OPERATOR SPLITTING METHODS FOR DEGENERATE CONVECTION–DIFFUSION EQUATIONS II: NUMERICAL EXAMPLES WITH EMPHASIS ON RESERVOIR SIMULATION AND SEDIMENTATION

HELGE HOLDEN, KENNETH HVISTENDAHL KARLSEN, AND KNUT–ANDREAS LIE

Abstract. We present an accurate numerical method for a large class of scalar, strongly degenerate convection-diffusion equations. Important subclasses are hyperbolic conservation laws, porous medium type equations, two-phase reservoir flow equations, and strongly degenerate equations coming from the recent theory of sedimentation-consolidation processes. The method is based on splitting the convective and the diffusive terms. The nonlinear, convective part is solved using front tracking and dimensional splitting, while the nonlinear diffusion part is solved by an implicit–explicit finite difference scheme. In addition, one version of the implemented operator splitting method has a mechanism built in for detecting and correcting unphysical entropy loss, which may occur when the time step is large. This mechanism helps us gain a large time step ability for practical computations. A detailed convergence analysis of the operator splitting method was given in Part I. Here we present numerical experiments with the method for examples modelling secondary oil recovery and sedimentation-consolidation processes. We demonstrate that the splitting method resolves sharp gradients accurately, may use large time steps, has first order convergence, exhibits small grid orientation effects, has small mass balance errors, and is rather efficient.

1. Introduction

We study numerically a large class of convection–diffusion equations of the type

$$u_t + \sum_{j=1}^{m} V_j(x) f_j(u)_{x_j} = \varepsilon \sum_{j=1}^{m} \big(K_j(x) A_j(u)_{x_j}\big)_{x_j}, \qquad u(x,0) = u_0(x), \tag{1}$$

for all $(x,t) \in Q_T = \mathbb{R}^m \times [0,T]$. Here $u = u(x,t)$ is the unknown function that is sought, while $f = (f_1, \dots, f_m), K = (K_1, \dots, K_m) > 0, A = (A_1, \dots, A_m)$, and u_0 are given, sufficiently regular functions. The velocity field $V = (V_1, \dots, V_m)$ is assumed to be divergence free. Therefore the convective part of (1) is written on transport (or non-conservative) form.

For the nonlinear diffusion coefficient $A = A(s)$ we require only that

$$A'(s) \geq 0 \text{ for all } s,$$

thus making (1) a so-called *strongly degenerate* (sometimes we use the simpler term degenerate) parabolic equation, which means that hyperbolic conservation laws of the type

$$u_t + \nabla \cdot f(u) = 0$$

constitute an important subclass of (1). Other important subclasses include one-point degenerate porous media equations [43], two-point degenerate convection-diffusion equations arising in models for two-phase flow in porous media [10], and strongly degenerate parabolic equations coming from the recent theory of sedimentation-consolidation processes [9].

Nonlinear partial differential equations of hyperbolic-parabolic type such as (1) possess solutions which may exhibit quite complex behaviour (such as formation of sharp transitions and singularities) in a small region in space (and time) and this makes them particularly hard to solve numerically. It is therefore of considerable interest to construct and analyse numerical methods

Date: November 30, 1999.

Key words and phrases. Degenerate convection–diffusion equations, operator splitting, front tracking, two-phase flow, porous media, sedimentation-consolidation.

for such mixed type equations that work "uniformly" in the diffusion coefficient $A' \geq 0$, and as such are able to resolve sharp transitions and singularities in the solutions.

If (1) is allowed to degenerate at certain points, that is, $A'(u) = 0$ for some values of u, solutions are not necessarily smooth (but typically continuous) and weak solutions must be sought. On the other hand, if $A'(u)$ is zero on a set of positive measure, (weak) solutions may be discontinuous and are not uniquely determined by their initial data, as can be easily inferred from the maximum principle and what is known about the hyperbolic conservation law. Consequently, additional admissibility criteria—*entropy conditions*—must be imposed to single out the physically correct solution. It is outside the scope of this paper to go into details about the notion of entropy solutions for degenerate parabolic equations and the current state of the corresponding mathematical theory. Instead, we refer to the overview given in our companion paper [31], hereafter denoted as Part I, and the references cited therein.

Perhaps somewhat surprisingly, there have been few attempts up to very recently (confer the list of references given below) to develop a systematic treatment of such nonlinear partial differential equations within a unified mathematical framework. In fact, the construction and analysis of numerical methods for first order hyperbolic and second order parabolic problems are usually considered as separate subject areas. A partial aim of our work is to demonstrate that it is possible to give a coherent treatment of numerical methods for such a large class of nonlinear partial differential equations. Our ultimate goal is to have the same mathematical/numerical framework for the hyperbolic case ($A' \equiv 0$) and the parabolic case ($A' > 0$) as well as for the mixed hyperbolic-parabolic case ($A' \geq 0$). In the present paper and Part I (see also [26, 27, 21, 6]), we are concerned with operator splitting methods. For related work on numerical methods for strongly degenerate parabolic equations, see Evje and Karlsen [23, 24, 25, 22] for upwind difference schemes; Kurganov and Tadmor [37] for central difference schemes; and Bouchut, Guarguaglini, and Natalini [5] for kinetic BGK schemes. In passing, we also mention the work by Afif and Amaziane [1, 2, 3] which deals with finite volume methods for two-point degenerate convection-diffusion equations with a monotonely increasing flux function (for which *weak* solutions are unique).

It has become a well accepted practice to utilize conservative methods satisfying a discrete entropy condition when solving first order hyperbolic conservation laws. The reason for this is of course that if such methods converge, they do so to the unique physically relevant weak solution of the conservation law, whereas merely conservative methods may converge to a wrong solution. It turns out that the situation is similar for strongly degenerate parabolic equations. When solving such equations, one should always employ (conservative) numerical methods that satisfy a discrete entropy principle [23, 24]. To back up this claim, we refer the reader to an example presented by Evje and Karlsen [23] which shows numerically that a "standard" finite difference scheme for uniformly parabolic equations may fail to produce correct entropy solutions in the case of strong degeneracy. Although the scheme is convergent also in the case of strong degeneracy, it does not seem to satisfy a discrete entropy condition.

In Part I, we outlined various operator splitting methods for (1) and presented a convergence analysis within the entropy solution framework for one specific method. In particular, L^1 convergence results and precise entropy estimates were provided. The entropy estimates allowed us to conclude that any convergent sequence of operator splitting approximations must converge to an entropy solution of (1), thereby showing that the operator splitting methods can be used with confidence as numerical methods for strongly degenerate parabolic equations such as (1).

The operator splitting methods described in Part I are constructed to accurately resolve sharp fronts or transitions. A key feature of the methods is the use of an unconditionally stable, front-tracking method for hyperbolic sub-steps. As a result, the splitting methods can in principle use very large time steps. In this paper we apply these splitting methods successfully to several test problems arising in applications. The emphasis is on reservoir simulation, but we also include two examples from the simulation of sedimentation-consolidation processes, and two mixed hyperbolic-parabolic model problems in multidimensions.

Although our methods allow for large time steps, there is a trade-off in that large time steps mean great computer efficiency but tend to produce too much diffusion near steep gradients. Our aim is to resolve sharp fronts, and there is a complicated interplay between the hyperbolic operator and the parabolic operator. To compensate for the entropy loss that takes place between the Oleinik solution of the hyperbolic equation and the diffusive nature of the parabolic equation, we apply a corrected operator splitting technique introduced by Karlsen and Risebro [35] (see Section 2 for more details) which helps us to gain the large time step ability for practical computations. This is demonstrated for a one-dimensional Buckley–Leverett equation, for which we also demonstrate that the splitting method is more efficient than standard finite difference methods.

In multidimensions, large splitting steps also tend to produce lack of mass conservation. We study this effect for a classical test case from oil reservoir simulation—the quarter five-spot with water injection at $(0,0)$ and production at $(1,1)$. Earlier studies [34] have demonstrated that lack of mass conservation is very small before water breaks through in the production well, also for large time steps (CFL numbers of magnitude 10). Here, we observe negligible lack of mass conservation for small CFL numbers (around 1.0) also after water breakthrough. For CFL numbers above 1.0, the lack of mass conservation increases drastically as water breaks through due to the singularity in the Darcy velocity field at the production well. A sensible solution to this problem is to keep high CFL numbers until breakthrough, at which point the CFL number is reduced to unity.

Dimensional splitting techniques are known to exhibit certain grid orientation effects. For viscosity ratios close to one, little such effects are observed for the quarter five-spot case. However, for viscosity ratios of $1:10$, these effects are noticeable, since in this case the effective flux function is linear and contains no self-sharpening effect to counteract splitting errors.

To support the theoretical convergence analysis in Part I, we present numerical convergence studies that indicate first order convergence.

Our final example regarding flow in porous media shows two highly heterogeneous quarter five-spots (permeability fields generated as the exponential of Gaussian fields) where we demonstrate that the method resolves viscous fingering accurately.

Our application is taken from sedimentation–consolidation. Here the degeneracy of the diffusive term is considerably more complicated than in the cases we studied for reservoir simulation. Our two examples are taken from Bustos et al. [9] and describe the settling of a flocculated suspensions in a one-dimensional idealized continuous thickener (ICT). An unusual feature of the governing equation is its mixed hyperbolic-parabolic nature, where the mixed type nature corresponds to the interface between the compression zone, where the so-called solid effective stress varies, and the hindered settling zone, in which this quantity is assumed to be constant or to vanish. Above the interface region the equation is hyperbolic, below it is parabolic, and this interface is not known a priori. From conservation of mass and momentum and a constitute law one obtains the model (26). Our operator splitting technique is quite effective on this problem.

Finally, we compare our method with the recent second-order method of Kurganov and Tadmor [37] on a multidimensional Burgers' equation with a strongly degenerate diffusion term, see (28). For this problem operator splitting is considerably more efficient. In our final example we keep the strongly degenerate diffusive term, but introduce in addition a variable velocity field. The behavior in this case is highly involved with a complicated interaction between parabolic and hyperbolic regions. We demonstrate first order convergence, once more, and discuss the error mechanisms in the method.

In closing, we mention that Dahle, Espedal, Ewing, and collaborators [12, 15, 13, 14, 20] as well as Dawson, Wheeler, and collaborators [16, 17, 18, 19, 44] are using operator splitting algorithms similar to ours. We refer to the lecture notes by Espedal and Karlsen [21] for a general introduction to operator splitting algorithms and a long list of relevant references as well as a discussion of how the splitting algorithms in the literature relate to ours.

Acknowledgement. We thank Magne Espedal, Vidar Haugse, and Nils Henrik Risebro for valuable discussions. Our pressure solver was coded by Johnny Frøyen and Vidar Haugse. The permeability fields were generated by Håkon Tjelmeland. Karlsen would like to thank Raimund Bürger and Steinar Evje for discussions and collaboration on the subject of degenerate parabolic equations and their applications to sedimentation-consolidation processes.

2. Operator Splitting Based on Front Tracking

We are primarily interested in capturing sharp gradient variations in the solution of (1). To this end, we use operator splitting to augment efficient and accurate methods for conservation laws. Our basic building block is a front-tracking method [29, 39] for hyperbolic equations in one spatial dimension, where the nonlinear flux is approximated by its piecewise linear interpolant and one tracks an evolving piecewise-constant approximation. Front tracking is unconditionally stable, highly efficient, and gives exact resolution of discontinuities. Combining front tracking with appropriate methods for the diffusive forces, we can design methods that are unconditionally stable and deliver more than the standard resolution with surprisingly high efficiency.

2.1. **One-dimensional equations.** Let us for a moment consider the one-dimensional problem

$$u_t + V(x)f(u)_x = \varepsilon\big(K(x)A(u)_x\big)_x, \qquad u(x,0) = u_0(x). \tag{2}$$

The core of the splitting method lies in the solution of the hyperbolic problem

$$v_t + V(x,t)f(v)_x = 0, \qquad v(x,0) = v_0(x). \tag{3}$$

Henceforth, let $\mathcal{S}_t$ denote the front tracking solution operator. The second step of the operator splitting method consists of solving

$$w_t = (K(x)A(w)_x)_x, \qquad w(x,0) = w_0(x). \tag{4}$$

The approximate solution operator $\mathcal{H}_t$ can either be an explicit or an implicit finite difference scheme, or a finite element method. The corresponding operator splitting method reads

$$u(x, n\Delta t) \approx \big[\mathcal{H}_{\Delta t} \circ \pi \circ \mathcal{S}_{\Delta t}\big]^n \pi v_0, \tag{5}$$

where π is the (piecewise constant) projection operator defined with respect to an underlying fixed grid. The projection operator π has been included since the front tracking method is grid-independent and generally produces piecewise constant solutions on a nonuniform grid with arbitrary large mesh ratios. Reversing the operators (i.e., $\pi \circ \mathcal{S}_{\Delta t} \circ \mathcal{H}_{\Delta t}$) is not recommended since the self-sharpening effects in the hyperbolic step would tend to produce discontinuous solutions.

The operator splitting (5) will converge as the discretization parameters Δx and Δt tend to zero with $\Delta t/\Delta x$ fixed, see [26, 31] and below. Unfortunately, this does not guarantee that the quality of the solution is good for an arbitrary ratio between the splitting step and the spatial discretization. In general, large splitting steps tend to produce approximations that are too diffusive near steep fronts in the solution. The mechanism is as follows (assume for simplicity that $A'(u) > 0$): in the hyperbolic step, the solution is picked according to an entropy principle, which introduces a local linearization (or Oleinik convexification) of the flux function. The structure of (self-sharpening) parabolic fronts is in some sense controlled by the difference between the flux function and its local convexification. This flux residual is thrown away in the hyperbolic step, giving an unphysical loss of entropy that results in a lack of self sharpening.

