Time-of-Flight + Fast Marching + Transport Collapse: An Alternative to Streamlines for Two-Phase Porous Media Flow with Capillary Forces?

I. Berre[a], H. K. Dahle[a], K. Hvistendahl Karlsen[a], K.–A. Lie[b] and J. R. Natvig[b]

[a]Department of Mathematics, University of Bergen,
Johs. Brunsgt. 12, N–5008 Bergen, Norway

[b]SINTEF, Applied Mathematics,
P. O. Box 124 Blindern, N–0314 Oslo, Norway

1. INTRODUCTION

State-of-the-art technology in reservoir characterization allows for the generation of high-resolution reservoir models with multi-million cells. Moreover, increased demands for assessment of uncertainties and history matching require fast and accurate flow simulations on multiple plausible geological models on a routinely basis. Conventional reservoir simulators fail to fulfill this need. This has lead to a revival of streamline simulators based on time-of-flight coordinates.

We suggest an alternative to streamline methods that avoids the geometrical complexity inherent in representing the individual streamlines. Our novel method has four essential building blocks: (i) a reformulation of the advection of characteristics into a steady time-of-flight equation; (ii) an efficient fast marching method for the time-of-flight equation; (iii) the transport-collapse operator for appropriate averaging of multivalued solutions; and (iv) an operator splitting to separate advective and capillary transport of fluids.

A combination of the method of characteristics and the transport-collapse operator has been used by several authors to derive large-time-step numerical methods. Here we show how the multivalued solutions needed in the transport collapse approach easily can be computed by recasting the transportation of characteristics to an Eulerian framework, in which it becomes a boundary value problem for a steady linear advection equation. The advection equation is discretized by a simple upwind method and high computational efficiency is obtained through an optimal ordering of the unknowns in a manner that is reminiscent of the fast marching approach [12,5,11]. The resulting method is unconditionally stable in the sense that the time step in the saturation solver is not restricted by any CFL-condition. Moreover, the method is a generalization of the fast marching approach developed in [8] (which could not handle capillary forces).

We present an application of the method for simulation of two-phase flow in a reservoir with capillary effects, together with comparison to another front tracking method. The conclusion is that the method compares well with the front tracking approach.

2. GOVERNING EQUATIONS

We consider a black-oil model for two-phase, two-component, immiscible, incompressible flow in two space dimensions. The phases are the non-wetting oil phase (n) and the wetting water phase (w). Assuming that gravity can be neglected, the equations describing this type of flow can be written as

$$-\nabla \cdot \big[\lambda_T \mathbf{K} \cdot \nabla p\big] = q, \tag{1}$$

$$\mathbf{v} = -\lambda_T \mathbf{K} \cdot \nabla p, \tag{2}$$

$$\phi \frac{\partial S}{\partial t} + \mathbf{v} \cdot \nabla f(S) - \varepsilon \nabla\big(\mathbf{D}(S) \cdot \nabla S\big) = 0 \tag{3}$$

by means of global pressure p and total Darcy velocity $\mathbf{v}$, see e.g., [3]. Here, the porosity ϕ and the absolute permeability $\mathbf{K}$ of the porous medium are spatially dependent rock parameters; S is the saturation of the wetting phase; and λ_i is the mobility of phase i. The mobility is a rock-fluid property given by the ratio of the relative permeability to the viscosity, $\lambda_i = k_{ri}/\mu_i$. The total mobility of the phases is given by $\lambda_T = \lambda_w + \lambda_n$. Transport effects are given by the fractional flow function $f = \lambda_w/(\lambda_n + \lambda_w)$ and capillary forces by the diffusion tensor $\mathbf{D}$. The relative importance of capillary forces is given by ε, which typically is a small parameter. Finally, q represents contributions from injection and production wells. An additional constraint is given by $S_n + S_w = 1$. To close the system of equations we need constitutive relationships for the capillary pressures and relative permeabilities. In the experiments reported below we use $\mathbf{D} = \mathbf{K}$. The relative permeabilities are defined by a standard Corey model, $k_w = S^2$ and $k_n = (1 - S)^2$. In the numerical experiments we consider a quarter-five spot problem, for which no-flow conditions are imposed on the boundaries.

3. NUMERICAL SOLUTION METHOD

To decouple the pressure-velocity equations (1)–(2) from the saturation equation (3), we use a sequential time-stepping procedure. Suppose that we know the saturation at time $t^n = n\Delta t$. To advance the solution forward a small time-step Δt we use a standard operator splitting. First the pressure equation (1) is solved with the saturation held fixed and the total Darcy velocity is estimated from (2). Then the saturation equation (3) can be solved with the velocity field kept fixed. This can be continued up to a final computing time $T = N\Delta t$.

