A FRONT TRACKING APPROACH TO A TWO-PHASE FLUID FLOW MODEL WITH CAPILLARY FORCES

K. Hvistendahl Karlsen, K.–A. Lie, N. H. Risebro, J. Frøyen

Abstract. We consider a prototype two-phase fluid flow model with capillary forces. The pressure equation is solved using standard finite elements and multigrid techniques. The parabolic saturation equation is addressed via a novel *corrected operator splitting* approach. In typical applications, the importance of advection versus diffusion (capillary forces) may change rapidly during a simulation. The corrected splitting is designed so that any combination of advection and diffusion is resolved accurately. It gives a hyperbolic conservation law for modelling advection and a parabolic equation for modelling diffusion. The conservation law is solved by front tracking, which naturally leads to a dynamically defined residual flux term that can be included in the diffusion equation. The residual term ensures that self-sharpening fronts are given the correct structure. A Petrov–Galerkin finite element method is used to solve the diffusion equation. We present several examples that demonstrate potential shortcomings of standard viscous operator splitting and highlights the corrected splitting strategy. This is the first time a front tracking simulator is applied to a flow model including capillary forces.

0. Introduction

The nonlinear partial differential equations that describe certain two-phase flow situations in porous media can be separated into an elliptic pressure equation and a parabolic saturation equation. The saturation equation is often advection dominated and describes a physical problem where abrupt changes occur in the physical data (steep fronts). In commercial reservoir simulation the finite difference method is the most commonly used discretization technique. Due to the advection dominated nature of the saturation equation, one uses upstream weighting methods to prevent non-physical oscillations from polluting the simulation. However, these methods introduce large amounts of numerical diffusion destroying the structure of important fronts.

Since the flow is advection dominated, some people argue that capillary forces can be neglected. This gives a hyperbolic saturation equation which possesses exact discontinuities instead of the steep (discontinuity like) fronts seen in the parabolic saturation equation. Several groups within the petroleum community have taken such an approach, which enables them to choose from a diversity of sophisticated numerical methods developed over the last two decades for hyperbolic conservation laws. See [27] for a comprehensive introduction to modern numerical techniques for conservation laws. Here we choose to mention the front tracking approach developed by Glimm et al. [17], the higher order Godunov methods employed by Bell et al. [3,4], and the front tracking approach developed by Bratvedt et al. [5,6]. Of particular interest to us is the latter approach which is based on an idea of Dafermos [9] (see also [19]). It has been documented in previous studies [5,6] that this front tracking concept, which differs from the finite difference based method of Glimm et al., produces highly accurate and CPU efficient simulations of two-phase flow models based on a "hyperbolic formulation".

It is well known that the importance of advection versus diffusion (capillary forces) may change rapidly during a simulation in the sense that the flow can be highly advection dominated in certain parts of the reservoir, whereas in other parts the diffusion can be more important. From this point of view, the capillary forces should not be neglected, instead it becomes increasingly important to have a numerical solution strategy that can handle various balances of advection and diffusion within the same application. In particular, *we* want a strategy that maintains the advantages of the front tracking method and uses the algorithms developed when the partial differential equations are "almost hyperbolic".

A standard method for including capillary forces in hyperbolic solvers is viscous operator splitting (OS henceforth). A second order, diffusive term appears in the saturation equation and the OS strategy is to isolate the hyperbolic part in order to apply e.g. front tracking. The remaining diffusive part can be handled by some standard finite element or difference method. This approach, or at least certain variations on this approach, has indeed been taken by several authors, we only mention Beale and Majda [2], Douglas and Russell [13], [30], [16], Espedal and Ewing [14], [15], Dawson [12], and more recently [22].

However, numerical experiments [22,24] suggest that OS can be severely diffusive near steep fronts, at least when the splitting time step is large. OS is therefore not particularly well suited to use with hyperbolic solvers

---

Key words and phrases. Two-phase immiscible flow, operator splitting, front tracking, Petrov–Galerkin finite element method.

The first author has been supported by VISTA, a research cooperation between the Norwegian Academy of Science and Letters and Den norske stats oljeselskap a.s. (Statoil). The second author has been supported by the Research Council of Norway under grant 100555/410.

that allow for large time steps. Let us elaborate on this feature by studying an application of OS to a one-dimensional model equation. Consider therefore Burgers' equation [7], $\partial_t s + \partial_x \left(\frac{1}{2}s^2\right) = \varepsilon \partial_x^2 s$, with Riemann data $s(x,0) = 1$ for $x < 0$ and $s(x,0) = 0$ for $x > 0$. The true solution is a single (shock) front moving with positive velocity. In particular, the size of the shock layer is $\mathcal{O}(\varepsilon)$ (see e.g. [32]) which contrasts the well-known $\mathcal{O}(\sqrt{\varepsilon})$ - layers seen for linear equations.