To compensate for the entropy loss, a *corrected operator splitting* algorithm was introduced by Karlsen and Risebro [35] where the local flux residuals are included in the parabolic step (or possibly in a separate hyperbolic step). Consider a one-dimensional problem with a simple moving front that solves $u_t + f(u)_x = \varepsilon A(u)_{xx}$. In the hyperbolic step $v_t + f(v)_x = 0$, this front is represented by a discontinuity with left and right limits v_L and v_R. The *local* flux residual is identified as $f_{\text{res}} = f - f_c$, where f_c denotes the convex or concave envelope (depending on the entropy condition) of the flux f on the interval bounded by v_L and v_R. Then the new diffusion step is given by

$$w_t + f_{\text{res}}(w;x)_x = \varepsilon A(w)_{xx}.$$

In accordance with the discussion above, the residual flux f_{res} is defined to be nonzero only in a small domain around the point of discontinuity. Although the equation is on the same form as the original unsplit equation, the nonlinearity in the flux function is less severe since the advective part has been removed in the hyperbolic step. The remaining residual flux ensures that the front gets the correct amount of self sharpening. An example of the corrected operator splitting is given in Section 3.3.

2.2. **Multidimensional equations.** For the multidimensional problem (1), we are left with two possible approaches for splitting, once we have chosen to use the one-dimensional front tracking method as our basic building block. One choice is to split (1) into a a hyperbolic problem

$$v_t + \sum_{j=1}^{m} V_j(x) f_j(v)_{x_j} = 0, \qquad v(x,0) = v_0(x) \tag{6}$$

and a degenerate parabolic problem

$$w_t = \varepsilon \sum_{j=1}^{m} \big(K_j(x) A_j(w)_{x_j}\big)_{x_j}, \qquad w(x,0) = w_0(x). \tag{7}$$

Letting $\mathcal{S}_t$ denote the solution operator associated with (6) and $\mathcal{H}_t$ the solution operator associated with (7), we define the (semi-discrete) operator splitting solution as

$$u(x, n\Delta t) \approx \left[\mathcal{H}_{\Delta t} \circ \mathcal{S}_{\Delta t}\right]^n u_0. \tag{8}$$

However, for numerical computations exact solution operators have to be replaced by approximate ones. To this end, we introduce a regular grid in $\mathbb{R}^m$ with mesh size Δx and a piecewise constant projection operator onto this grid, denoted by π. To solve (7) we use a suitable finite difference method, which will be introduced below. For the hyperbolic step we can further use dimensional splitting to obtain a sequence of one-dimensional hyperbolic problems

$$v_t + V_j(x) f_j(v)_{x_j} = 0, \qquad v(x,0) = v_0(x), \tag{9}$$

which are solved by front tracking, see below. If $\mathcal{S}^j_t$ denotes the front tracking solution operator associated with (9) and $\mathcal{H}_t$ the finite difference solution operator associated with (7), we can compactly define the operator splitting solution as

$$u(x, n\Delta t) \approx \left[\mathcal{H}_{\Delta t} \circ \pi \circ \mathcal{S}^m_{\Delta t} \circ \cdots \circ \pi \circ \mathcal{S}^1_{\Delta t}\right]^n \pi u_0. \tag{10}$$

This way, the composite operator $\pi \circ \mathcal{S}^m_{\Delta t} \circ \cdots \circ \pi \circ \mathcal{S}^1_{\Delta t}$ may be viewed as an approximate solution operator (at time $t = \Delta t$) for the multidimensional hyperbolic problem (6). The use of dimensional splitting together with front tracking as a numerical method for (6) was first considered by Holden and Risebro [30] and later by Lie et al. [38, 40].

Algorithmically, our splitting goes as follows in two spatial dimensions:

Define $n = 0$ and get parameter Δt.
Compute initial approximation on grid $\{x_i, y_j\}$: $u(x,y,0) := \pi u_0(x,y)$.
While $n\Delta t < T$
  For each row $j = 1, \dots, M$ in the grid (i.e., $y_j \le y \le y_{j+1}$),
    Compute front tracking solution on row j up to $t = \Delta t$:

$$v_t + V_1(x; y_j) f_1(v)_x = 0, \quad v(x,0) = u(x,y,n\Delta t)$$

    Project solution back onto grid: $u(x,y,n\Delta t) := \pi^j_x v(x, \Delta t)$
  For each column $i = 1, \dots, N$ in the grid (i.e., $x_i \le x \le x_{i+1}$),
    Compute front tracking solution on column i up to $t = \Delta t$:

$$v_t + V_2(y; x_i) f_2(v)_y = 0, \quad v(y,0) = u(x,y,n\Delta t),$$

    Project solution back onto grid: $u(x,y,n\Delta t) := \pi^i_y v(y, \Delta t)$
  Solve parabolic problem on 2-D grid up to $t = \Delta t$:

$$w_t = \varepsilon \nabla\big(K(x,y) \nabla A(w)\big), \quad w(x,y,0) = u(x,y,n\Delta t).$$

  Increase time: $n := n + 1$
  Update solution: $u(n\Delta t, x, y) := w(x,y,\Delta t)$

The second splitting method first uses dimensional splitting on (1) to give a sequence of one-dimensional equations of the form (2) for each spatial direction. Then each of these equations are solved using the splitting (5). The overall splitting method takes the form

$$u(x, n\Delta t) \approx \left[\mathcal{H}^m_{\Delta t} \circ \pi \circ \mathcal{S}^m_{\Delta t} \circ \cdots \circ \mathcal{H}^1_{\Delta t} \circ \pi \circ \mathcal{S}^1_{\Delta t}\right]^n \pi u_0, \tag{11}$$

where $\mathcal{H}^j_{\Delta t}$ denotes the finite difference solution operator associated with the one-dimensional problem

$$w_t = \varepsilon\big(K_j(x)A_j(w)_{x_j}\big)_{x_j}, \qquad w(x,0) = w_0(x). \tag{12}$$

We will henceforth refer to the splitting (10) as OS and the splitting (11) as $\mathrm{OS}^{\mathrm{ds}}$. The first approach avoids dimensional splitting of diffusive forces and is therefore likely to have smaller splitting errors (see, e.g., Figure 8). On the other hand, $\mathrm{OS}^{\mathrm{ds}}$ will give faster solution of the parabolic steps. In case of an implicit discretization of the parabolic steps, $\mathrm{OS}^{\mathrm{ds}}$ involves solving tridiagonal linear systems by LU factorization, whereas OS typically gives $(2m+1)$-diagonal systems that must be solved by some iterative method (e.g., conjugated gradients). For explicit schemes, $\mathrm{OS}^{\mathrm{ds}}$ results in less numerical operations than OS. Moreover, the storage requirements for $\mathrm{OS}^{\mathrm{ds}}$ are less than for OS.

The second approach ($\mathrm{OS}^{\mathrm{ds}}$) opens up for the use of the corrected operator splitting idea, that is, instead of solving a parabolic step of the type (12), we solve

$$w_t + V_j(x,t) f_{j,\mathrm{res}}(w)_{x_j} = \varepsilon(K_j(x)A_j(w)_{x_j})_{x_j}, \qquad w(x,0) = w_0(x), \tag{13}$$

where $f_{j,\mathrm{res}}$ is a residual flux term determined from the jth directional hyperbolic solution, i.e., the solution of (9) (with suitable initial data). If $\mathcal{H}^{c,j}_{\Delta t}$ denotes the approximate solution operator of (13), the multidimensional corrected operator splitting reads

$$u(x, n\Delta t) \approx \left[\mathcal{H}^{c,m}_{\Delta t} \circ \pi \circ \mathcal{S}^m_{\Delta t} \circ \cdots \circ \mathcal{H}^{c,1}_{\Delta t} \circ \pi \circ \mathcal{S}^m_{\Delta t}\right]^n \pi u_0. \tag{14}$$

We will refer to this splitting as COS. As an alternative to (14), the residual fluxes can be included in an ad hoc, multidimensional fashion in (10); see Evje et al. [27] for more details and a discussion of different correction strategies. Since the algorithmic descriptions of (11) and (14) are very similar to the one that we gave for (10), we choose not to present them here.

2.3. **Front tracking.** Now that the operator splitting methods have been introduced, it is time to discuss the numerical methods used for each part in more detail. For the hyperbolic problems we use front tracking [29, 39]. Given an equation on the form (3), we first approximate the initial data by a step function πv_0 and the flux function and the velocity (which we for simplicity assume only to depend on x) by piecewise linear functions f^δ and V^σ. This way, the Cauchy problem is reduced to a series of Riemann problems. Since the flux function is piecewise linear, each Riemann problem is solved by a set of constant states separated by discontinuities (henceforth called fronts). A front with left and right state v^- and v^+ propagates according to the ordinary differential equation

$$\dot{x} = V^\sigma(x)\Big(\frac{f^\delta(v^+) - f^\delta(v^-)}{v^+ - v^-}\Big),$$

which can be solved explicitly if, for instance, V^σ is piecewise linear. The numerical method consists of tracking discontinuities as the time evolves. Each time two or more discontinuities interact, a new Riemann problem is solved. The front-tracking algorithm is illustrated in Figure 1. Except for the specification of πv_0, the method is grid-independent and has no intrinsic time step. The tracking of fronts is implemented using two linked lists of front objects; one list represents the fronts in x from left to right, and one list keeps track of the queue of possible collisions. Each front object typically contains the left and right states of each discontinuity, its Rankine–Hugoniot speed, start position (x_0, t_0), and collision point (x_c, t_c), see Risebro and Tveito [42] for further details on the data structure.

The virtue of front tracking is that Riemann problems can be solved exactly. Furthermore, since there is no intrinsic time step, no CFL condition is imposed. Extension of the front tracking method to multidimensions by dimensional splitting (as outlined above) is described in detail by Holden and Risebro [30] and Lie et al. [38, 40].

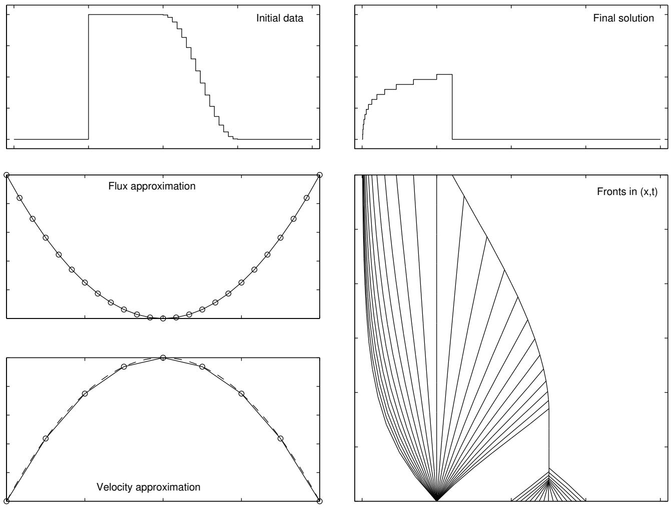


FIGURE 1. Illustration of the front tracking algorithm for a variable coefficient problem.

2.4. **Finite difference scheme.** In the parabolic step we solve (degenerate) equations

$$w_t = \varepsilon \sum_{j=1}^{m} \partial_{x_j}\big(K_j(x)\partial_{x_j} A_j(w)\big), \qquad w(x,0) = w_0(x),$$

for $m \geq 1$. To this end, we use an implicit–explicit finite difference method. Assume a mesh with a uniform spacing Δx in each spatial direction and a time step τ. Let $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{Z}^m$ be a set of indices and define e_j such that the jth component equals unity and all other components equal zero. Let W_α^n denote the approximate solution in grid cell α at time $t = n\tau$. Furthermore, let $\beta_\alpha = \varepsilon\tau K^j_{\alpha+e_j/2}/\Delta x^2$ and $A_\alpha^{j,n} = A_j(W_\alpha^n)$. Then the scheme reads

$$\begin{aligned} W_\alpha^{n+1} =& W_\alpha^n + \theta \sum_{j=1}^{m}\Big[\beta_\alpha^j\big(A_{\alpha+e_j}^{j,n} - A_\alpha^{j,n}\big) - \beta_{\alpha-e_j}^j\big(A_\alpha^{j,n} - A_{\alpha-e_j}^{j,n}\big)\Big] \\ &+ (1-\theta)\sum_{j=1}^{m}\Big[\beta_\alpha^j\big(A_{\alpha+e_j}^{j,n+1} - A_\alpha^{j,n+1}\big) - \beta_{\alpha-e_j}^j\big(A_\alpha^{j,n+1} - A_{\alpha-e_j}^{j,n+1}\big)\Big], \end{aligned} \tag{15}$$

where the parameter θ is in the interval from zero to one; $\theta = 0$ giving a fully implicit scheme and $\theta = 1$ a fully explicit scheme. The scheme is stable provided [31]

$$\theta \frac{\varepsilon\tau}{\Delta x^2} \max_j\big(\sup_x K_j(x) \sup_u A_j'(u)\big) \leq \frac{1}{2m}. \tag{16}$$

2.5. **Convergence of splitting methods.** Here we give a brief review of the main results from Part I. The (semi-discrete) splitting method (8) says that the solution u at time $t \in ((n-1)\Delta t, n\Delta t]$ of (1) is approximated by the piecewise constant time interpolation $u_{\Delta t}(t)$:

$$u_{\Delta t}(t) = \big[\mathcal{H}_{\Delta t} \circ \mathcal{S}_{\Delta t}\big]^n u_0, \quad t \in ((n-1)\Delta t, n\Delta t], \quad n = 1, \ldots, N, \quad N\Delta t = T,$$

or alternatively replace $\mathcal{S}_{\Delta t}$ by $\mathcal{S}_{\Delta t}^m \circ \cdots \circ \mathcal{S}_{\Delta t}^1$. For this approximation we prove the following result (Theorem 2.1 in Part I). The approximation $u_{\Delta t}$ converges along a subsequence as $\Delta t \to 0$ in $L^1_{loc}(Q_T)$ to an entropy solution u that satisfies $u(\cdot,t) \in L^1(\mathbb{R}^m) \cap L^\infty(\mathbb{R}^m) \cap BV(\mathbb{R}^m)$ uniformly in t and $u(\cdot,t)$ is uniformly L^1 Hölder continuous in time with exponent $1/2$. Here we assume that the initial data u_0 belongs to $L^1(\mathbb{R}^m) \cap L^\infty(\mathbb{R}^m) \cap BV(\mathbb{R}^m)$ and that the coefficients f_j, V_j, K_j, and A_j are sufficiently regular (see Part I for details).