To solve the saturation equation, we use operator splitting. First, advective transport of fluids is accounted for by solving the hyperbolic equation

$$\phi \frac{\partial S}{\partial t} + \mathbf{v} \cdot \nabla f(S) = 0, \qquad S(\mathbf{x}, 0) = S^n(\mathbf{x}) \tag{4}$$

one time-step Δt forward in time. This gives a first approximation, $\bar{S}^{n+1}$, of the solution S^{n+1}. Then capillary forces are accounted for by solving the parabolic equation

$$\phi \frac{\partial S}{\partial t} - \varepsilon \nabla \cdot \big(\mathbf{D}(S) \cdot \nabla S\big) = 0, \qquad S(\mathbf{x}, 0) = \bar{S}^{n+1}(\mathbf{x}),$$

and we have computed the saturation at the next time level. To solve the parabolic equation we use a finite difference method. For a complete introduction to operator splitting algorithms see for example Espedal and Karlsen [4] and references therein. For theory and convergence results relevant for the present application, we refer to Holden, Karlsen and Lie [7,6].

The numerical solution method for the hyperbolic equation will be our main concern in this article, and will be described in detail in subsequent sections.

3.1. Averaged Multivalued Solutions for Hyperbolic Conservation Laws

We now give a description of the transport-collapse operator, according to Brenier [2]. If the solution of (4) remains smooth, the initial value problem can be solved by the method of characteristics. Consider characteristic curves $\Gamma = (\mathbf{x}(\tau), t(\tau))$ satisfying

$$\dot{\mathbf{x}} = f(S)\mathbf{v}, \qquad \mathbf{x}(0) = \mathbf{x}_0, \qquad \dot{t} = \phi, \qquad t(0) = 0, \tag{5}$$

where $\dot{}$ denotes derivation with respect to τ. Now $\dot{S} = \phi S_t + \mathbf{v} \cdot \nabla f(S) = 0$. Due to nonlinearity, the problem usually has non-classical solutions resulting from crossing of characteristics. When discontinuities develop, the method breaks down. If the transportation of the initial data along the characteristic curves is continued, a multivalued solution is obtained instead of the correct entropy weak solution. However, as we will see, a proper vertical averaging of this solution gives an approximation to the entropy weak solution.

The central idea behind the collapse transform is to average the values of the multivalued solution for each point $\mathbf{x}$ in order to satisfy the following conservation principle [2]: *The algebraic length between the graph of the approximated entropy solution and the multivalued solution at any time t must be zero.* This gives us that the appropriate averaged solution $\hat{u}$ at a point $\mathbf{x}$ can be written as $\hat{S} = \sum_{i=0}^{2p}(-1)^i S_i$, where $\{S_i\}$ denote the values of the multivalued solution at a point $\mathbf{x}$, ordered such that $S_0 \leq \cdots \leq S_{2p}$. Note that $1 + 2p$ must be an odd number since the multivalued solution is obtained by a continuous transformation from the graph of the initial data. The transport-collapse transform is illustrated in Figure 1. For additional theory and convergence results, see [2].

We now have a formula for estimating the entropy solution from the multivalued solution. Our next task is to obtain such a multivalued solution by transporting the initial data along characteristic curves. To this end, we will apply a novel approach based on a reformulation of the hyperbolic problem in an Eulerian framework. This is the subject of the next section.

3.2. Stationary Formulation – Time-of-Flight

Following the approach of Karlsen, Lie and Risebro [8], which in turn was inspired by the level set idea of Osher and Sethian [10], we consider the k - level set of the saturation function $S(\mathbf{x}, t)$. The k - level curve (set) is defined as $\Xi_k(t) = \{\mathbf{x} : S(\mathbf{x}, t) = k\}$. According to the characteristic equations (5), the curve will move with a characteristic speed given by $\mathbf{V}(\mathbf{x}) = \frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x})f'(k)/\phi(\mathbf{x})$. Let $T_k(\mathbf{x})$ be the time $\Xi_k(t)$ crosses the point $\mathbf{x}$. The arrival time function T_k satisfies the arrival time equation $F(\mathbf{x})|\nabla T_k| = 1$, where $F(\mathbf{x})$ is the outward normal velocity of the moving curve. Now $F(\mathbf{x}) = \mathbf{V}(\mathbf{x}) \cdot \mathbf{n} =$

