sci. Comput., 28 (2006), pp. 1490–1508.

[27] N. Djilali, *Computational modelling of polymer electrolyte membrane (PEM) fuel cells: Challenges and opportunities*, Energy, 32 (2007), pp. 269–280.

[28] J. Douglas, *On the numerical integration of* $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ *by implicit methods*, J. Soc. Indust. Appl. Math., 3 (1955), pp. 42–65.

[29] J. Douglas and D. Peaceman, *Numerical solution of two-dimensional heat flow problems*, Amer. Inst. Chem. Engrg. J., 1 (1955), pp. 505–512.

[30] J. Douglas and D. Peaceman, *On the numerical solution of heat conduction problems in two and three space variables*, Trans. Amer. Math. Soc., 82 (1960), pp. 421–439.

[31] J. Douglas, D. Peaceman, and H. Rachford, *A method for calculating multi-dimensional immiscible displacement*, Trans. AIME, 216 (1959), pp. 297–308.

[32] E. D'yakonov, *On an iterative method for the solution of a system of finite-difference equations*, DAN SSSR, 143 (1961), pp. 522–525.

[33] M. G. Edwards, *Multi-dimensional wave-oriented upwind schemes with reduced cross-wind diffusion for flow in porous media*, Internat. J. Numer. Methods Fluids, 67 (2011), pp. 33–57.

[34] P. Ellsiepen, *Zeit- und ortsadaptive Verfahren angewandt auf Mehrphasenprobleme poröser Medien*, Dissertation, Institut für Mechanik, Lehrstuhl II, Universität Stuttgart, 1999.

[35] M. Espedal and R. Ewing, *Characteristic Petrov-Galerkin subdomain methods for two-phase immiscible flow*, Comput. Methods Appl. Mech. Engrg., 64 (1987), pp. 113–135.

[36] M. Espedal and K. Karlsel, *Numerical Simulation of Reservoir Flow Models Based on Large Time Step Operator Splitting Algorithms*, Lecture Notes in Math. 1734, Springer-Verlag, New York, 2000.

[37] J. P. de S. R. Gago, D. W. Kelly, O. C. Zienkiewicz, and I. Babuška, *A posteriori error analysis and adaptive processes in the finite element method: Part* II*—Adaptive mesh refinement*, Internat. J. Numer. Methods Engrg., 19 (1983), pp. 1621–1656.

[38] S. Geiger, K. Schmid, and Y. Zaretskiy, *Mathematical analysis and numerical simulation of multi-phase multi-component flow in heterogeneous porous media*, Current Opinion Colloid Interface Sci., 17 (2012), pp. 147–155.

[39] M. Gerritsen and L. Durlofsky, *Modeling fluid flow in oil reservoirs*, Annu. Rev. Fluid Mech., 37 (2005), pp. 211–238.

[40] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[41] J. Guermond and R. Pasquetti, *Entropy-based nonlinear viscosity for Fourier approximations of conservation laws*, C. R. Math. Acad. Sci. Paris, 346 (2008), pp. 801–806.

[42] R. Helmig, *Multiphase Flow and Transport Processes in the Subsurface*, Springer-Verlag, Heidelberg, 1997.

[43] R. Hinkelmann, *Efficient Numerical Methods and Information-Processing Techniques in Environment Water*, Habilitation, Universität Stuttgart, 2002.

[44] S. Ingebritsen, S. Geiger, S. Hurwitz, and T. Driesner, *Numerical simulation of magmatic hydrothermal systems*, Rev. Geophys., 48 (2010), 2009RG000287.

[45] P. Jenny, H. Tchelepi, and S. Lee, *Unconditionally convergent nonlinear solver for hyperbolic conservation laws with s-shaped flux functions*, J. Comput. Phys., 228 (2009), pp. 7497–7512.

[46] K. Karlsen, K. Lie, J. Natvig, H. Nordhaug, and H. Dahle, *Operator splitting methods for systems of convection-diffusion equations: Nonlinear error mechanisms and correction strategies*, J. Comput. Phys., 173 (2001), pp. 636–663.

[47] J. Kozdon, M. Gerritsen, and M. Christie, *Grid orientation revisited: Near-well, early-time effects and solution coupling methods*, Transp. Porous Media, 73 (2008), pp. 255–277.

[48] M. Kronbichler, T. Heister, and W. Bangerth, *High accuracy mantle convection simulation through modern numerical methods*, Geophys. J. Int., 191 (2012), pp. 12–29.

[49] D. Kröner, *Numerical Schemes for Conservation Laws*, John Wiley, Chichester, UK, Teubner, Stuttgart, 1997.

[50] E. Kumbur, K. Sharp, and M. Mench, *Validated Leverett approach for multiphase flow in PEFC diffusion media* I. *Hydrophobicity effect*, J. Electrochem. Soc., 154 (2007), pp. B1295–B1304.

[51] E. Kumbur, K. Sharp, and M. Mench, *Validated Leverett approach for multiphase flow in PEFC diffusion media* II. *Compression effect*, J. Electrochem. Soc., 154 (2007), pp. B1305–B1314.

[52] E. Kumbur, K. Sharp, and M. Mench, *Validated Leverett approach for multiphase flow in PEFC diffusion media* III. *Temperature effect and unified approach*, J. Electrochem. Soc., 154 (2007), pp. B1315–B1324.

[53] J. Larminie and A. Dicks, *Fuel Cell Systems Explained*, John Wiley, New York, 2006.

[54] R. Li and T. Tang, *Moving mesh discontinuous Galerkin method for hyperbolic conservation laws*, J. Sci. Comput., 27 (2006), pp. 347–363.