The fully discrete splitting is obtained in (10) using front tracking and finite difference methods. In the front tracking method the flux function f_j and the velocity field V_j are replaced by f_j^δ and

V_j^σ, respectively, that is, by continuous, and piecewise linear approximations such that $f_j^\delta \to f_j$ and $V_j^\sigma \to V_j$ as $\delta \to 0$ and $\sigma \to 0$. Let $\mathcal{S}^j_{\delta,\sigma,t}$ denote the approximate solution operator. For the parabolic equation we use an implicit–explicit difference scheme (see equation (15)), and the corresponding solution operator is denoted by $\mathcal{H}_{\Delta x,t}$. Thus we obtain the fully discrete method

$$u_\eta(t) = \left[\mathcal{H}_{\Delta x,\Delta t} \circ \pi \circ \mathcal{S}^m_{\delta,\sigma,\Delta t} \circ \cdots \circ \pi \circ \mathcal{S}^1_{\delta,\sigma,\Delta t}\right]^n \pi u_0, \quad t \in ((n-1)\Delta t, n\Delta t],$$

where $n = 1, \dots, N$, $N\Delta t = T$, and the discretization parameters are collected in $\eta = (\Delta t, \sigma, \Delta x)$. In this case we prove (Theorem 2.2 in Part I) that u_η converges along a subsequence as $\eta \to 0$ in $L^1_{loc}(Q_T)$ to an entropy solution u that satisfies $u(\cdot,t) \in L^1(\mathbb{R}^m) \cap L^\infty(\mathbb{R}^m) \cap BV(\mathbb{R}^m)$ uniformly in t and $u(\cdot,t)$ is uniformly L^1 Hölder continuous in time with exponent $1/2$. Again we assume $u_0 \in L^1(\mathbb{R}^m) \cap L^\infty(\mathbb{R}^m) \cap BV(\mathbb{R}^m)$ as well as appropriate regularity of f_j, V_j, K_j, and A_j.

3. Reservoir Simulation

In the following we will give some numerical examples of the above operator splitting techniques applied to the simulation of a secondary recovery (water injection) process in oil reservoirs. First, however, we give a short introduction to the mathematical model which consists mainly of a saturation equation (25) and a pressure equation (24). The saturation equation is a parabolic equation on the form (1), generally possessing a two-point degeneracy. The operator splitting strategy will only be applied to stable displacement processes. Unstable displacement are discussed elsewhere [28]. Unless stated otherwise, we use the fully explicit scheme (equation (15) with $\theta = 1$) for the (degenerate) parabolic steps.

3.1. **Mathematical model.** Let s denote the water saturation (and $1 - s$ the oil saturation). Then the incompressible displacement of oil by water in a porous medium can be described by the following set of partial differential equations (given in dimensionless form):

$$\nabla \cdot V = q_1(x), \tag{17}$$

$$V = -K(x)\lambda(s)\big(\nabla p - \rho(s)\nabla h\big), \tag{18}$$

$$\phi(x)s_t + \nabla \cdot \big(f(s)V + f_g(s)K\nabla h\big) - \varepsilon\nabla \cdot \big(K(x)d(s)\nabla s\big) = q_2(x), \tag{19}$$

where q_1 and q_2 denote the injection/production wells and ε is a dimensionless scaling parameter. The total Darcy velocity is defined as $V = V_w + V_o$, where V_ℓ is Darcy velocity of phase $\ell = w, o$ (water, oil). Moreover, μ_ℓ are the viscosities of the fluids; $k_{r\ell}$ are the relative permeabilities; and $\lambda(s)$ denotes the total mobility of the phases,

$$\lambda(s) = \lambda_w(s) + \lambda_o(s),$$

where $\lambda_\ell = k_{r\ell}/\mu_\ell$ is the mobility of phase $\ell = w, o$. In (18), p is the global pressure [10]

$$p = \frac{1}{2}(p_w + p_o) + \frac{1}{2}\int_{s_c}^{s} \Big(\frac{\lambda_o - \lambda_w}{\lambda}\frac{\partial p_c}{\partial \xi}\Big)\, d\xi, \tag{20}$$

where p_w and p_o denote the pressure of water and oil, respectively, $p_c(x,\xi) = p_o - p_w$ is the capillary pressure function, and s_c is chosen such that $p_c(x, s_c) = 0$. In (18)–(19), we have

$$\rho(s) = g(\lambda_w\rho_w + \lambda_o\rho_o),$$

where ρ_ℓ is the density of phase $\ell = w, 0$, and

$$f_g(s) = (\rho_w - \rho_o)f(s)\lambda_o.$$

Equation (19)—the so-called saturation equation—is the fractional flow formulation of the mass balance equation for water. The fractional flow function $f(s)$, which is typically an S-shaped function of s, is given as

$$f(s) = \frac{\lambda_w(s)}{\lambda_w(s) + \lambda_o(s)}. \tag{21}$$

The diffusion function is given as

$$d(s) = -f(s)\lambda_o(s)\frac{\partial p_c}{\partial s} = -\frac{\lambda_w\lambda_n}{\lambda_w+\lambda_n}\frac{\partial p_c}{\partial s}, \tag{22}$$

where the derivative of the capillary pressure function $p_c(x,s)$ is assumed to be negative. We refer to [4, 10, 41] for a complete survey and justification of the model. For computational purposes, the analytical forms for the relative permeabilities are chosen as

$$k_{rw} = s^p, \qquad k_{ro} = (1-s)^q, \qquad p,q = 2 \text{ or } 3. \tag{23}$$

Assuming incompressible fluids, no gravity forces, and unit porosity, the model (17)–(19) simplifies to

$$-\nabla\big(K(x)\lambda(s)\nabla p\big) = q_1, \qquad V = -K(x)\lambda(s)\nabla p, \tag{24}$$

$$s_t + \nabla\cdot(V(x)f(s)) = \varepsilon\nabla\cdot(K(x)d(s)\nabla s) + q_2. \tag{25}$$

3.2. **Solution strategy.** To solve the fractional flow system (24)–(25), it is common to use some form of operator splitting. Here we will use the IMPES approach (implicit pressure, explicit saturation). Starting from the initial saturation distribution, we first solve the pressure equation (24) with the given total mobility. Subsequently, the total velocity field is calculated from the pressure solution, using Darcy's law. Then, the velocity field is assumed to be constant throughout the solution of the saturation equation (25). Having now established both pressure and saturation one time step forward in time, we can repeat the process, and so on.

The pressure equation is discretized using a nine-point scheme with upstream weighting and a harmonic average on the mobility. The resulting linear system for the pressure in the cell centers is solved by a multigrid method. A conservative flux consideration is used to determine the velocity at the centre of the grid cell boundaries. The saturation equation is solved numerically by one of the operator splitting methods introduced above.

3.3. **One-dimensional Buckley–Leverett problem.** The purpose of the first example is twofold; we wish to demonstrate the effect of the correction strategy introduced above and the efficiency of the operator splitting method. To this end, consider the one-dimensional saturation equation

$$u_t + \Big(\frac{u^2}{u^2+(1-u)^2}\Big)_x = \varepsilon\big(4u(1-u)u_x\big)_x$$

on the domain $x \in [0,1]$ with initial data $u_0(x) = 1$ for $x < 0.1$ and $u_0(x) = 0$ otherwise. We set $\varepsilon = 0.01$ and compute the solution at time $t = 0.5$ on a grid with 64 cells. To solve the modified parabolic step we use explicit finite differences with an Engquist–Osher discretization (see below) of the residual flux. Figure 2 shows the solution computed with one splitting step with and without correction and with 32 steps (CFL number 2.0) without correction. Figure 3 shows the (intermediate) solution after the first step and the induced local linearization of the flux. We see that by including the corresponding residual flux in the parabolic step, the shock layer is resolved (almost) correctly. For the rest of this section, the correction strategy can be considered as more of a curiosity that may come in handy if we wish to use large splitting steps.

The operator splitting is a bit complicated, and a natural question is whether something is gained by using it. To answer this question, we will make a comparison with two simple finite difference methods, that is, we compare with a central scheme

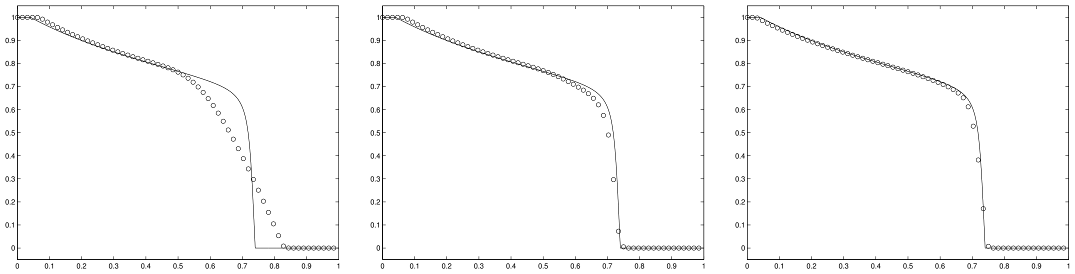
$$\frac{u_i^{n+1}-u_i^n}{\Delta t} + \frac{f(u_{i+1}^n)-f(u_{i-1}^n)}{2\Delta x} = \varepsilon\frac{A(u_{i+1}^n)-2A(u_i^n)+A(u_{i-1}^n)}{\Delta x^2}$$

and an Engquist–Osher scheme

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} + \frac{f^-(u_{i+1}^n)-f^-(u_i^n)}{\Delta x} + \frac{f^+(u_i^n)-f^+(u_{i-1}^n)}{\Delta x} = \varepsilon\frac{A(u_{i+1}^n)-2A(u_i^n)+A(u_{i-1}^n)}{\Delta x^2},$$

where

$$f^-(u) = \int^u \min(f'(s),0)\,ds, \qquad f^+(u) = \int^u \max(f'(s),0)\,ds.$$



L^1: 4.28e-02, L^∞: 2.98e-01 L^1: 1.23e-02, L^∞: 1.95e-01 L^1: 6.05e-03, L^∞: 2.02e-01

FIGURE 2. Solution and errors obtained with one splitting step (left), one step with correction (middle) and 32 steps (right). Reference solution is computed by the first-order upwind method on a fine grid.

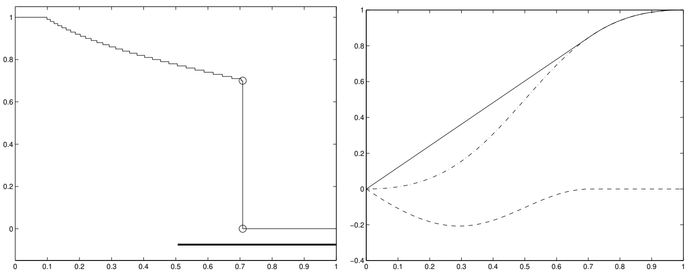


FIGURE 3. (Left) Solution after one hyperbolic step. The circles indicate the position where the residual flux is identified and the thick line the interval on which it is applied. (Right) Flux function (dash dot), local linearization (solid), and residual (dot).

The stability conditions for the two methods are

$$h \max |f'(u)| < 2\varepsilon \qquad \text{and} \qquad \max |f'(u)| \frac{\Delta t}{\Delta x} + 2\varepsilon \max |A'(u)| \frac{\Delta t}{\Delta x^2} \le 1,$$

respectively. Since the flux function is strictly increasing in this example, the Engquist–Osher scheme reduces to the upwind method.

A natural measure for comparing the methods is to plot error versus runtime. Figure 4 gives such a comparison for operator splitting, the central scheme, and the upwind scheme. The errors are measured relative to a reference solution computed by the upwind scheme on a grid with 2^{12} cells. The number of grid cells were 2^n for $n = 5, \ldots, 10$. For the operator splitting method we used a fixed CFL number 2.0, since this is typical for the multidimensional examples below. The central difference scheme has a rather severe stability restriction for small values of ε and was therefore not used for $\varepsilon = 0.001$. Comparing with simple difference schemes is fair, since our parabolic step uses such a scheme. We see that by isolating the advection in a separate step we gain slightly in efficiency.

For multidimensional problems we expect to gain even more in efficiency. During the hyperbolic step we focus the computations only on the regions where there are dynamics. Furthermore, a more sophisticated method could be used for the parabolic step, e.g., an implicit method (see (15)). Such methods would typically involve iterations to resolve the nonlinear diffusion, for which the separate hyperbolic step would provide a good initial guess (and reduce the number of iterations).