Formally, let $\mathcal{S}^f(t)$ denote the solution operator associated with the nonlinear conservation law

$$\partial_t v + \partial_x f(v) = 0, \tag{1}$$

and let $\mathcal{H}(t)$ denote the solution operator associated with the heat equation

$$\partial_t w = \varepsilon \partial_x^2 w. \tag{2}$$

Then the OS approximation takes the form

$$s(x, n\Delta t) \approx \left[\mathcal{H}(\Delta t) \circ \mathcal{S}^f(\Delta t)\right]^n s_0(x). \tag{3}$$

Let us calculate the first step in (3) for Burgers' equation. The entropy solution to the convex conservation law (1) is $v(x,\Delta t) = 1$ for $x < \frac{1}{2}\Delta t$ and 0 for $x > \frac{1}{2}\Delta t$. Using $v(x,\Delta t)$ as discontinuous initial data when solving the heat equation (2), we obtain the following explicit formula for the OS approximation

$$s(x,\Delta t) \approx \int_{-\infty}^{\infty} \frac{1}{\sqrt{4\pi\varepsilon\Delta t}} \exp\left[\frac{-(x-y)^2}{4\varepsilon\Delta t}\right] v(y,\Delta t)\, dy. \tag{4}$$

It is not difficult to deduce from this expression that the shock layer has size $\mathcal{O}(\sqrt{\varepsilon\Delta t})$. Consequently, the shock layer is not properly resolved unless a small time step ($\Delta t = \mathcal{O}(\varepsilon)$) is used, a claim that is in fact supported by numerical evidence [22,24].

An interesting observation is the following. Let $f_c(u)$ denote the upper concave envelope of $f(s) = \frac{1}{2}s^2$ in the interval $[0,1]$. Applying OS to the linear equation $\partial_t s + \partial_x f_c(s)_x = \varepsilon\partial_x^2 s$ still yields the solution (4). In fact, applying OS to the equation $\partial_t s + \partial_x g(s) = \varepsilon \partial_x^2 s$ for any flux function $g(u)$ that lies below or equals $f_c(u)$ will give the approximation (4). As opposed to the exact solution, the OS solution does not take into account the particular shape of the flux function, that is, the information needed to determine if the front is self-sharpening and therefore $\mathcal{O}(\varepsilon)$ - sized. However, the part of the flux function that is neglected can be identified as a residual flux term of the form $f_{\text{res}} \equiv f - f_c$. Now the idea is to include this residual term into the equation modelling diffusion (2), i.e., we shall use an approximation formula of the form

$$u(x, n\Delta t) \approx \left[\mathcal{P}^{f_{\text{res}}}(\Delta t) \circ \mathcal{S}^f(\Delta t)\right]^n u_0(x) \tag{5}$$

instead of (3), where $\mathcal{P}^{f_{\text{res}}}(\Delta t)$ is the solution operator associated with the nonlinear diffusion equation $\partial_t w + \partial_x f_{\text{res}}(w) = \varepsilon\partial_x^2 w$. Due to the special form of f_{res} (convex with $f_{\text{res}}(0) = f_{\text{res}}(1) = 0$), this equation will contain the necessary information needed to ensure the correct balance between advection and diffusion.

As we have seen, it is possible to derive *à priori* the explicit expression for f_{res} for a single Riemann problem. This was first observed by Espedal and Ewing [14] (see also [10]) who suggested a splitting method based on the linear conservation law $\partial_t v + \partial_x f_c(v) = 0$ and the nonlinear diffusion equation $\partial_t w + \partial_x f_{\text{res}}(w) = \varepsilon \partial_x^2 w$, instead of (1) and (2). This two-step method, which can be viewed as a slightly different alternative to the splitting (5), has the advantage of giving the correct size of the shock layer and makes it possible to extend the characteristic methods [13,30] to nonlinear problems without severe time step restrictions.

Of course, an *à priori* construction of the residual flux f_{res} is not possible for general problems. However, Karlsen and Risebro [23] recently observed that by using front tracking (as defined by Dafermos) to solve the nonlinear conservation law (1), it is possible to dynamically construct a residual flux function $f_{\text{res}}(x,\cdot)$ for general problems, so that the corrected splitting approach (5) makes sense in general. Furthermore, the existence of $f_{\text{res}}(x,\cdot)$ turned out to be a direct consequence of front tracking being based on solving Riemann problems.