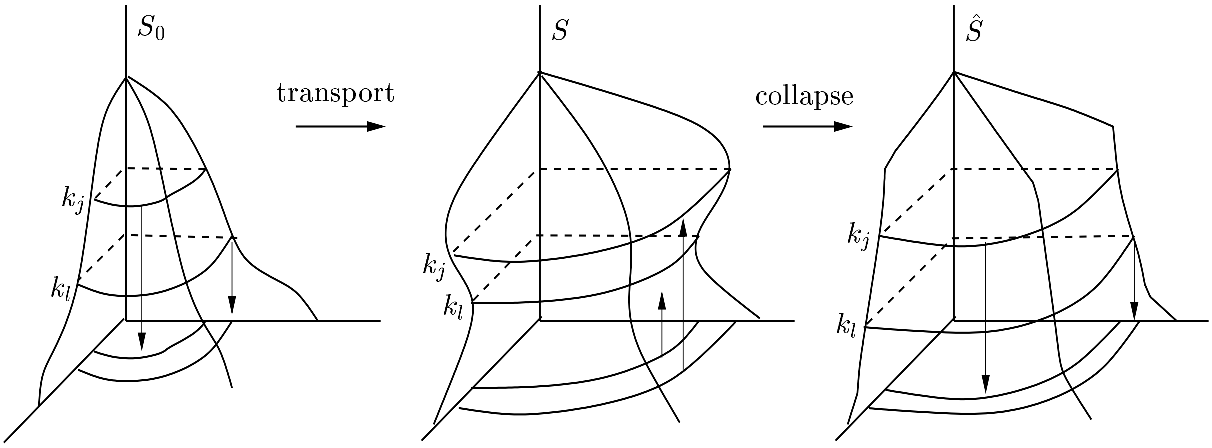

Figure 1. Illustration of the transport-collapse transform in two dimensions. Level curves corresponding to solution values k_l and k_j are shown in the x,y-plane.

$\mathbf{V}(\mathbf{x}) \cdot \nabla T_k / |\nabla T_k|$ and thus

$$\nabla T_k \cdot \mathbf{v}(\mathbf{x}) = \frac{\phi(\mathbf{x})}{f'(k)}, \qquad f'(k) > 0.$$

By defining $T = T_k f'(k)$, the equation becomes

$$\nabla T \cdot \mathbf{v}(\mathbf{x}) = \phi(\mathbf{x}), \tag{6}$$

as shown in [8]. The evolution problem for the propagating front has now been reformulated in an Eulerian framework. In the following section we give a description of the discretization of (6) and an optimal reordering of the unknowns that gives a very efficient solution procedure. This method resembles the fast marching method developed by Tsitsiklis [12], Helmsen et al. [5], and Sethian [11].

Equation (6), which in [8] was called the *arrival time equation*, is equivalent to the equation for the *time-of-flight* coordinate τ used in many modern streamline simulators, $\frac{d\tau}{ds} = \frac{\phi}{|\mathbf{v}|}$, see, e.g., [9]. However, as opposed to streamline stimulators that compute the time-of-flight on one-dimensional pathlines, we compute the time-of-flight on a *fixed* grid in two or three spatial dimensions. This way we loose the subgrid resolution in streamline methods, but avoid the complexity inherent in distributing streamlines to all blocks in the geological model.

3.3. The Fast Marching Method

In the following we assume that the initial function S_0 is monotone and that the velocity field is irrotational, so that the level curves of the saturation function never pass a point twice. This is not a restrictive assumption for the reservoir problem, since the flow will be directed from injection wells towards production wells. Information will flow in one direction, from regions with smaller arrival times to regions with larger arrival times. Hence, the arrival time at a grid point can only depend on points having smaller values.

This fact motivates an upwind discretization of the arrival time equation (6). Following [8], we apply a first-order 9-point scheme based on linear approximation of streamlines, locally through each grid point.

Based on the above observations it is possible to find an optimal ordering of the grid points and compute the solution of (6) very rapidly without iteration.

Following [8], we assume that the solution has been computed for a set of nodes, which we denote alive nodes $\mathcal{A}$. Candidate nodes for the next update are neighboring nodes of the alive nodes and are entitled narrow-band nodes $\mathcal{N}$. An estimated arrival time has been computed for all these nodes from the discretization formula. The remaining nodes in the grid are called far-away nodes $\mathcal{F}$.