3.4. **Homogeneous quarter five-spot.** Our second example is the well-known quarter five-spot case where water is injected at the origin and oil is produced at the four points $(\pm 1, \pm 1)$. This five-well configuration is repeated periodically to infinity in all four quadrants in the plane. Symmetry reduces the test case to a problem on the unit square with an injection well placed at (0,0) and a production well at (1,1). Both wells have rates equal unity and we assume unit mobility ratio.

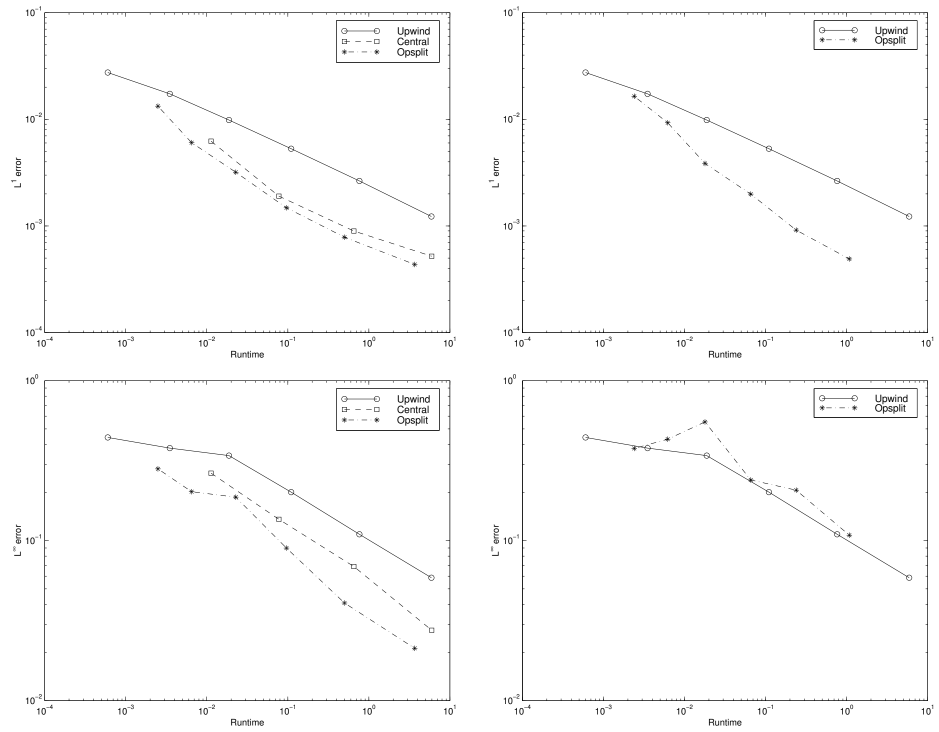


FIGURE 4. Error versus runtime for $\varepsilon = 0.01$ (left) and 0.001 (right). The errors are in L^1 (upper row) and L^∞ (lower row).

For the fractional flow function, we have used (21) and (23) with $p = q = 2$, i.e.,

$$f(s) = \frac{s^2/\mu_w}{s^2/\mu_w + (1-s)^2/\mu_o}.$$

Our splitting method for the saturation equation is unconditionally stable and thus allows for arbitrary large time steps. In [34] it was demonstrated that the method allows for relatively large (fixed) time steps. To achieve better accuracy we introduce adaptive steps aimed at keeping a fixed CFL number. This is achieved by measuring the maximum wave speed during each pass through one-dimensional front tracking. The wave speed is easily monitored in the routine computing possible wave interactions. Initially, the time steps are small due to the high velocities near the injection well. As the water front moves into the reservoir, the time steps increase and then start to decrease as the water front approaches the production well, see Figure 5.

Mass conservation. The use of dimensional splitting for the hyperbolic part of the saturation equation leads to lack of mass conservation. Karlsen et al. [34] observed small errors in the mass balance before water breakthrough. Figure 6 shows the observed mass error also after water breakthrough on a 129×129 grid with 32 pressure updates and $\varepsilon = 0.0005$. The corresponding plot of oil and water production is given in Figure 5. The mass error (or material balance error) is here defined as the difference between the net injection of water and the net increase of water remaining in the reservoir.

With fixed CFL number 1.0, the mass error remains below $6 \cdot 10^{-4}$ throughout the whole simulation. After the initial phase, the relative error is below 0.1%. For CFL=2.0, the relative mass error is around 0.12% before water breakthrough and then increases dramatically. A natural strategy would therefore be automatically to reduce the CFL number to 1.0 at water breakthrough. Applying this strategy with an initial CFL number 4.0, we observe that the relative mass error increases to around 0.27% before water breakthrough and then decreases to 0.15% at $t = 1.0$. (Time equals the number of pore volumes injected).

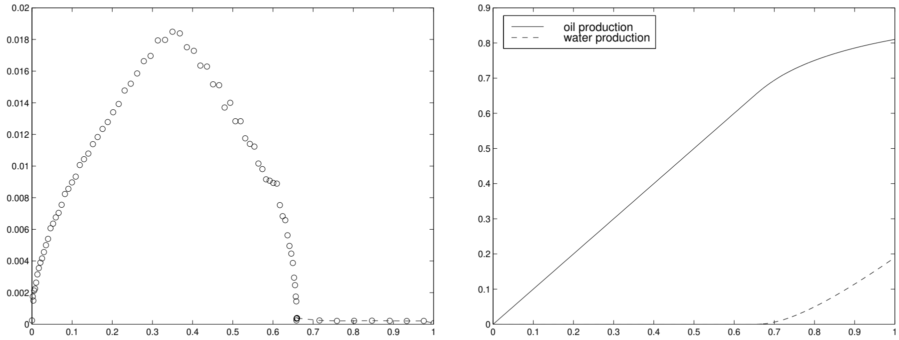


FIGURE 5. (Left) A typical plot of the time step in the saturation solver versus accumulated time. After water breakthrough only every two-hundred step is plotted. (Right) Accumulated production of oil and water.

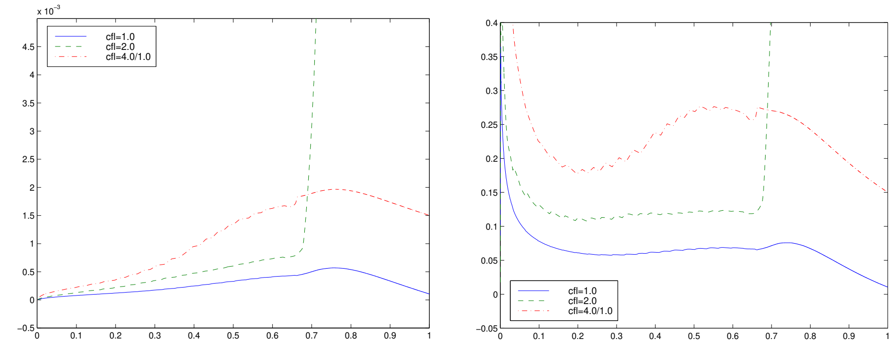


FIGURE 6. (Left) Mass error versus the cumulative number of pore volumes injected for three choices of CFL numbers. (Right) Percent mass error relative to the cumulative number of pore volumes injected.

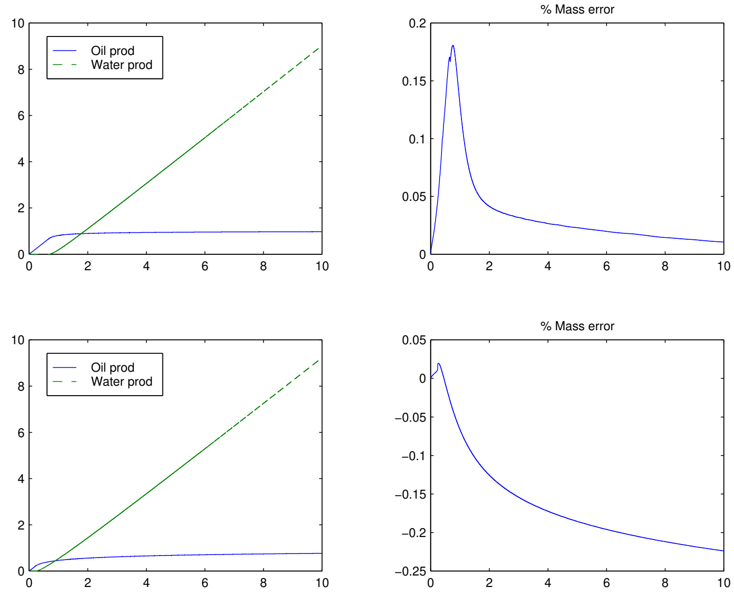


FIGURE 7. Mass error (relative to initial amount of oil) and oil and water production for viscosity ratios $\mu_o : \mu_w$ equal 1:1 (upper row) and 20:1 (lower row).

Figure 7 shows the result of simulations up to time $t = 10.0$ for two different viscosity ratios. The grid is 129×129, no capillary effects are included (i.e., $\varepsilon = 0.0$), and we have used 1 000 IMPES steps. As in Figure 6, the mass error is positive and increases before water breakthrough. To explain this phenomenon, we consider one row (or column) of the grid near the upper (or right) boundary. The oil bank trapped along the boundary (see e.g., Figure 9) will cause a flow of oil into the grid block that is connected to the well. Hence the water production is underestimated in each splitting step. This effect is also discussed in [28]. At the same time, the grid blocks near the producer will be partially filled with water (during the splitting step in the other direction), eventually giving an increased fractional flow in the well. Thus, the mass error decreases as the reservoir becomes more and more flooded and gradually stabilizes itself at a constant value (see Figure 7).

TABLE 1. *Measured errors for solution on an $N \times N$ grid relative to a $2N \times 2N$ grid at time $t = 0.45$ for four different values of ε. The upper half is for* OS *with explicit finite differences. The lower half is for* OS^{ds} *with implicit differences, i.e.,* (15) *with $m = 1$ and $\theta = 1$.*

N	$\varepsilon = 5 \cdot 10^{-2}$		$\varepsilon = 5 \cdot 10^{-3}$		$\varepsilon = 5 \cdot 10^{-4}$		$\varepsilon = 5 \cdot 10^{-5}$	
64	9.60e-03	—	6.05e-03	—	5.83e-03	—	5.85e-03	—
128	4.78e-03	1.01	3.38e-03	0.84	2.94e-03	0.99	3.32e-03	0.82
256	2.50e-03	0.94	1.88e-03	0.85	1.47e-03	1.00	1.54e-03	1.11
64	5.97e-03	—	5.81e-03	—	5.77e-03	—	5.84e-03	—
128	3.91e-03	0.61	3.08e-03	0.92	2.88e-03	1.00	3.32e-03	0.81
256	2.17e-03	0.85	2.01e-03	0.62	1.47e-03	0.97	1.52e-03	1.13

Convergence. Another interesting question is how fast the algorithm converges. Since the analytical solution is not known, we have chosen to investigate the self-convergence of our splitting methods. To this end, we define a sequence of meshes $M_{\Delta x}, M_{\Delta x/2}, \ldots$ and measure the L^1 error on mesh $M_{\Delta x}$ relative to $M_{\Delta x/2}$. Table 1 gives the errors and corresponding convergence rates measured by refining a 64×64 grid three times for four different values of ε. In all runs we used viscosity ratio equal one, only one IMPES step, and CFL number 8.0. (Due to restrictions on the number of unknowns caused by the multigrid method in our pressure solver, the actual computations were performed on grids with $2^N + 1$ cells in each direction and projected onto 2^N grids.)

TABLE 2. CPU *times for simulations on a 257×257 grid for three operator splittings.*

Method	θ	$\varepsilon = 5 \cdot 10^{-2}$	$\varepsilon = 5 \cdot 10^{-3}$	$\varepsilon = 5 \cdot 10^{-4}$	$\varepsilon = 5 \cdot 10^{-5}$
OS	1.0	301.0 sec	62.7 sec	36.5 sec	32.8 sec
$\mathrm{OS}^{\mathrm{ds}}$	0.0	55.9 sec	41.7 sec	37.7 sec	33.5 sec
$\mathrm{OS}^{\mathrm{ds}}$	1.0	73.4 sec	37.5 sec	32.9 sec	31.4 sec

Efficiency. Runtime is an important issue when choosing the optimal splitting strategy. Table 2 gives the runtimes on the 257×257 grid for the simulations reported in Table 1. In addition we have included runtimes for $\mathrm{OS}^{\mathrm{ds}}$ with an implicit diffusion solver. $\mathrm{OS}^{\mathrm{ds}}$ with explicit diffusion solver is the most efficient method, except for $\varepsilon = 0.05$, where the implicit solver is most efficient. The reader should keep in mind that in the computations presented in Table 2, we have chosen the time step in the implicit diffusion solver such that the L^1 error for the corresponding splitting method is roughly the same as for the splitting method with the explicit diffusion solver. This explains the somewhat disappointing (but not surprising) runtimes found in Table 2 with the implicit diffusion solver ($\theta = 0.0$).

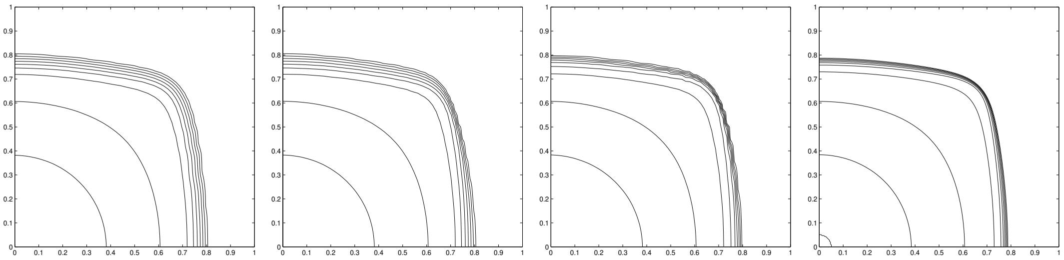


FIGURE 8. OS, OS$^{\mathrm{ds}}$, and COS solutions computed on a 65×65 grid compared with a reference solution on a 257×257 grid.