The purpose of the present paper is to make the corrected operator splitting technique (COS henceforth) fit into the context of two-phase flow simulation and the equations arising there. In particular, we apply both OS and COS based strategies to a few water flooding test cases in order to highlight the features of COS. The main conclusion drawn from the present study is that a numerical solution strategy for two-phase flow models based on front tracking and the corrected splitting formula (5) provides a highly accurate and efficient strategy that seems to manage different balances of advection and diffusion, ranging from the strongly advection dominated case (including the purely hyperbolic one) to the diffusion dominated case.

The rest of this paper is organized as follows: In section 1 we describe the prototype mathematical model used for simulation purposes. In section 2 we explain in detail the solution strategy and the corrected operator splitting technique (5). In section 3 we study an application of our solution strategy to the well-known quarter five-spot test case and a water flooding problem involving multiple wells. Finally, in section 4 we make some concluding remarks.

1. The Two-Phase Flow Model

We study immiscible flow of water and oil in a two-dimensional oil reservoir Ω over a time period $(0,T]$. To focus on the main ideas of the algorithm, we consider a simplified prototype two-phase model [8], assume incompressible flow, and neglect gravity. If we choose the global Darcy velocity u, the global fluid pressure p, and the water saturation s as primary variables, the immiscible displacement of oil by water is governed by a (non-dimensional) system of nonlinear partial differential equations.

We have the pressure equation

$$-\nabla\cdot\left(K(\lambda_w+\lambda_o)\nabla p\right)=0, \text{ in } \Omega\times(0,T], \tag{6}$$

where $K=K(x,y)$ denotes the absolute permeability tensor, $\lambda_i=k_{ri}/\mu_i$ are the mobilities, $k_{ri}=k_{ri}(s)$ the relative permeabilities, and μ_i the viscosities of water and oil, $i=w,o$, respectively.

The pressure equation is coupled via the global Darcy velocity $u=-K\left(\lambda_w+\lambda_o\right)\nabla p$ to the saturation equation

$$\phi\partial_t s+\nabla\cdot(uf)-\varepsilon\nabla\cdot(D\nabla s)=0, \text{ in } \Omega\times(0,T]. \tag{7}$$

Here $\phi=\phi(x,y)$ is the porosity of the rock in the reservoir. The fractional flow function $f=f(s)$ and the diffusion tensor $D=D(x,y,s)$ are known functions given by the expressions

$$f=\frac{\lambda_w}{\lambda_w+\lambda_o}, \qquad D=K\frac{\lambda_o\lambda_w}{\lambda_w+\lambda_o}\frac{\partial p_c}{\partial s},$$

where $p_c=p_c(s)$ is the capillary pressure (a known quantity). The small scaling parameter $\varepsilon>0$ is introduced when the equations are converted to non-dimensional form, and gives the relative balance between advective and capillary forces. The system must be constrained by appropriate initial and boundary conditions, here we will use "no flow" boundary conditions,

$$\begin{aligned} & & s|_{t=0}=s_0, & \qquad \text{in } \Omega \\ u\cdot n=0, & \qquad & D(s)\nabla s=0, & \qquad \text{on } \partial\Omega\times(0,T]. \end{aligned} \tag{8}$$

We model injection and production wells as area wells; that is, we let each well be defined over a small area Ω_i with an outer boundary $\partial\Omega_i$. For the pressure equation we demand that $u\cdot n_i=q_i$ on $\partial\Omega_i$, where q_i represents the injection rate and n_i is the outer normal of $\partial\Omega_i$. For the saturation equation, the production wells are modelled as outflow boundaries and the injection wells as inflow boundaries with saturation kept fixed at 1. To confine our presentation, the boundary conditions will not be explicitly mentioned later when we describe the solution techniques for the pressure and saturation equation.

For computational purposes, let us employ mobilities of the form

$$\lambda_w=s^{p_w}, \qquad \lambda_o=(1-s)^{p_o}, \qquad p_w,p_o>1.$$

Thus, the fractional flow (or flux) function has the usual s-shape, i.e.,

$$f(s)=\frac{s^{p_w}}{s^{p_w}+(1-s)^{p_o}}. \tag{9}$$

Furthermore, we replace the absolute permeability tensor by the identity matrix, and take the diffusion tensor to be diagonal,

$$D(s)=\operatorname{diag}\left(\nu(s),\nu(s)\right), \qquad \nu(s)=4s(1-s). \tag{10}$$

Hence, the components of the diffusion tensor have the typical bell-shape, with degenerate behaviour at $s=0$ and $s=1$. We assume the rock porosity to be constant. Finally, for convenience we take the reservoir domain Ω to be rectangular.