To advance the solution outwards, we freeze the T-value at the narrow-band node with the least arrival time. Since no node point can be affected by points having greater arrival times, the T value at this node must be correct. The node is, consequently, removed from the narrow band $\mathcal{N}$ and added to $\mathcal{A}$. The next step is to update the values at all neighboring nodes that are not already alive. If a neighbor is far-away it must be added to the set of narrow-band nodes $\mathcal{N}$. Since recomputing T-values at neighboring downwind points cannot yield a value smaller than the value at the alive nodes, we can continue to move forward.

The efficiency of the algorithm strongly depends upon how fast we can find the narrow-band node with the least arrival time. By storing the narrow-band nodes in a *complete binary tree*, from smallest to largest arrival times, the node with the smallest arrival time can be removed in $\mathcal{O}(1)$ operations. The far-away nodes are inserted in the binary three in $\mathcal{O}(\log S_{\mathcal{N}})$ operations if $S_{\mathcal{N}}$ is the size of the narrow band. If the total number of grid points is N, then $S_{\mathcal{N}}$ is typically proportional to $\sqrt{N}$ and the constant of proportionality depends on the geometric complexity of the flow. Thus the method is $\mathcal{O}(N \log N)$.

3.4. Initialization

We now describe how to determine the initial sets $\mathcal{A}_j$, $\mathcal{N}_j$ and $\mathcal{F}_j$, for each level set k_j, $j = 1, \ldots, L$, for monotone and decreasing initial data. We will also introduce a method for updating the arrival times in the narrow-band nodes adjacent to the initial position of the front.

If we start the algorithm at time $t = 0$, then $T_k(\mathbf{x}) = 0$ for all $\mathbf{x} \in \Xi_k(0)$. Since we have assumed that the fronts always will move from regions with larger values of S to regions with smaller values of S, we can define the alive nodes as $\mathcal{A}_j = \{n : S_0(\mathbf{x}^n) \geq k_j\}$ and the corresponding set of narrow-band nodes as $\mathcal{N}_j = \{n : S_0(\mathbf{x}^n) < k_j,\ \mathcal{B}_n \cap \mathcal{A}_j \neq \emptyset\}$, for all level sets, $j = 1, \ldots, L$, where $\mathcal{B}_n$ is the set of the 8 neighboring nodes to n. The set of far-away nodes, $\mathcal{F}_j$, are the remaining nodes in the grid.

In order to start the fast marching algorithm for each level curve, we need to generate boundary data for the arrival time equation. That is, to find the arrival times in the initial narrow-band nodes corresponding to each curve. An approximate way to do this would be to let $T_{k_j}(n) = 0$, if $n \in N_j$. It is possible to find a better estimate by integrating the arrival time equation. Assume that the total Darcy velocity in $\mathbf{x}^m$ is $\mathbf{v}^m$. The intersection, $\mathbf{y}^i$, between the streamline γ and level surface $\Xi_{k_j}(0)$ is given by

$$\mathbf{y}^i = \{\mathbf{x} \in \mathbb{R}^2 : \mathbf{x} = \mathbf{x}^m - \mathbf{v}^m \theta^m,\ S(\mathbf{x}) = k\}.$$

This is a backward intersection for the narrow-band nodes adjacent to the front and a forward intersection for the alive nodes adjacent to the front. Since the S-value at $\mathbf{y}^i$ is k, the distance between $\mathbf{x}^m$ and $\mathbf{y}^i$ can be approximated by interpolation from saturation values in neighboring nodes.

If the initial saturation is assumed to be smooth, this interpolation will not result in large approximation errors. If it is discontinuous, we will have an error of order $\mathcal{O}(\Delta x)$, where Δx is a measure of the grid spacing.

Since $\mathbf{y}^i$ is a point on the boundary, $T_k(\mathbf{y}^i) = 0$. The arrival time in a narrow-band node $\mathbf{x}^m$ can now be computed by integrating the arrival time equation (6) along the streamline. By use of the definition $T = T_k f'(k)$, we can write

$$T^m = \frac{\phi^m |\mathbf{x}^m - \mathbf{y}^i|}{|\mathbf{v}^m|}. \tag{7}$$

To obtain correct interpolation values, we also need to estimate the arrival times in the alive nodes adjacent to the narrow-band nodes. The approximation formula for the arrival times in these nodes is the same as the formula (7), but with negative sign.