Corrected splitting. Next, we demonstrate the effect of the correction strategy. Figure 8 shows a plot of the solution at time $t = 0.45$ for $\varepsilon = 0.005$ obtained on a 65×65 grid with OS, OS$^{\mathrm{ds}}$, and COS. To enhance the differences between the three methods, the CFL number is 16.0 in all runs. For reference, we have included a solution computed on a 257×257 grid by OS with CFL number 2.0. We observe that both OS and OS$^{\mathrm{ds}}$ give a too wide representation of the water front due to errors in the viscous splittings. Moreover, OS$^{\mathrm{ds}}$ shows some minor grid orientation effects. By including the correction strategy, the water front is resolved (almost) correctly.

Grid orientation effects. Methods based on dimensional splitting are known to exhibit grid orientation effects. To investigate this effect, we rotated the grid 45° clockwise. The problem setup is then as follows: a square of length $\sqrt{2}$ with injection wells in the lower left and upper right corner and production wells in the other two corners. In the original setup, the main flow direction is along the diagonal, while in the new setup the flow follows the axis of the grid. We set $\varepsilon = 0.005$ and use a 129×129 grid with a CFL number of 2.0 and 52 IMPES steps to reach time $t = 1.3$. Hence the cells on the rotated grid are a factor $\sqrt{2}$ coarser than on the original grid. Table 3 shows the times to water breakthrough and oil production at time $t = 1.3$ for viscosity ratios $\mu_o : \mu_w$ equal 10:1, 1:1, and 1:10 for both the rotated and the original grid. Notice that the differences are minor for the 10:1 runs, a bit larger for the 1:1 runs, and quite large for the 1:10 runs. The latter case corresponds to near piston-like displacement (linear effective flux functions) and is a worst case for our saturation solver due to the lack of self-sharpening in the hyperbolic steps.

TABLE 3. *Water breakthrough, accumulated oil production at final time $t = 1.3$, and largest observed mass balance error (relative to initial oil volume).*

	10:1		1:1		1:10	
	original	rotated	original	rotated	original	rotated
water b.t.	0.321	0.319	0.671	0.666	0.970	0.960
oil prod.	0.582	0.580	0.848	0.845	0.981	0.971
error (%)	-0.037	-0.034	0.049	-0.056	0.23	-0.70

Figure 9 shows the saturation at time $t = 0.66$ for the 1:1 simulation on the rotated geometry and the corresponding saturation on the original grid. Observe that the plots are almost identical. The only difference is in the tip of the water finger. For the original setup, the tip has a slightly rounded profile due to numerical diffusion in the two grid directions. For the rotated grid, the finger occurs at the right boundary of the computational domain and is therefore subject to less diffusion. Hence, the tip is kept sharp. The same observations were made for runs with viscosity ratio 10:1.

The results for the 1:10 runs are shown in Figure 10. Here, there is a pronounced difference in the two plots (as one would expect from the different times to water breakthrough in Table 3). For the rotated geometry, the water is flowing too fast along the upper and the right boundary in the original domain. These boundaries correspond to the diagonal on the computational grid, see Figure 11. Due to numerical diffusion, the tip of the *oil* finger(s) is rounded, which gives faster flow

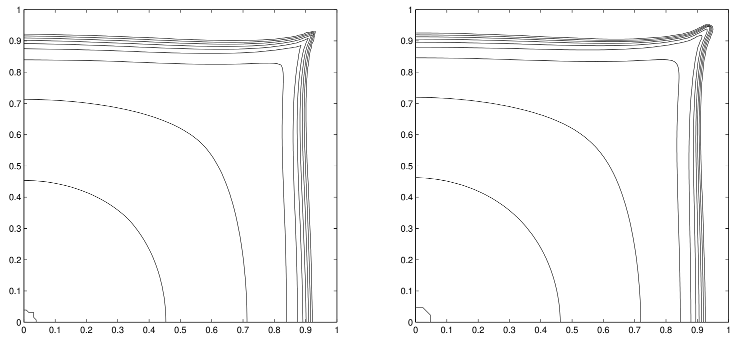


FIGURE 9. Saturation on the rotated geometry (left) and on the original setup (right) at time $t = 0.66$ for viscosity ratio 1:1.

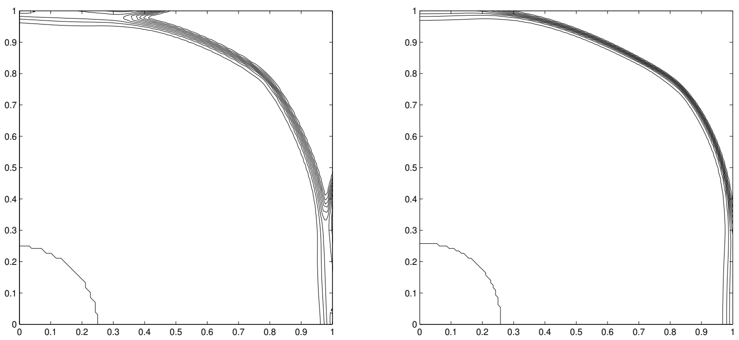


FIGURE 10. Saturation on the rotated geometry (left) and on the original setup (right) at time $t = 0.85$ for viscosity ratio 1:10.

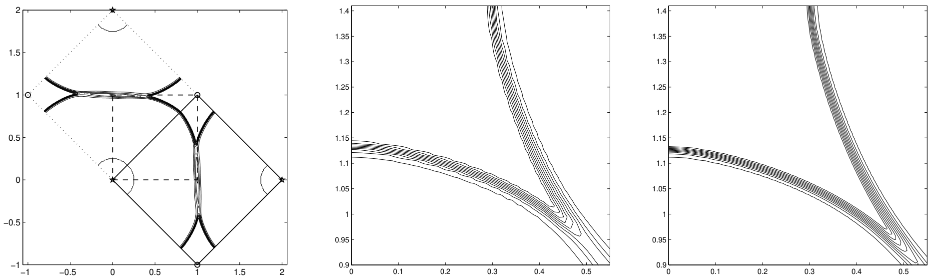


FIGURE 11. (Left) Relation between the rotated grid (solid line) and the original grid (dashed square). The dotted square is a reflection of the computational grid around the diagonal $x = y$. (Middle) The oil finger on a 129×129 grid and on a 257×257 grid (right).

of water towards the production well. *Decreasing* the number of IMPES steps reduces this effect. In fact, with 13 IMPES steps the water breaks through at time 0.946 on the rotated grid and at time 0.949 on the original grid. The oil productions are 0.976 and 0.981, respectively and the errors are -0.28% and 0.11%. In the case of $\mathrm{OS}^{\mathrm{ds}}$, the grid orientation effects are more pronounced, since this method uses dimensional splitting also for the diffusive part of the equation.

Similar grid orientation studies have been carried out for *unstable* displacements [28], where it is demonstrated that the operator splitting method converges to different solutions for the two orientations of the grid.

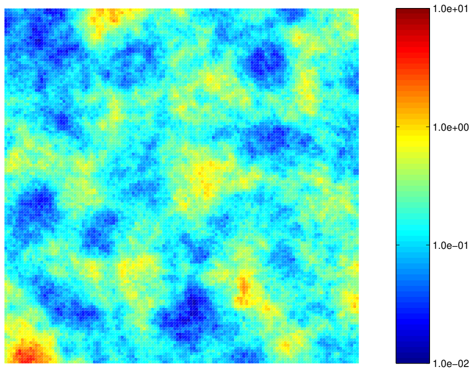


FIGURE 12. Permeability field plotted on a logarithmic color scale. The permeability varies between 13 mD and 4.2 D.

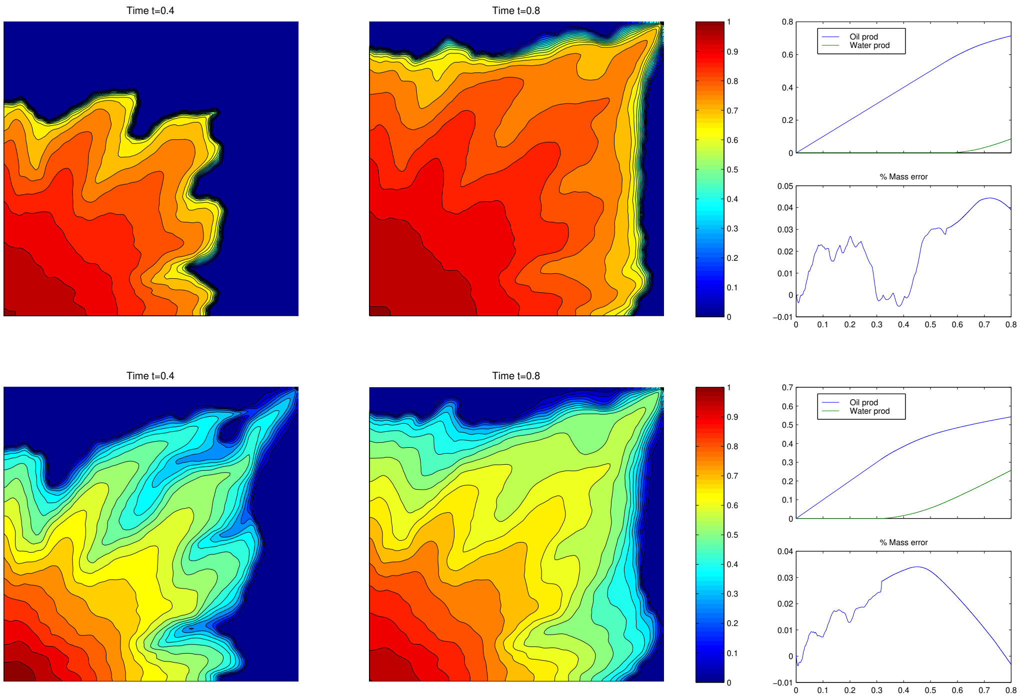


FIGURE 13. Quarter five-spot simulation for viscosity ratios $\mu_o : \mu_w$ equal 1:1 (top) and 5:1 (bottom). The error is measured relative to the initial amount of oil inplace.

3.5. **Heterogeneous quarter five-spots.** In the next example, we include a heterogeneous permeability field in the quarter five-spot simulation (see Figure 12). The permeability field is generated as $K(x) = \exp(Z(x))$, where $Z(x)$ is a Gaussian field. The Gaussian field is specified by its covariance function (giving the smoothness) and the expectation.

Figure 13 shows solutions computed for viscosity ratios $\mu_o : \mu_w$ equal 1:1 and 5:1. The diffusion coefficient is $\varepsilon = 0.005$, the simulation grid consists of 129×129 blocks, and we use 20 IMPES steps to reach final time $t = 0.8$ with a CFL number of 2.0 for the saturation solver up to water breakthrough. Notice that the error is of the same magnitude as observed for the homogeneous quarter five-spot simulations.

Next, we consider a permeability field containing low-permeable blocks which are barriers to the flow (see Figure 14). The blocks follow a Poisson point process, that is, the number of barriers is Poisson-distributed and the positions are uniformly distributed over the domain. The areas

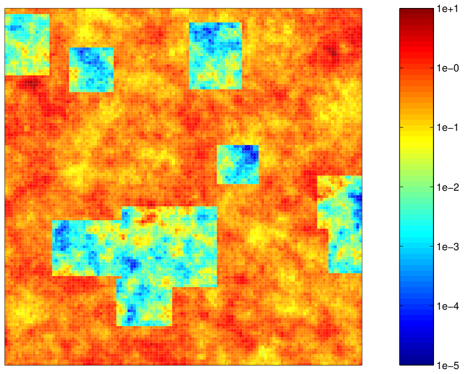


FIGURE 14. Permeability field with low-permeable regions plotted on a logarithmic color scale. The permeability varies between 0.012 mD and 8.6 D.

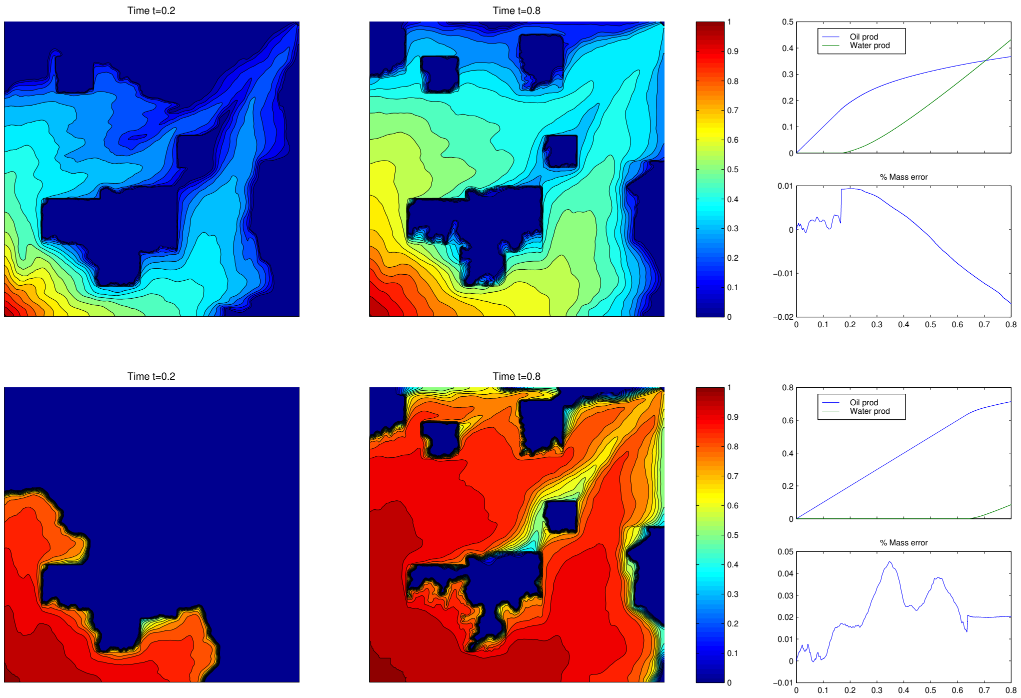


FIGURE 15. Quarter five-spot simulation for viscosity ratios $\mu_o : \mu_w$ equal 20:1 (top) and 1:2 (bottom). The error is measured relative to the initial volume of oil inplace.

of the blocks are independent, while the extents in each direction are correlated and bi-normally distributed. The Gaussian field was generated using an expectation with a (low) constant inside the blocks and another (higher) constant outside.