2. The Numerical Strategy

The governing equations (6) and (7) are coupled. A sequential time stepping procedure is used to decouple the equations, which essentially consists of solving one equation at the time, starting with the pressure equation to generate a velocity field. Subsequently, this velocity field is used as input in the saturation equation. This strategy reflects the different nature of the elliptic pressure equation (6) and the advection dominated parabolic saturation equation (7). For an analysis of this procedure, we refer the reader to [26,31]. The velocity is assumed to be smoother than the saturation, which may develop large gradients. Let the pair (p^n, s^n) denote the approximate solution to (6) and (7) at time $t = n\Delta t_s$, where $\Delta t_s > 0$ denotes the sequential time step. The approximation at the next time level is computed in the following two steps:

(1) *Pressure:* Let $\Lambda^n(x,y) = \lambda_w(s^n) + \lambda_o(s^n)$ be the total mobility at time $t = n\Delta t$, and let $\overline{p}$ be the solution to the elliptic equation

$$-\nabla \cdot (\Lambda^n(x,y)\nabla\overline{p}) = 0, \qquad \text{in } \Omega \times (0, \Delta t_s].$$

This equation is first discretized using the Galerkin finite element method with the usual "hat" basis functions. This gives us a linear system of equations. As solver for this system we have used the conjugate gradient method, together with a V-cycle multigrid method as preconditioner. This procedure gives us a piecewise linear approximation to the pressure. Once the pressure is found we can compute the velocity. A straightforward differentiation of the piecewise linear pressure approximation would yield a piecewise constant velocity approximation and convergence of linear order. Instead we have computed the velocity field based on flux consideration. This method, which is conservative, gives a piecewise linear velocity field, and the convergence is superlinear. A detailed description of the pressure and velocity solver is given in [11].

(2) *Saturation:* Update the global Darcy velocity; $\overline{u} = -\Lambda^n\nabla\overline{p}$. Let $\overline{s}$ be the solution to the parabolic equation (with $\overline{s}|_{t=0} = s^n$)

$$\partial_t\overline{s} + \nabla\cdot(\overline{u}f(\overline{s})) - \varepsilon\nabla\cdot(\nu(\overline{s})\nabla\overline{s}) = 0, \qquad \text{in } \Omega\times(0,\Delta t_s],$$

where we have rescaled time to get rid of the constant ϕ.

Finally, the approximation at the next time level is defined by $(p^{n+1}, s^{n+1}) = (\overline{p}, \overline{s})$.

Under the assumption of incompressible flow, the saturation equation can be written in the following non-conservative form

$$\begin{aligned} &\partial_t s + u(x,y)\cdot\nabla f(s) = \varepsilon\nabla\cdot(\nu(s)\nabla s), && \text{in } \Omega\times(0,T], \\ &s(x,y,0) = s_0(x,y), && \text{in } \Omega, \end{aligned} \tag{11}$$

The parabolic equation (11) is "almost hyperbolic", and will be solved by operator splitting; the basic philosophy of this splitting was explained in the introduction. Let us now describe it in detail. Consider therefore the one-dimensional advection–diffusion problem

$$\begin{aligned} &\partial_t s + u(x)\partial_x f(s) = \varepsilon\partial_x(\nu(s)\partial_x s), && (x,t)\in(a,b)\times(0,T], \\ &s(x,0) = s_0(x), && x\in(a,b). \end{aligned} \tag{12}$$

where the velocity $u(x)$ is a piecewise smooth scalar function defined on (a,b). The non-conservative form of the saturation equation is chosen because this is the most natural form if we wish to apply front tracking for the advective part. This method is described in the appendix toghether with the Petrov–Galerkine finite element method used for the diffusion equation.

Divide the time interval $(0,T]$ into time strips $(n\Delta t, (n+1)\Delta t]$, where $n = 0,\dots,N-1$ and $N\Delta t = T$, and let $\{x_i; h\}$ be a uniform subdivision of the interval $[a,b]$. Furthermore, let u_h be the linear interpolant of u at $\{x_i\}$ and f_δ the linear interpolant of f at $\{i\delta\}$. Here, h and δ denote the spatial and polygonal discretization parameters, respectively. Let s^n denote the approximate solution to (12) at time $t = n\Delta t$, for some fixed $n > 0$. We now describe how to inductively construct the solution s^{n+1} from s^n.