[55] Y. Li and W. Bangerth, *The* `deal.II` *Tutorial Manual: Step*-21, available online from http://dealii.org/developer/doxygen/deal.II/step_21.html.

[56] K. Lie, S. Krogstad, I. Ligaarden, J. Natvig, H. Nilsen, and B. Skaflestad, *Open-source MATLAB implementation of consistent discretisations on complex grids*, Comput. Geosci., 16 (2012), pp. 297–322.

[57] B. Markicevic, A. Bazylak, and N. Djilali, *Determination of transport parameters for multiphase flow in porous gas diffusion electrodes using a capillary network model*, J. Power Sources, 171 (2007), pp. 706–717.

[58] D. McKenzie, *The extraction of magma from the crust and mantle*, Earth Planet. Sci. Lett., 74 (1985), pp. 81–91.

[59] M. Mench, *Fuel Cell Engines*, John Wiley, New York, 2008.

[60] D. Peaceman, *Fundamentals of Numerical Reservoir Simulation*, Elsevier, Amsterdam, 1977.

[61] D. W. Peaceman and H. H. Rachford, *The numerical solution of parabolic and elliptic differential equations*, J. Soc. Indust. Appl. Math., 3 (1955), pp. 28–41.

[62] M. Piri and M. Blunt, *Three-dimensional mixed-wet random pore-scale network modeling of two- and three-phase flow in porous media.* I. *Model description*, Phys. Rev. E, 71 (2005), 026301.

[63] M. Piri and M. Blunt, *Three-dimensional mixed-wet random pore-scale network modeling of two- and three-phase flow in porous media.* II. *Results*, Phys. Rev. E, 71 (2005), 026302.

[64] C. Powell, *Parameter-free H(div) preconditioning for a mixed finite element formulation of diffusion problems*, IMA J. Numer. Anal., 25 (2005), pp. 783–796.

[65] C. E. Powell and D. Silvester, *Optimal preconditioning for Raviart–Thomas mixed formulation of second-order elliptic problems*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 718–738.

[66] P. Raviart and J. Thomas, *A mixed finite element method for* 2nd *order elliptic problems*, in Mathematical Aspects of the Finite Element Method, Lecture Notes in Math. 606, Springer-Verlag, Berlin, 1977, pp. 292–315.

[67] T. Rusten and R. Winther, *A preconditioned iterative method for saddlepoint problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.

[68] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[69] J. Sheldon, B. Zondek, and W. Cardwell, *One-dimensional, incompressible, non-capillary, two-phase fluid flow in a porous medium*, T. SPE AIME, 216 (1959), pp. 290–296.

[70] D. Silvester and A. Wathen, *Fast iterative solution of stabilised Stokes systems. Part* II: *Using general block preconditioners*, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.

[71] H. Stone and J. Garder, *Analysis of gas-cap or dissolved-gas reservoirs*, T. SPE AIME, 222 (1961), pp. 92–104.

[72] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*, Oxford Science Press, New York, 2001.

[73] A. Tambue, G. Lord, and S. Geiger, *An exponential integrator for advection-dominated reactive transport in heterogeneous porous media*, J. Comput. Phys., 229 (2010), pp. 3957–3969.

[74] Z. Tan, T. Tang, and Z. Zhang, *A simple moving mesh method for one- and two-dimensional phase-field equations*, J. Comput. Appl. Math., 190 (2006), pp. 252–269.

[75] Z. Tan, Z. Zhang, Y. Huang, and T. Tang, *Moving mesh methods with locally varying time steps*, J. Comput. Phys., 200 (2004), pp. 347–367.

[76] H. Tang and T. Tang, *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal., 41 (2003), pp. 487–515.

[77] A. Taniguchi, T. Akita, K. Yasuda, and Y. Miyazaki, *Analysis of electrocatalyst degradation in PEMFC caused by cell reversal during fuel starvation*, J. Power Sources, 130 (2004), pp. 42–49.

[78] M. Todd, P. O'Dell, and G. Hiraski, *Methods for increased accuracy in numerical reservoir simulation*, SPE J. (1972), pp. 515–530.

[79] B. van Leer, *Upwinding and high-resolution methods for compressible flow: From donor cell to residual-distribution schemes*, Commun. Comput. Phys., 1 (2006), pp. 192–206.

[80] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh Refinement Techniques*, John Wiley, New York, Teubner, Stuttgart, 1996.

[81] P. Vinsome and A. Au, *One Approach to the Grid Orientation Problem in Reservoir Simulation*, SPE paper 8247, Las Vegas, NV, 1979.

[82] M. von Paul, *Simulation of Two-phase Flow Processes in Heterogeneous Porous Media with Adaptive Methods*, Dissertation, Institut für Wasserbau, Universität Stuttgart, 2003.

[83] E. Wendland and D. Flensberg, *Numerical solution of two-phase flow for the advection-dominated and non-linear case*, Adv. Water Resources, 28 (2005), pp. 643–660.

[84] D. Wolf-Gladrow, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction*, Lecture Notes in Math. 1725, Springer-Verlag, Berlin, 2000.

[85] N. Yanenko, *Méthode a Pas Fractionnaires*, Librairie Armand Colin, Paris, 1968.

[86] F. Zhang, *The Schur Complement and Its Applications*, Springer-Verlag, New York, 2005.

[87] Y. Zhang, H. Wang, and T. Tang, *Simulating two-phase viscoelastic flows using moving finite element methods*, Commun. Comput. Phys., 7 (2010), pp. 333–349.