Figure 15 shows the solutions computed for viscosity ratios $\mu_o : \mu_w$ equal 20:1 and 1:2. The diffusion coefficient is $\varepsilon = 0.005$ and we use 257×257 grid with 80 IMPES steps to reach final time 0.8 and CFL number 2.0 in the saturation solver. As for the above case, the fingering effects are more pronounced at the adverse viscosity ratio. Notice also the improved areal sweep and penetration into low-permeable regions in the lower plot. Figures 16 and 17 show saturation surfaces at four different times up to water breakthrough for each simulation.

It is also interesting to compare the time steps of the two simulations as in Figure 18. Here we clearly see the effect of the switching in CFL numbers at water breakthrough (we set CFL equal 1.0 when the water production exceeds 10^{-4}). For the 20:1 simulation, the CFL number was reduced

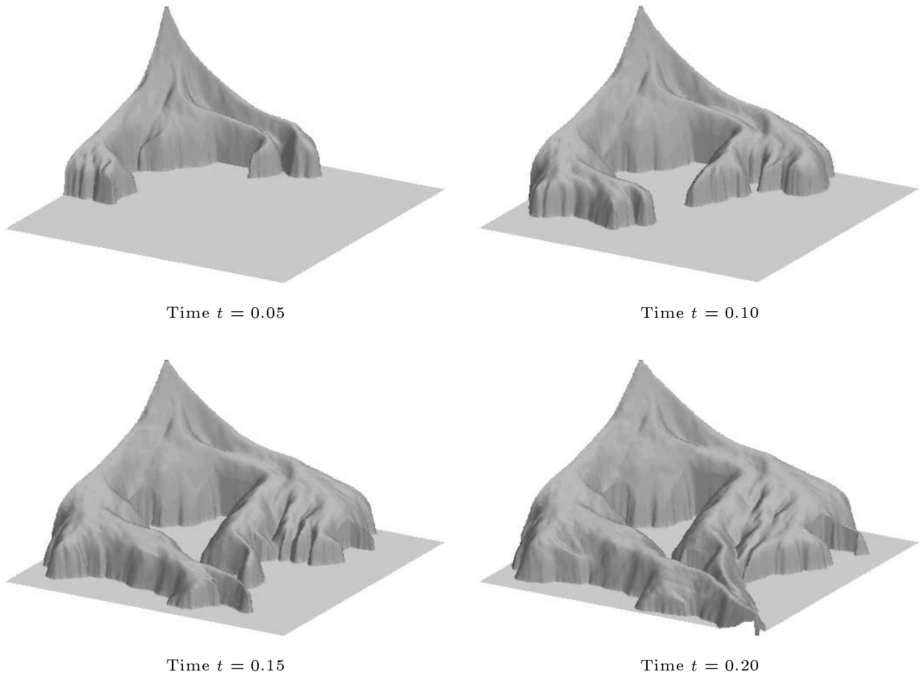


FIGURE 16. Saturation surface plots for viscosity ratio 20:1.

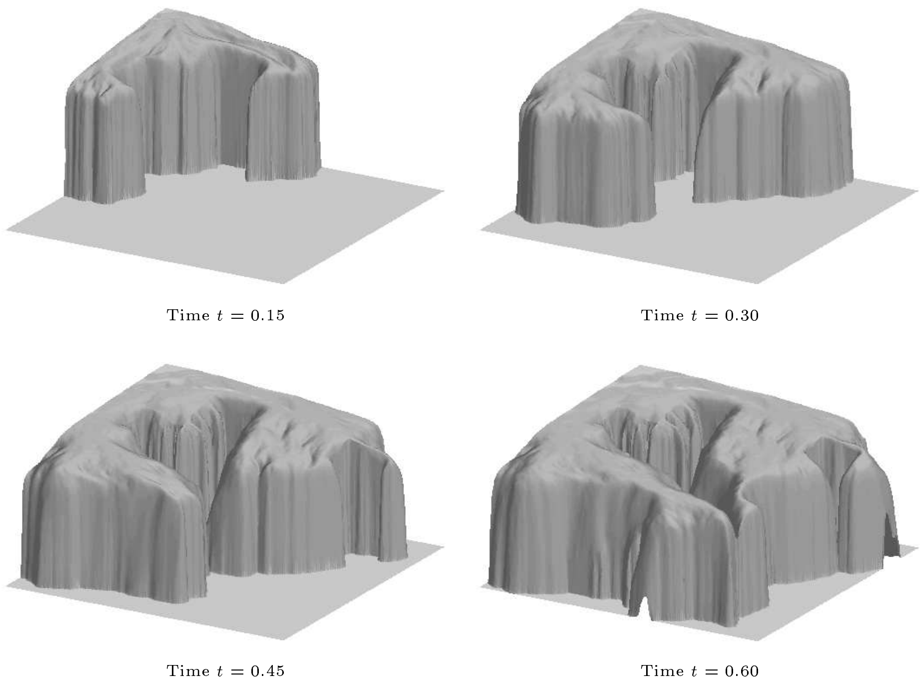


FIGURE 17. Saturation surface plots for viscosity ratio 1:2.

already at time $t = 0.168$. The simulation spent a total of 21 668 time steps in the saturation solver, out of which 555 were with CFL number 2.0. For the 1:2 simulation, the CFL number was reduced at time $t = 0.637$, having spent 929 out of 5 201 time steps.

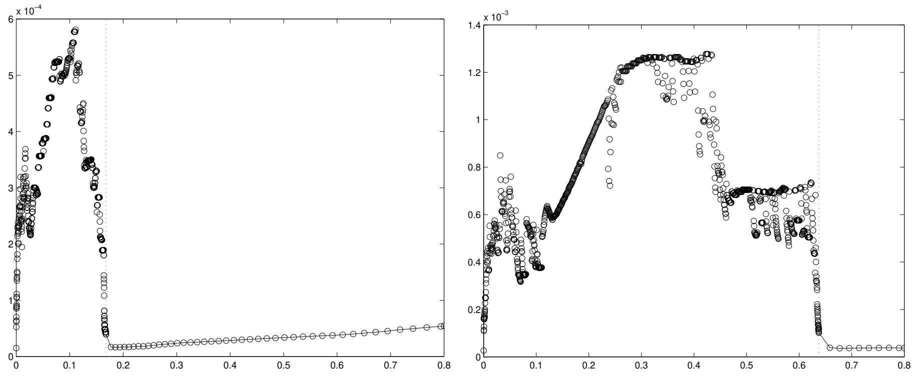


FIGURE 18. Time steps in the saturation solver for the 20:1 simulation (left) and 1:2 simulation (right). After water breakthrough (vertical dotted line), only every one-hundred step is plotted.

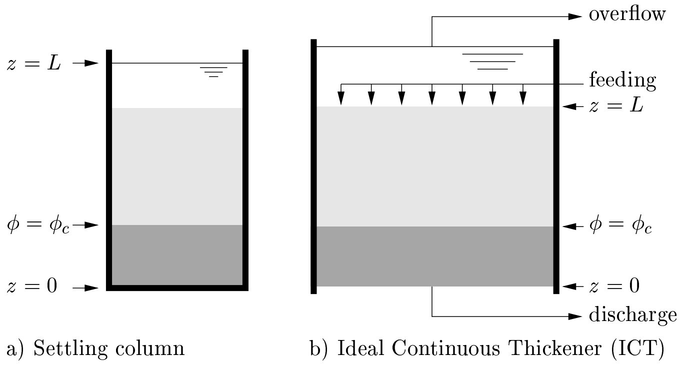


FIGURE 19. Idealized continuous thickener (ICT)

4. SEDIMENTATION

In the previous section, the parabolic equations possessed at most a two-point degeneracy. In this section we consider strongly degenerate equations, i.e., equations with a mixed hyperbolic–parabolic nature. An important application of such equations occur in the phenomenological theory for batch and continuous sedimentation of flocculated suspensions [9]. We present the result of our solution strategy applied to two simple examples of batch sedimentation from [9].

4.1. **Mathematical model.** Sedimentation is the process under which a mixture of fluid and fine solid particles is separated into its solid and liquid components under the influence of gravity. The process is applied in a variety of industrial applications, especially in the mining industry. The settling of flocculated suspensions in a one-dimensional idealized continuous thickener (ICT) leads to a mathematical model of mixed hyperbolic–parabolic type of the form (1) for the volumetric solid concentration ϕ, see [9]. Here, the mixed nature corresponds to the interface between upper part of the system where the solid effective stress σ_e is constant and the portion in space where this quantity varies. This stress is transmitted when the solid flocs get into contact with each other and form a network at a critical concentration value ϕ_c. Above the interface the governing equation is nonlinearly hyperbolic, and below it is nonlinearly parabolic. The location of the interface is unknown beforehand. By modeling the process as a mixed hyperbolic–parabolic problem, one avoids the use of different models in the two regions combined with some interface tracking method.

An ICT is a cylindrical vessel showing no wall effects and in which all field variables are assumed to depend only on height z and time t. The thickener is fed at the top ($z = L$) and emptied at the bottom ($z = 0$). Based on local mass and momentum balances and constitutive laws for solid

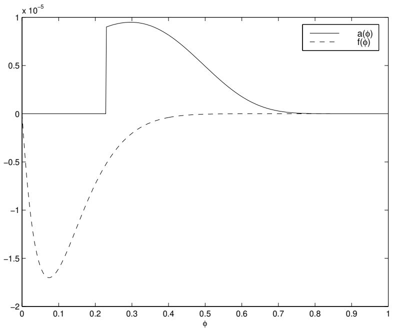


FIGURE 20. Flux $f(\phi,t) = f_{bk}(\phi)$ and diffusion function $a(\phi)$.

and liquid components one derives the model

$$\phi_t + f(\phi,t)_z = (a(\phi)\phi_z)_z, \quad \phi(z,0) = \phi_0(z), \tag{26}$$

where

$$f(\phi,t) = q(t)\phi + f_{\mathrm{bk}}(\phi), \quad f_{\mathrm{bk}}(\phi) = -\Delta\rho g \phi^2(1-\phi)^2/\alpha(\phi), \quad a(\phi) = -\frac{f_{\mathrm{bk}}(\phi)\sigma_{\mathrm{e}}'(\phi)}{L\Delta\rho g\phi}.$$

Here $q \leq 0$ is the volume average velocity of the mixture, α a function relating solid-fluid interaction force per mass to solid-fluid relative velocity, g gravity, and $\Delta\rho$ the density difference of solid and fluid. The Kynch batch flux density $f_{\mathrm{bk}}(\phi)$ is negative in the interval $(0,1)$ and zero outside, and $a(\phi) > 0$ for $\phi_c \leq \phi < 1$ and zero elsewhere. A typical plot of f and a is given in Figure 20.

The equation is further restricted by the boundary conditions

$$\big(f_{\mathrm{bk}}(\phi) - a(\phi)\phi_z\big)\big|_{z=0} = 0, \qquad \phi(L,t) = \phi_L(t). \tag{27}$$

The well-posedness of (26) and (27) within a suitable entropy solution framework is established by Bürger and Wendland [8] and more recently by Bürger, Evje, and Karlsen [7].

4.2. **Solution strategy.** To solve (26) we use the operator splitting (5) introduced previously in this paper, augmented with a special treatment of the boundary conditions (27), see also [6]. The condition at the upper boundary is imposed on both the hyperbolic and the parabolic step, whereas the condition on the lower boundary is only imposed on the parabolic step. In the hyperbolic step, no condition is specified at the lower boundary, corresponding to an outflow condition where waves are allowed to leave the domain without any kind of reflections. Thus the hyperbolic step reads

$$v_t + f(v,t)_z = 0, \quad v(z,0) = v_0(z), \quad v(L,t) = \phi_L(t),$$

and similarly the parabolic step

$$w_t = \big(a(w)w_z\big)_z, \quad w(z,0) = w_0(z), \quad \big(f_{\mathrm{bk}}(w) - a(w)w_z\big)\big|_{z=0} = 0, \quad w(L,t) = \phi_L(t).$$

We have also experimented with other forms of splitting for the lower boundary condition. One could for instance impose $f_{\mathrm{bk}}(v(0,t)) = 0$ on the hyperbolic step and $a(w(0,t))w_z(0,t) = 0$ on the parabolic step, but this approach implies a zero gradient at $z = 0$, which is unphysical.