Advection update: Solve the following hyperbolic conservation law by front tracking (see the appendix),

$$\partial_t v + u_h(x)\partial_x f_\delta(v) = 0, \qquad v(x,0) = s^n(x), \qquad (x,t)\in(a,b)\times(0,\Delta t]. \tag{13}$$

Let $\mathcal{S}_{u_h}^{f_\delta,x}(t)$ denote the approximate (front tracking) solution operator associated with (13), i.e., $v(x,t) \approx \mathcal{S}_{u_h}^{f_\delta,x}(t)s^n(x)$. Using this notation, we may define an intermediate solution by

$$s^{n+1/2} = \mathcal{S}_{u_h}^{f_\delta,x}(\Delta t)s^n.$$

The function $s^{n+1/2}(x)$ is piecewise constant on a finite number of intervals. Let the discontinuities of $s^{n+1/2}$ be located at the points $\{y_i\}$, and let $\{\overline{x}_i\}$ be a sequence of spatial positions chosen so that $y_i \in (\overline{x}_i, \overline{x}_{i+1})$. In the interval $[y_i, y_{i+1})$ we denote the value of $s^{n+1/2}$ by $\tilde{s}_{i+1}$. Then we define the residual flux term by

$$f_{\text{res}}(x,s) = \begin{cases} f_\delta(s) - f_{\delta,c}(s;\tilde{s}_i,\tilde{s}_{i+1}), & \text{for } x\in[\overline{x}_i,\overline{x}_{i+1}) \text{ and } s\in[\tilde{s}_i,\tilde{s}_{i+1}], \\ 0, & \text{for } x\in[\overline{x}_i,\overline{x}_{i+1}) \text{ and } s\notin[\tilde{s}_i,\tilde{s}_{i+1}]. \end{cases} \tag{14}$$

Here $f_{\delta,c}$ denotes the correct envelope (either convex or concave) of f_δ restricted to $[\widetilde{s}_i, \widetilde{s}_{i+1}]$. Notice that the envelope function $f_{\delta,c}$ depends on the sign of the velocity function $u_h(x)$ and the sign of $(\widetilde{s}_i - \widetilde{s}_{i+1})$, see the appendix. Several ways of choosing $\{\overline{x}_i\}$ exist, see [23,24,25]. Here we choose $\overline{x}_i$ to be (approximately) equal to the midpoint of the intervals where $s^{n+1/2}$ changes monotonicity; hence, each interval $(\overline{x}_i, \overline{x}_{i+1})$ coincides with the monotonicity interval of $s^{n+1/2}$ on which y_i is found, see Figures 3.6 and 3.11. By a monotonicity interval we mean an interval on which $s^{n+1/2}$ is either non-increasing or non-decreasing. The reason that this works well is that only one significant discontinuity is located in each monotonicity interval.

Diffusion update: Let $w(x,t)$ be the Petrov–Galerkin finite element solution (see the appendix) to the parabolic diffusion problem

$$\partial_t w + u_h(x)\partial_x f_{\mathrm{res}}(x,w) = \varepsilon \partial_x\left(\nu(w)\partial_x w\right), \qquad w(x,0) = s^{n+1/2}(x), \qquad (x,t) \in (a,b)\times(0,\Delta t]. \tag{15}$$

The Petrov–Galerkin approximation is a piecewise linear function on a grid chosen such that its nodes coincide with the discontinuity points of the front tracking solution. To ensure convergence of our method, we have to add nodes whenever the spacing between two discontinuities becomes larger than h. Note that this results in a grid well suited for resolving shock layers. We define the COS solution at time $t = (n+1)\Delta t$ in terms of the formula

$$s^{n+1} = \mathcal{P}_{u_h}^{f_{\mathrm{res}},x}(\Delta t)s^{n+1/2},$$

where $\mathcal{P}_{u_h}^{f_{\mathrm{res}},x}(t)$ is the approximate solution operator associated with equation (15) at time t.

To sum up, the COS solution at time $t = T$ can be written abstractly using the product formula

$$s^N = \left[\mathcal{P}_{u_h}^{f_{\mathrm{res}},x}(\Delta t)\circ \mathcal{S}_{u_h}^{f_\delta,x}(\Delta t)\right]^N s_0. \tag{16}$$

Remark. Observe that in the presence of shock fronts, the advection part of (15), f_{res}, only has a sharpening or anti-diffusive effect whose purpose is to balance the diffusion to the correct order. Furthermore, the f_{res} -term is relatively small (but important) in most applications; see the numerical section and [24]. We shall later see that the residual flux term substantially decreases the temporal splitting error.