To construct the solution for times greater than $t = 0$, we can now apply the fast marching algorithm to each narrow band. If the final time is t^{final}, the final arrival time for each narrow band corresponding to level set k is $T^{\text{final}} = t^{\text{final}} f'(k)$.

3.5. Transported Solution

When the arrival time equation is solved to a prescribed final time, fronts corresponding to larger values of S may have crossed fronts corresponding to smaller values. This corresponds to the development of shocks, as illustrated in Figure 1. However, if we can find the multivalued solution, an approximation to the correct entropy weak solution can be obtained by means of the collapse transform [2], since the level curves have been transported according to the characteristic speed.

To build the (possibly multivalued) solution, after the level curves are evolved to a prescribed final time, we consider the position of the curves. Let L be the number of evolved level sets, so that $0 = k_0 < k_1 < \cdots < k_L = 1$. If a node $\mathbf{x}^m$ lies between one level curve Ξ_j and the next Ξ_{j+1}, that is, the node is an alive node for one and only one of the level curves, we assign a value to the node, which is the average of the level values, i.e., $S(\mathbf{x}^m) = (k_j + k_{j+1})/2$.

The procedure with transporting and collapsing the solution can be repeated during the time-step Δt to obtain a better approximation of the entropy weak solution. Too small time-steps $\Delta t^n = \Delta t / N^{\Delta t}$ will, however, cause problems since the final time $t^{\text{final}} = \Delta t^n$ in the fast marching algorithm will be reached too soon. That is, some level sets will not have time to pass any grid nodes. A more complete description of the method will be given in a forthcoming paper, see also [1].

4. NUMERICAL EXPERIMENTS

In this section we present a numerical example. We consider a standard quarter five-spot test problem for a heterogeneous reservoir. The viscosity ratio is $\mu_n : \mu_w = 8 : 1$. Wells are represented as point sources, and we use a no-flow boundary condition.

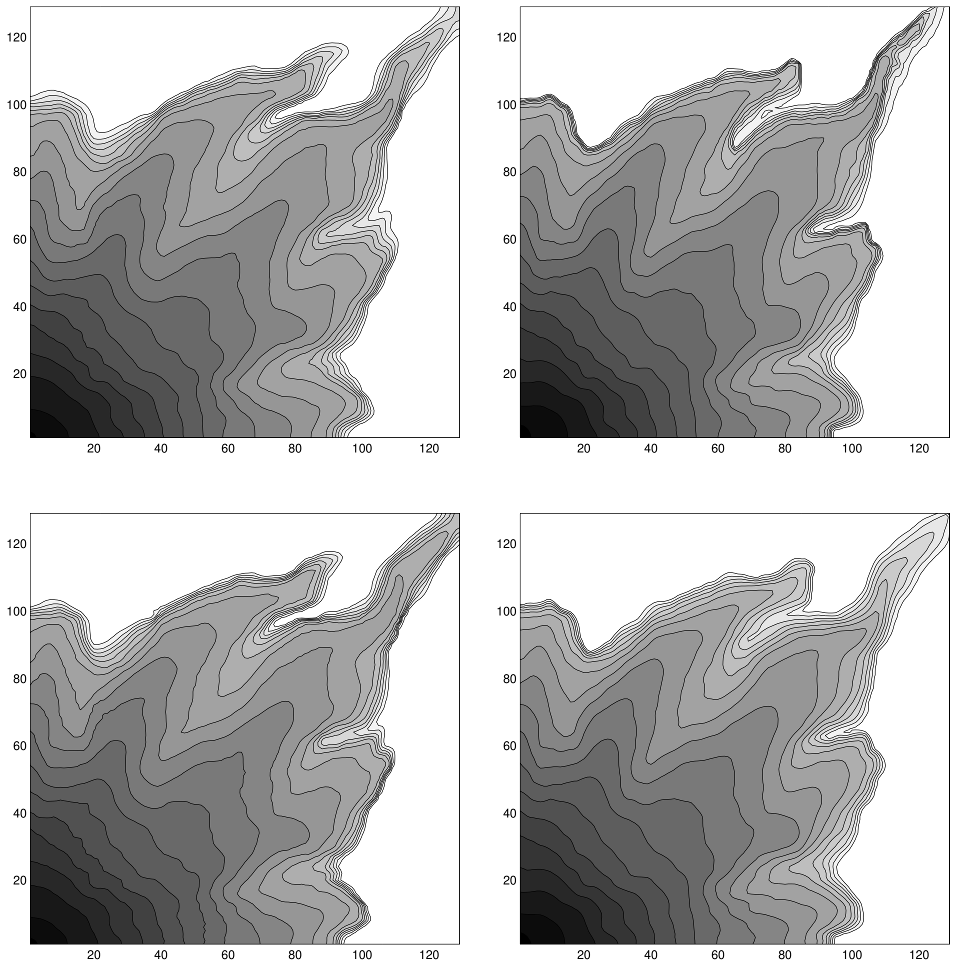

Figure 2. Saturation profiles for a quarter five-spot test problem. TC-method/CFL=13 (upper-left), FT-method/CFL=16 (upper-right), TC-method/CFL=6.5 (lower-left), and FT-method/CFL=4 (lower-right).