4.3. **Batch settling of a uniform suspension.** We consider a settling column of height $L = 6\,[\mathrm{m}]$ filled with a flocculated suspension of uniform concentration $\phi_0 = 0.123$. (In [9] the height is $L = 2\,[\mathrm{m}]$). The column is closed at the bottom and without feed. For the Kynch batch flux density, we use a Richardson and Zaki type function

$$f_{\mathrm{bk}}(\phi) = -6.05 \times 10^{-4}\phi(1-\phi)^{12.59}\ [\mathrm{m/s}],$$

and solid effective stress function of the form

$$\sigma_{\mathrm{e}}(\phi) = \begin{cases} 0, & \phi \leq \phi_{\mathrm{c}} = 0.23, \\ 5.35 \times \exp(17.9\phi)\ [\mathrm{N/m^2}], & \phi > \phi_{\mathrm{c}}. \end{cases}$$

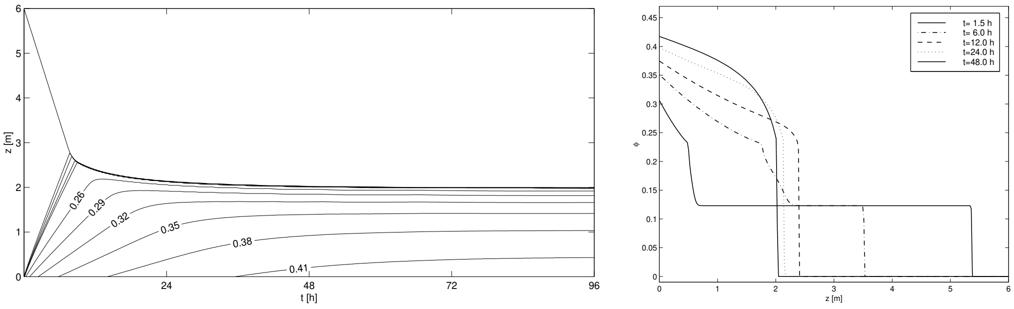


FIGURE 21. (Left) Settling plot for batch sedimentation with compression. (Right) Concentration profiles at various times.

TABLE 4. *Maximum* CFL *number,* CPU *time, maximum mass error in percent, and relative L^1 error in percent at four selected times for different number of splitting steps n. The L^1 errors are computed relative to a fine grid solution computed on a* 3 000 *grid.*

n	CFL	CPU time	m.m.e.	$t = 6$ h	$t = 12$ h	$t = 24$ h	$t = 48$ h
4000	2.5	6.0 sec	-0.12	1.543	0.604	0.365	0.524
2000	5.0	4.3 sec	-0.20	0.842	0.523	0.288	0.456
1000	10.0	3.7 sec	-0.46	1.239	0.686	0.482	0.551
500	20.0	3.1 sec	-1.08	1.668	1.023	1.039	1.049

This gives a discontinuous diffusion coefficient $a(\theta)$, see Figure 20. The other parameters are $\Delta\rho = \rho_{\mathrm{s}} - \rho_{\mathrm{f}} = 1500$ [kg/m^3] and $g = 9.81$ [m/s^2].

Figure 21 shows a settling plot (isocontours of ϕ in the (t, z) plane) for the first 96 hours of the settling process. We have also included five selected concentration profiles. Due to the gravity, the flocs start to move downwards and come into contact at the bottom. Simultaneously a column of pure liquid forms at the top; see the concentration profile at time $t = 1.5$. Notice especially the kink in the sediment profile at $\phi = \phi_c = 0.23$, which is caused by the discontinuity in the diffusion coefficient $a(\phi)$ at the interface between the hyperbolic and the parabolic region. As time increases, the interface between pure liquid and the suspension (with concentration $\phi_0 = 0.123$) moves downwards, while the interface between sediment and suspension moves upwards. Around $t = 9.0$, the initial suspension has disappeared and there is only one interface between sediment and pure liquid. The interface moves downwards causing increasing compression and the process goes towards a steady state.

The plots in Figure 21 are from a simulation with 3 000 splitting steps on a grid with 300 cells. The quality of the solution is very good with a lack of mass conservation of at most 0.12%. Reducing the number of splitting steps gives higher efficiency, but increases the mass error, see Table 4. On the other hand, minimum L^1 error is observed for 2 000 splitting steps at all four times.

4.4. **Repeated batch sedimentation.** In the next example we start with the same setup as above. Then after 96 hours we replace the pure liquid above the sediment with a new suspension. Numerically this corresponds to replacing all concentration values below ϕ_c by the initial value ϕ_0. Figure 22 shows the corresponding settling plot and four concentration profiles. The results are in good correspondence with those obtained in [9].

5. STRONGLY DEGENERATE PROBLEMS IN MULTIDIMENSIONS

In this section we give a few examples of mixed hyperbolic–parabolic problems in multidimensions. In the first example, we compare our splitting methods with a recent second-order central difference scheme by Kurganov and Tadmor [37]. In the second example, we investigate convergence rates for a variable coefficient problem.

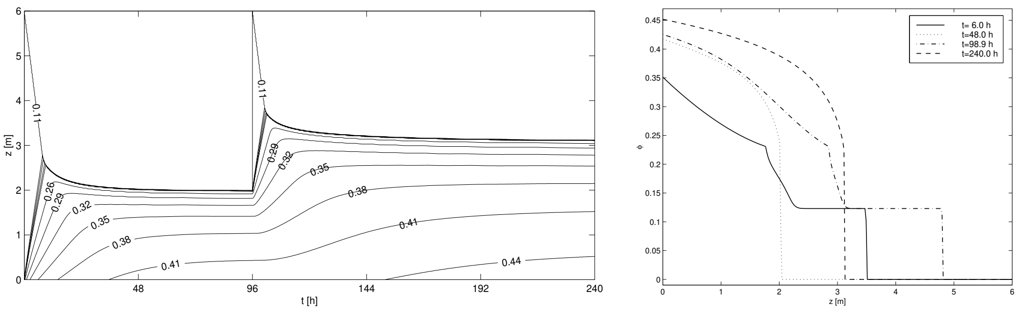


FIGURE 22. (Left) Settling plot for repeated batch sedimentation with compression. (Right) Concentration profiles at various times.

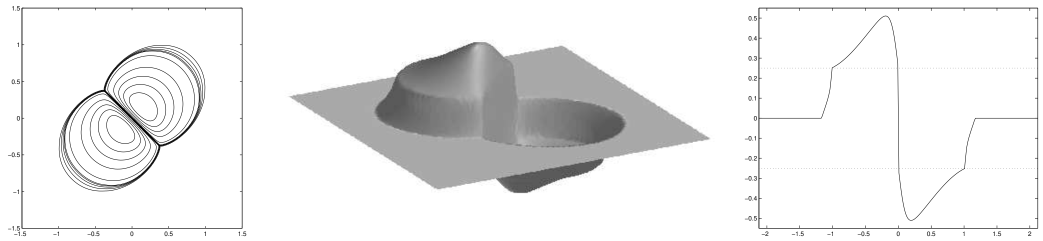


FIGURE 23. Solution at time $t = 0.5$ for Burgers' equation with degenerate diffusion; (left) contour plot, (middle) surface plot, (right) solution along the diagonal.

5.1. **Burgers' equation with degenerate diffusion.** Consider

$$u_t + \nabla u^2 = \varepsilon \nabla \cdot \big(a(u)\nabla u\big), \qquad u(x,0) = u_0(x), \tag{28}$$

where $a(u) = 0$ for $|u| \leq 0.25$ and $a(u) = 1$ elsewhere. Thus the parabolic equation contains a hyperbolic region, $u \in [-0.25, 0.25]$. The initial data equals -1 and 1 inside two circles of radius 0.4 centred at $(0.5, 0.5)$ and $(-0.5, -0.5)$, respectively, and zero elsewhere inside the square $[-1.5, 1.5] \times [-1.5, 1.5]$. This example was first introduced in [27]. Figure 23 shows the solution at time $t = 0.5$ for $\varepsilon = 0.1$. Notice in particular the transition between the hyperbolic and parabolic regions at $u = \pm 0.25$.

To assess the efficiency of our splitting methods, we compare with a semi-discrete, high-resolution, central difference scheme introduced recently by Kurganov and Tadmor [37]. To discretize the ODEs arising from the semi-discrete formulation we have used Runge–Kutta methods given in Table 5.1 in [37]. Table 5 gives L^1 and symmetry errors for a grid refinement study for the operator splitting methods. Table 6 gives L^1 errors for the finite difference scheme. The L^1 errors are computed relative to a fine grid solution ($N = 2^{10}$).

We see that OS gives less error but has a slightly higher runtime than $\mathrm{OS}^{\mathrm{ds}}$. The central difference scheme is less accurate and has much higher runtime on the same grid. Figure 24 gives a plot of numerical error versus runtime for the four methods. The difference in efficiency between the operator splitting methods and the central difference schemes is striking (a factor around 100). This is first of all due to the strict stability restriction of the two Runge–Kutta solvers. Kurganov and Tadmor [37] used an explicit embedded integration method with a large stability region. This is likely to give a significant improvement in the efficiency of the central difference scheme. However, there is also a similar speedup potential in the operator splitting methods. With 128 grid blocks in each direction, around 75% of the runtime is spent in the diffusive steps (which use forward Euler for the time integration). For the 256 grid, the diffusive steps represent 90% of the runtime. By introducing super-time-stepping as in [27], the runtime was reduced by 30 to 60%.

TABLE 5. *Runtimes in seconds and measured L^1 and symmetry errors for operator splitting computations on an $N \times N$ grid with n splitting steps.*

		Method: OS			Method: $\mathrm{OS}^{\mathrm{ds}}$		
N	n	L^1 error	sym. err.	CPU	L^1 error	sym. err.	CPU
32	4	1.161e-01	4.822e-02	0.04	1.272e-01	1.500e-01	0.04
64	8	6.638e-02	1.337e-02	0.24	7.201e-02	8.130e-02	0.23
128	16	3.109e-02	3.754e-03	2.57	3.788e-02	4.171e-02	2.38
256	32	1.499e-02	1.466e-03	44.1	1.940e-02	2.120e-02	39.9

TABLE 6. *Runtime in seconds and measured L^1 errors for central difference computations on an $N \times N$ grid with* CFL *number ν.*

	Runge–Kutta order 2			Runge–Kutta order 3		
N	ν	L^1 error	CPU	ν	L^1 error	CPU
32	0.200	1.806e-01	0.12	0.25	1.835e-01	0.14
64	0.110	1.025e-01	1.58	0.124	1.032e-01	2.15
128	0.055	5.520e-02	31.1	0.070	5.535e-02	37.2
256	0.027	2.822e-02	543.0	0.035	2.824e-02	626.0

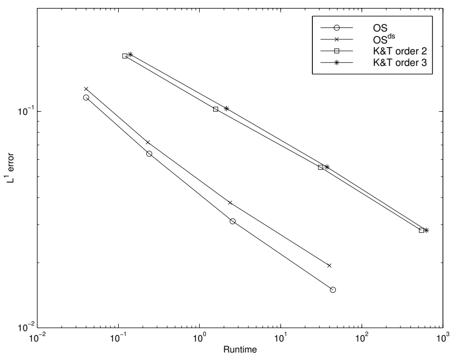


FIGURE 24. Estimated L^1 error versus runtime for the operator splitting methods and the central difference schemes.

5.2. **Burgers' equation with variable coefficients.** Consider

$$u_t + \big(U(y,t), V(x,t)\big) \cdot \nabla(u^2) = \varepsilon \nabla(a(u)\nabla u), \qquad u(x,y,0) = u_0(x,y),$$

where

$$U(y,t) = \cos\big(\pi(y+t)\big), \qquad V(x,t) = \sin\big(\pi(x+t)\big)$$

and u_0 equals one inside a circle with radius 0.4 centered at the origin and zero elsewhere. The diffusion coefficient is $\varepsilon = 0.01$ and $a(u)$ equals zero for $u < 0.5$ and one elsewhere, giving a hyperbolic subregion for $u \in [0, 0.5]$. All computations are carried out using OS.

Figure 25 shows the solution at four distinct times up to $t = 2.0$. Figure 26 shows a comparison with the solution for the purely hyperbolic problem ($a(u) \equiv 0$) and the purely parabolic problem ($a(u) \equiv 1$). The purely hyperbolic problem was studied in [38]. The parabolic solution is obviously a 'diffusive' version of the hyperbolic solution, whereas the solution in the degenerate case has certain features not seen in the other two cases. These features arise from the interplay of the hyperbolic and the parabolic region.

Tables 7 gives the result of a convergence study with respect to the spatial discretization. The convergence rates indicate first order convergence. Similarly, Table 8 reports a convergence study with respect to the number of splitting steps. With few splitting steps, the splitting error in the hyperbolic step gives a large contribution to the total error. As the number of splitting steps

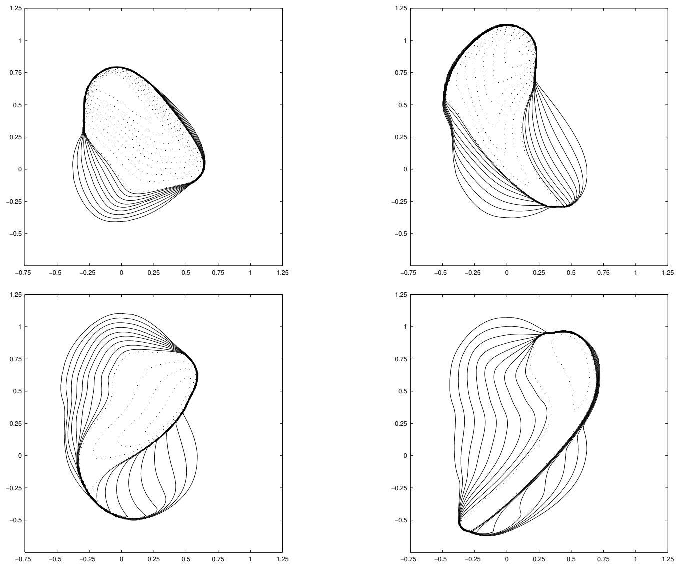


FIGURE 25. Contour plots of the solution at times t=0.5, 1.0, 1.5, and 2.0 (from upper left to lower right) computed on a 200×200 grid with 100 splitting steps. There are 10 equally spaced contour lines in the hyperbolic region $u \in [0, 0.5]$ (solid lines) and in the parabolic region $u \in [0.5, 1]$ (dotted lines).