Note that according to (14), a "local" flux term $f_{\mathrm{res}}(\overline{x}_i+, s)$ is constructed with respect to each shock in the front tracking solution. In particular, since front tracking replaces smooth parts of the flow by series of small shocks, a large number of these local flux terms will zero and will therefore not influence the flow. In addition, to increase the efficiency of the COS algorithm we neglect a local flux term when $|s_R - s_L| \le c_{\mathrm{tr}}$, where s_L and s_R denote the values that the local flux term is based on, and c_{tr} is a (problem dependent) threshold parameter. This way the Petrov–Galerkin solver will only pay attention to anti-diffusive flux terms based on strong shocks in the advection solution, which correspond to significant shock layers in the solution of the advection-diffusion equation. We should also mention that the anti-diffusive flux terms are given (implicitly) as a part of the front tracking method and can be collected by only a few extra computations.

We now turn our attention to the two-dimensional problem (12). The COS scheme presented above is genuinely one-dimensional. There are two obvious ways of generalizing it to several space dimensions; the method of streamlines or the method of dimensional splitting [20,28,24]. Here we will rely on the latter approach. The streamline approach will be considered elsewhere.

Let us explain the dimensional splitting in some detail. Assume that the domain Ω is rectangular and consider a uniform Cartesian grid $\{z_{i,j}; h\}$, where each square grid cell is of the form $z_{i,j} = [x_i, x_{i+1})\times[y_j, y_{j+1})$. Let π denote the two-dimensional projection operator defined on $\{z_{i,j}; h\}$, i.e.,

$$\pi s(x,y) = \frac{1}{h^2}\int_{z_{i,j}} s(\xi,\eta)\, d\xi\, d\eta, \qquad \forall (x,y)\in z_{i,j}.$$

Let f_δ and g_δ be Lipschitz continuous, piecewise linear approximations to f and g, respectively. Furthermore, let $u_h(x;y)$ denote an approximation to the first component of the velocity vector $u(x,y)$. The approximation is continuous, piecewise linear in x and piecewise constant in y . Similarly, let $u_h(y;x)$ denote the approximation to the second component. Let s^n denote the piecewise constant splitting solution at some positive time $t = n\Delta t$, $s^0 = \pi s_0$. We now proceed by explaining how to construct s^{n+1} from s^n.

x - sweep: Let $v(x,\Delta t;y)$ be the front tracking solution at time $t=\Delta t$ to

$$\partial_t v + u_h(x;y)\partial_x f_\delta(v) = 0, \qquad v(x,0;y) = s^n(x;y). \tag{17}$$

Note that y only acts as a parameter in (17). Next, construct the residual flux function $f_{\mathrm{res}}(x,v;y)$ with respect to the constant values taken by $v(x,\Delta t;y)$. Let $w(x,\Delta t;y)$ be the solution at time $t=\Delta t$ to the parabolic equation

$$\partial_t w + u_h(x;y)\partial_x f_{\mathrm{res}}(x,w;y) = \varepsilon\partial_x\left(\nu(w)\partial_x w\right). \qquad w(x,0;y) = v(x,\Delta t;y), \tag{18}$$

computed using the one-dimensional Petrov–Galerkin method on a grid with nodes determined by the discontinuities in $v(x, \Delta t; y)$. To avoid oscillations in the solution, the grid spacing of this one-dimensional grid may have to be choosen smaller than h if ε is small. The solution is piecewise linear on a non-uniform grid, and has to be projected to a piecewise constant solution on the regular grid $z_{i,j}$ to provide initial data for the y - sweep.

y - sweep: Let $v(y, \Delta t; x)$ be the front tracking solution at time $t = \Delta t$ to

$$\partial_t v + u_h(y; x)\partial_y g_\delta(v) = 0, \qquad v(y, 0; x) = (\pi w(\cdot, \Delta t; \cdot))(y; x). \tag{19}$$

Note that x only acts as a parameter in (19). Next, construct the the residual flux function $g_{\text{res}}(y, v; x)$ with respect to the constant values taken by $v(y, \Delta t; x)$. Let $w(y, \Delta t; x)$ be the Petrov–Galerkin solution at time $t = \Delta t$ to the parabolic equation

$$\partial_t w + u_h(y; x)\partial_y g_{\text{res}}(y, w; x) = \varepsilon \partial_y \left(\nu(w)\partial_y w\right), \qquad w(y, 0; x) = v(y, \Delta t; x). \tag{20}$$

The solution at time $t = (n+1)\Delta t$ is defined as $s^{n+1} = \pi w(\cdot, \Delta t; \cdot)$.