Since the velocity field is slowly varying we use only one initial pressure update. The diffusion parameter is set to $\varepsilon = 0.008$ and we use a 129×129 computational grid. Figure 2 shows saturation profiles at time $t = 0.3$ computed by the method presented herein, which is referred to as the *TC-method*. For comparison, we also show saturation profiles computed by an unconditionally stable front-tracking simulator [7,6] (based on dimensional splitting), which is referred to as the *FT-method*. In a forthcoming paper, we will conduct a more thorough investigation of the TC-method, including a comparison with a streamline method. The two plots in the upper-row of Figure 2 show the results of the TC-method (left) and the FT-method with large CFL numbers (respectively 13 and 16). In terms of numerical diffusion, it is well known that the FT-method performs at its best when the CFL-number is large, see, e.g., [7,6]. However, if the CFL-number is too large there is a dimensional splitting error resulting in a loss of "symmetry" in the solution. This is evident in the upper-right plot in Figure 2. On the other hand, we see that the TC-solution (upper-left plot in Figure 2) is slightly more diffusive than the FT-solution. However, it is free of symmetry error. To get rid of the symmetry error in the FT-solution we have to reduce the CFL-number. The lower-right plot in Figure 2 shows the FT-solution with CFL-number 4. The "symmetry" error has disappeared but at the expense of a more diffusive solution. The corresponding TC-solution (CFL-number 6.5) in the lower-left plot in Figure 2 is slightly less diffusive than the FT solution and is still free of symmetry error. The overall conclusions is that the TC-method compares well with the FT-method. A similar conclusion holds for comparisons with some other numerical methods as well (results will be presented elsewhere). Some of the advantages with the TC-method are: (i) it is computationally fast, (ii) it is accurate (e.g., free of "symmetry" error), and (iii) it is easy to implement (also in three dimensions).

REFERENCES

1. I. Berre, A fast marching method for hyperbolic conservation laws and application in reservoir simulation, cand. scient. thesis, Department of Mathematics, University of Bergen, 2001.
2. Y. Brenier, Averaged multivalued solutions for scalar conservation laws, *SIAM J. Numer. Anal.*, 21(6):1013–1037, 1984.
3. G. Chavent and J. Jaffre. *Mathematical models and finite elements for reservoir simulation*, volume 17 of *Studies in mathematics and its applications*. North Holland, Amsterdam, 1986.
4. M. S. Espedal and K. H. Karlsen. Numerical solution of reservoir flow models based on large time step operator splitting algorithms. In volume 1734 of *Lecture Notes in Mathematics*, pages 9–77. Springer, Berlin, 2000.
5. J. Helmsen, E.G. Puckett, P. Coella, and M. Dorr, Two new methods for simulating photolithography development in 3D, *Proc. SPIE* 2726:253–261, 1996.
6. H. Holden, K. H. Karlsen, and K.-A. Lie, Operator splitting methods for degenerate convection-diffusion equations I: convergence and entropy estimates, CMS Conf. Proc., 29, Amer. Math. Soc., Providence, RI, 2000.
7. ———, Operator splitting methods for degenerate convection-diffusion equations II: numerical examples with emphasis on reservoir simulation and sedimentation, *Computational Geosciences* 4(4):287-322, 2000.
8. K. H. Karlsen, K.-A. Lie, and N. H. Risebro, A fast marching method for reservoir simulation, *Computational Geosciences* 4(2):185–206, 2000.
9. M. J. King and A. D. Datta-Gupta. Streamline simulation: a current perspective. *In Situ (Special Issue on Reservoir Simulation)*, 22(1):91–140, 1998.
10. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
11. J.A. Sethian, Fast marching level set methods for three-dimensional photolithography development, *Proc. SPIE* 2726:261–272, 1996.
12. J.N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, *IEEE Transactions on Automatic Control* 40:1528–1538, 1995.