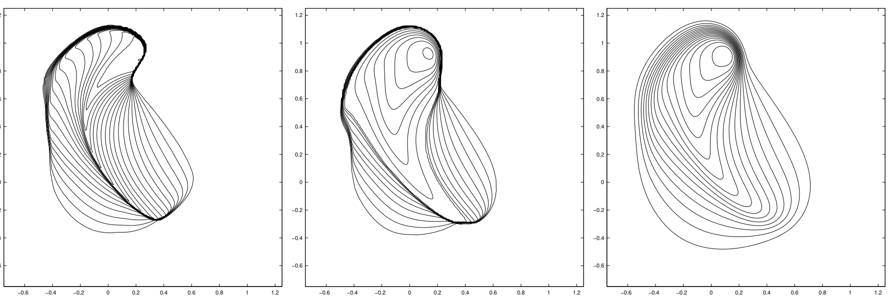


FIGURE 26. Solution at $t = 1.0$ for hyperbolic problem (left), mixed hyperbolic-parabolic (middle), and parabolic (left).

increases, the splitting error decreases and so does the total error. At the same time, the numerical dissipation increases in the hyperbolic steps (due to repeated projections). Around 64 steps (CFL number equal 6.0), the two error contributions balance at a minimum. For larger number of steps, the numerical dissipation becomes dominant and the total error increases. The effect of the projection errors increases with increasing time, as is observed for e.g., $n = 256$.

6. Concluding Remarks

In this paper we have demonstrated the applicability of operator splitting methods to a large class of scalar, convection-dominated problems. Due to explicit tracking of shocks in the hyperbolic part, and the possibility of correcting unphysical entropy loss, the methods give very accurate representation of sharp gradient phenomena. This has been demonstrated for both reservoir simulation and simulation of sedimentation processes. The main ingredient in the splitting methods is the use of a large-step front-tracking method, which is very computationally efficient compared with standard finite difference methods. In the parabolic steps, we have mostly used a simple explicit finite difference method. This step can be made more efficient by using a more sophisticated

TABLE 7. *Convergence rates and errors in L^1-norm relative to a fine grid solution (800 × 800 with 400 steps) for runs with m time steps on an $n \times n$ grid.*

n	m	$t = 0.5$		$t = 1.0$		$t = 1.5$		$t = 2.0$	
50	24	2.400e+01	—	2.730e-02	—	2.021e-02	—	1.590e-02	—
100	48	1.327e-02	1.04	1.002e-02	1.01	8.328e-03	0.93	1.005e-02	0.94
200	96	5.923e-03	1.16	4.618e-03	1.12	3.995e-03	1.06	4.778e-03	1.07
400	192	2.093e-03	1.50	1.679e-03	1.46	1.484e-03	1.43	1.760e-03	1.44

TABLE 8. *Convergence rates and errors in L^1 norm relative to a fine grid solution for runs with n steps on a* 200 × 200 *grid.*

n	$t = 0.5$		$t = 1.0$		$t = 1.5$		$t = 2.0$	
16	3.636e-02	—	2.555e-02	—	1.654e-02	—	1.998e-02	—
32	1.824e-02	1.00	1.271e-02	1.01	8.340e-03	0.99	1.014e-02	0.98
64	8.897e-03	1.04	6.330e-03	1.01	4.627e-03	0.85	5.600e-03	0.86
128	4.747e-03	0.91	4.317e-03	0.55	4.253e-03	0.12	4.943e-03	0.18
256	4.711e-03	0.01	5.909e-03	-0.45	6.285e-03	-0.56	7.115e-03	-0.53

method, based e.g., on semi-discretization and a large-step ODE solver or an implicit discretization combined with an efficient nonlinear solver.

REFERENCES

[1] M. Afif and B. Amaziane. On convergence of finite volume schemes for one-dimensional two-phase flow in porous media. Submitted to *IMA J. Numer. Anal.*

[2] M. Afif and B. Amaziane. Convergence of finite volume schemes for a degenerate convection-diffusion equation arising in two-phase flow in porous media. Preprint, 1999.

[3] M. Afif and B. Amaziane. Analysis of finite volume schemes for two-phase flow in porous media on unstructured grids. In R. Vilsmeier et al., editors, *Finite Volumes for Complex Applications II - Problems and Perspectives (Second International Symposium in Duisburg, Germany, 1999)*, pages 387–394, 1999.

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

[5] F. Bouchut, F. R. Guarguaglini, and R. Natalini. Diffusive BGK approximations for nonlinear multidimensional parabolic equations. Preprint, 1999.

[6] R. Bürger, S. Evje, K. H. Karlsen, and K.-A. Lie. Numerical methods for the simulation of the settling of flocculated suspensions. To appear in a special issue of *Separ. Purif. Technol.*

[7] R. Bürger, S. Evje, and K. H. Karlsen. On strongly degenerate convection-diffusion problems modeling sedimentation-consolidation processes. Preprint, 1999, University of Bergen, Norway. Submitted to *SIAM J. Math. Anal.*

[8] R. Bürger and W. L. Wendland. Existence, uniqueness, and stability of generalized solutions of an initial-boundary value problem for a degenerating quasilinear parabolic equation. *J. Math. Anal. Appl.*, 218(1):207–239, 1998.

[9] M. C. Bustos, F. Concha, R. Bürger, and E. M. Tory. *Sedimentation and Thickening: Phenomenological Foundation and Mathematical Theory.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[10] 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.

[11] M. G. Crandall and A. Majda. The method of fractional steps for conservation laws. *Numer. Math.*, 34:285–314, 1980.

[12] H. K. Dahle, M. S. Espedal, and R. E. Ewing. Characteristic Petrov-Galerkin subdomain methods for convection-diffusion problems. In *Numerical Simulation in Oil Recovery (Minneapolis, Minn., 1986)*, pages 77–87. Springer, New York, 1988.

[13] H. K. Dahle, M. S. Espedal, R. E. Ewing, and O. Sævareid. Characteristic adaptive subdomain methods for reservoir flow problems. *Numer. Methods Partial Differential Equations*, 6(4):279–309, 1990.

[14] H. K. Dahle, M. S. Espedal, and O. Sævareid. Characteristic, local grid refinement techniques for reservoir flow problems. *Internat. J. Numer. Methods Engrg.*, 34:1051–1069, 1992.

[15] H. K. Dahle, R. E. Ewing, and T. F. Russell. Eulerian-Lagrangian localized adjoint methods for a nonlinear advection-diffusion equation. *Comput. Methods Appl. Mech. Engrg.*, 122(3-4):223–250, 1995.

[16] C. N. Dawson. Godunov-mixed methods for advection-diffusion equations in multidimensions. *SIAM J. Numer. Anal.*, 30(5):1315–1332, 1993.

[17] C. N. Dawson. High resolution upwind-mixed finite element methods for advection-diffusion equations with variable time-stepping. *Numer. Methods Partial Differential Equations*, 11(5):525–538, 1995.

[18] C. N. Dawson. Godunov-mixed methods for advective flow problems in one space dimension. *SIAM J. Numer. Anal.*, 28(5):1282–1309, 1991.

[19] C. N. Dawson and M. F. Wheeler. Time-splitting methods for advection-diffusion-reaction equations arising in contaminant transport. In *ICIAM 91 (Washington, DC, 1991)*, pages 71–82. SIAM, Philadelphia, PA, 1992.

[20] M. S. Espedal and R. E. Ewing. Characteristic Petrov-Galerkin subdomain methods for two-phase immiscible flow. *Comput. Methods Appl. Mech. Engrg.*, 64:113–135, 1987.

[21] M. S. Espedal and K. H. Karlsen. Numerical solution of reservoir flow models based on large time step operator splitting algorithms. In A. Fasano and H. van Duijn, editors, *Filtration in Porous Media and Industrial Applications*, Lecture Notes in Mathematics. Springer. To appear.

[22] S. Evje and K. H. Karlsen. Discrete approximations of BV solutions to doubly nonlinear degenerate parabolic equations. To appear in *Numer. Math.*

[23] S. Evje and K. H. Karlsen. Monotone difference approximations of BV solutions to degenerate convection-diffusion equations. To appear in *SIAM J. Numer. Anal.*

[24] S. Evje and K. H. Karlsen. Degenerate convection-diffusion equations and implicit monotone difference schemes. In M. Fey and R. Jeltsch, editors, *Hyperbolic Problems: Theory, Numerics, Applications*, volume 129 in the *Int. Series of Numerical Mathematics*, pages 285–294. Birkhäuser Verlag, 1999.

[25] S. Evje and K. H. Karlsen. Second order difference schemes for degenerate convection-diffusion equations. Preprint, 1999.

[26] S. Evje and K. H. Karlsen. Viscous splitting approximation of mixed hyperbolic-parabolic convection-diffusion equations. *Numer. Math.*, 83(1):107–137, 1999.

[27] S. Evje, K. H. Karlsen, K.-A. Lie, and N. H. Risebro. Front tracking and operator splitting for nonlinear degenerate convection–diffusion equations. In P. Bjørstad and M. Luskin, editors, *Parallel Solution of Partial Differential Equations*, volume 120 in the *IMA Volumes in Mathematics and its Applications*, pages 209-228, Springer-Verlag, 2000.

[28] V. Haugse, K. H. Karlsen, K.-A. Lie, and J. R. Natvig. Numerical solution of the polymer system by front tracking. Preprint, 1999.

[29] H. Holden and L. Holden. On scalar conservation laws in one-dimension. In S. Albeverio, J. E. Fenstad, H. Holden, and T. Lindstrøm, editors, *Ideas and Methods in Mathematics and Physics*, pages 480–509. Cambridge University Press, Cambridge, 1988.

[30] H. Holden and N. H. Risebro. A method of fractional steps for scalar conservation laws without the CFL condition. *Math. Comp.*, 60(201):221–232, 1993.

[31] H. Holden, K. H. Karlsen, and K.-A. Lie. Operator splitting methods for degenerate convection-diffusion equations I: convergence and entropy estimates. Preprint.

[32] K. H. Karlsen, K. Brusdal, H. K. Dahle, S. Evje, and K.-A. Lie. The corrected operator splitting approach applied to a nonlinear advection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 167(3-4):239–260, 1998.

[33] K. H. Karlsen and K.-A. Lie. An unconditionally stable splitting for a class of nonlinear parabolic equations. *IMA J. Num. Anal.*, 19:609-635, 1999.

[34] K. H. Karlsen, K.-A. Lie, N. H. Risebro, and J. Frøyen. A front-tracking approach to a two-phase fluid-flow model with capillary forces. *In Situ* (Special Issue on Reservoir Simulation), 22(1):59–89, 1998.

[35] K. H. Karlsen and N. H. Risebro. Corrected operator splitting for nonlinear parabolic equations. To appear in *SIAM J. Numer. Anal.*

[36] K. H. Karlsen and N. H. Risebro. An operator splitting method for convection-diffusion equations. *Numer. Math.*, 77(3):365–382, 1997.

[37] A. Kurganov and E. Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. UCLA-CAM report no. 99-16.

[38] K.-A. Lie. A dimensional splitting method for nonlinear equations with variable coefficients. *BIT*, 39(4):683–700, 1999.

[39] K.-A. Lie. Front tracking for one-dimensional quasilinear hyperbolic equations with variable coefficients. Manuscript: `http://www.ifi.uio.no/~kalie/`.

[40] K.-A. Lie, V. Haugse, and K. H. Karlsen. Dimensional splitting with front tracking and adaptive grid refinement. *Numer. Methods for Partial Differential Equations*, 14(5):627–648, 1998.

[41] C. M. Marle. *Multiphase Flow in Porous Media.* Institut Francais du Petrole Publications, Editions Technip, 1981.

[42] N. H. Risebro and A. Tveito. Front tracking applied to a nonstrictly hyperbolic system of conservation laws. *SIAM J. Sci. Stat. Comput.*, 12(6):1401–1419, 1991.

[43] A. A. Samarskii, V. A. Galaktionov, S. P. Kurdyumov, and A. P. Mikhailov. *Blow-up in Quasilinear Parabolic Equations.* Walter de Gruyter & Co., Berlin, 1995. Translated from the 1987 Russian original by Michael Grinfeld and revised by the authors.

[44] M. F. Wheeler, W. A. Kinton, and C. N. Dawson. Time-splitting for advection-dominated parabolic problems in one space variable. *Comm. Appl. Numer. Methods*, 4(3):413–423, 1988.

(H. Holden)
Department of Mathematical Sciences
Norwegian University of Science and Technology
N-7491 Trondheim, Norway
E-mail address: holden@math.ntnu.no
URL: http://www.math.ntnu.no/~holden/

(K. Hvistendahl Karlsen)
Department of Mathematics
University of Bergen
Johs. Brunsgt. 12
N-5008 Bergen, Norway
E-mail address: kennethk@mi.uib.no
URL: http://www.mi.uib.no/~kennethk/

(K.–A. Lie)
Department of Informatics
University of Oslo
P.O. Box 1080 Blindern
N–0316 Oslo, Norway
E-mail address: kalie@ifi.uio.no
URL: http://www.ifi.uio.no/~kalie/