In terms of approximate solution operators, the corrected operator splitting solution at time $t = T$ can formally be given by the composition

$$s^N = \left[\mathcal{P}_{u_h}^{g_{\text{res}}, y}(\Delta t) \circ \mathcal{S}_{u_h}^{g_\delta, y}(\Delta t) \circ \mathcal{P}_{u_h}^{f_{\text{res}}, x}(\Delta t) \circ \mathcal{S}_{u_h}^{f_\delta, x}(\Delta t)\right]^N s^0. \tag{21}$$

Remark. Note that s^N is piecewise constant with respect to $\{z_{i,j}\}$. However, in applications we replace s^N by a proper piecewise linear function to obtain second order accuracy in space. Furthermore, in applications we also employ a Strang splitting for the x and y - sweeps, that is, for the dimensional splitting, but not the viscous splitting.

3. Numerical Experiments

In this section we apply the ideas from the previous section to model problems. The first problem is the well-known quarter five-spot case, and the second is a water flooding problem with multiple wells. This type of test cases is well established in the petroleum community. To illustrate the advantages of the corrected operator splitting approach (COS), we will compare it with the standard operator splitting approach (OS); here OS is realized as COS without correction terms and therefore suffers from the same possible deficiencies, i.e., splitting and projection errors. We stress that the numerical results presented here for our OS method are not necessarily representative for all OS methods.

The pressure and saturation solver make use of a compatible grid system; the node points in the pressure solver coincide with the corners of the grid cells in the saturation solver. The pressure (velocity) is updated only once in the examples presented below. We have used a fractional flow function given by (9) with $p_w = p_o = 2$. Since the main purpose is to demonstrate the effect of the residual flux terms, it is sufficient to take the diffusion coefficient $\nu(s)$ to be constant; $\nu(s) = 1$. However, to convince the reader that there is no loss of generality in doing so, we present one computation with the nonlinear diffusion coefficient (10) (see Figure 3.5). We let the reservoir domain Ω be given in unit coordinates and time T in PVI (Pore Volumes Injected). The reservoir is divided into 128×128 grid cells. Reference solutions are computed by OS on a 256×256 grid using 100 time steps. We are primarily interested in discussing the temporal splitting error and not the spatial discretization error, which is small compared with the splitting error when we use a fine grid such as 128×128. In the Petrov–Galerkin solver we use at most 5 Picard iterations. This has proven to be more than sufficient to resolve the nonlinear residual flux terms and is in agreement with the numerical observations reported in [18] on the convergence properties of the iteration.

Due to the well model, the near-well velocities may become very large and force the time step to be unreasonably small. On the other hand, the velocities are of moderate size away from the wells, and there a large time step is feasible. Therefore we have included two or three stages in the computations. During the first stage, when water fronts are close to injection wells, we use a small time step. In the intermediate stage, the water fronts have moved away from the injection wells and we can apply a large time step. Finally, the time step must be reduced again as the water fronts approach the production wells. Note that it is in the "large time step areas", when the problem is advection dominated, that COS compares most favourably with OS. In the other areas most of the behaviour can be attributed to the simplified well model.

Example 1 (Quarter five-spot). The first example is a quarter five-spot test case with a unit injection well placed at $(0, 0)$ and a production well with unit production rate placed at $(1, 1)$. The reservoir is $\Omega = [0, 1]\times[0, 1]$, and the solutions are advanced forward in time (until $T = 0.6$) as follows: $[0, 0.01]$ in one step, $[0.01, 0.5]$ in four steps, and finally $[0.5, 0.6]$ in ten steps. The diffusion coefficient is taken to be constant ($\nu(s) = 1$), and the scaling parameter ε is 0.01. In order to demonstrate that a nonlinear diffusion coefficient causes no trouble, a calculation using (10) is shown in Figure 3.5 (right).

The results of the computations are depicted in Figures 3.1–3.7. Figure 3.1 shows contour plots of the reference solution at times $t = 0.1$, 0.47, and 0.6. Due to large velocities near the injection well, the problem is highly advection dominated at time $t = 0.1$, whereas at time $t = 0.6$ we see that advection and diffusion are equally important (except in a small neighbourhood of the production well). At time $t = 0.47$ the problem is again advection dominated, but not as much as when $t = 0.1$. The importance of advection versus diffusion, often measured by the Péclet number $Pe = \|u\|_\infty \|f'\|_\infty / \varepsilon$, may change quite rapidly during a simulation. Consequently, a proper numerical scheme should be able to resolve the different balances of advection and diffusion in an accurate and consistent way. We feel that COS satisfies such demands, but that neither OS nor any hyperbolic solver ("zero diffusion" models) have this desirable feature, as will be demonstrated in the sequel.

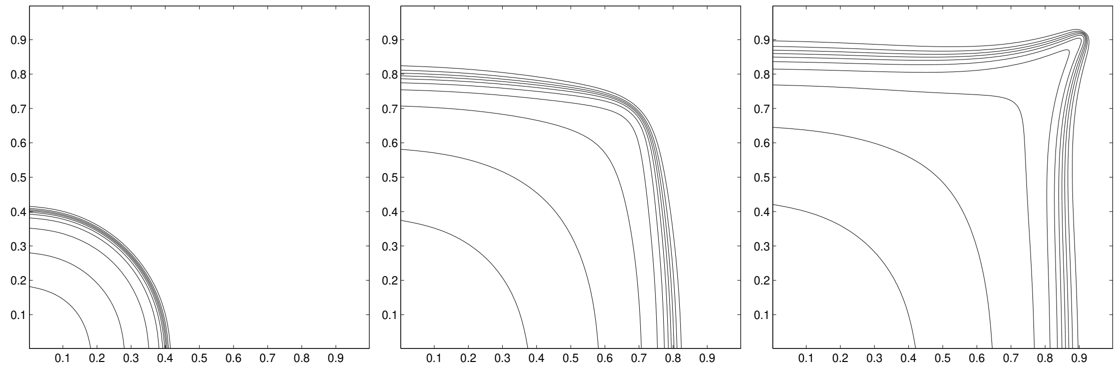


Figure 3.1. Example 1. Reference solution (OS) computed on a 256*256 grid using 100 time steps. Snapshots at times 0.1, 0.47, and 0.6 from left to right.

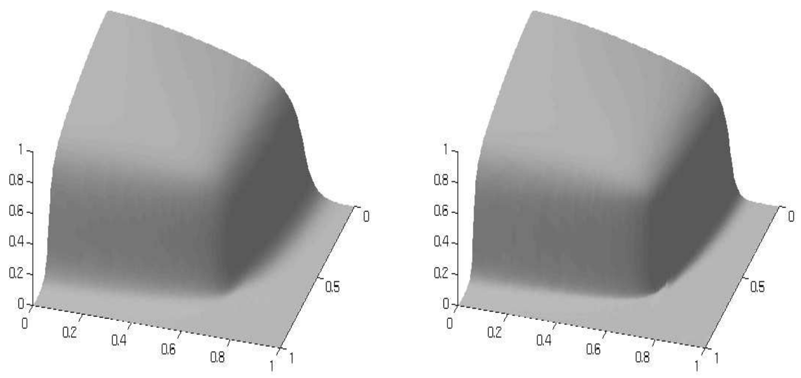


Figure 3.2. Example 1. Three-dimensional plots. (Left) OS solution at time 0.47. (Right) COS solution at time 0.47. Note that OS is more diffusive than COS.

Figure 3.3 shows contour plots of the OS calculations. We see that OS is far too diffusive at times $t = 0.1$ and 0.47, whereas at final time $t = 0.6$ the structure of the water front is resolved rather accurately. But there is a cumulative error which manifests itself in late breakthrough of water (see also Figure 3.7). This effect is perhaps easier to observe in Figure 3.5 (middle), where the saturation fronts along the diagonal of the grid are shown at times 0.47 and 0.6. Figure 3.4 shows the corresponding COS solutions. It is evident that COS produces both correct placement and structure of the water front for all three times. In particular, the improvement over OS at times $t = 0.1$ and 0.47 (advection dominated areas) is remarkable. Note that at time 0.47 neither the COS nor the OS solution is perfectly symmetric with respect to the diagonal. This is due to the dimensional splitting; by increasing the number of time steps slightly, this effect can be eliminated.

The total mass balance error of OS and COS is given in Figure 3.7 (left). We define the mass balance error as 100*(PVI-observed mass of water)/PVI. This error is much smaller for COS than for OS.

How much does the numerical diffusion influence the results? Figure 3.7 (right) presents the saturation profile at time $t = 0.47$ along the line $y \approx 0.15$ for a series of values of ε. We see that the structure of the front is well represented for the two larger values (ε=0.01,0.005), but not for the two smaller values (ε=0.001,0.0005), for which the fronts approximately equal that of a zero diffusion run (not shown in the figure). Notice that it is very difficult to determine the width of shock fronts in advance, since this depends upon the interplay of advection and diffusion and may vary throughout the reservoir. To check if the grid is sufficiently fine to represent fronts, one can do a comparison with runs from the zero diffusion model. Recall that a nonuniform (possibly refined) grid is used in the diffusion step. Therefore, numerical diffusion is mainly caused by the projection operator, which also appears in the zero diffusion model. Hence, setting $\varepsilon = 0$ gives a good indication of the amount of numerical diffusion in the corresponding diffusive runs.

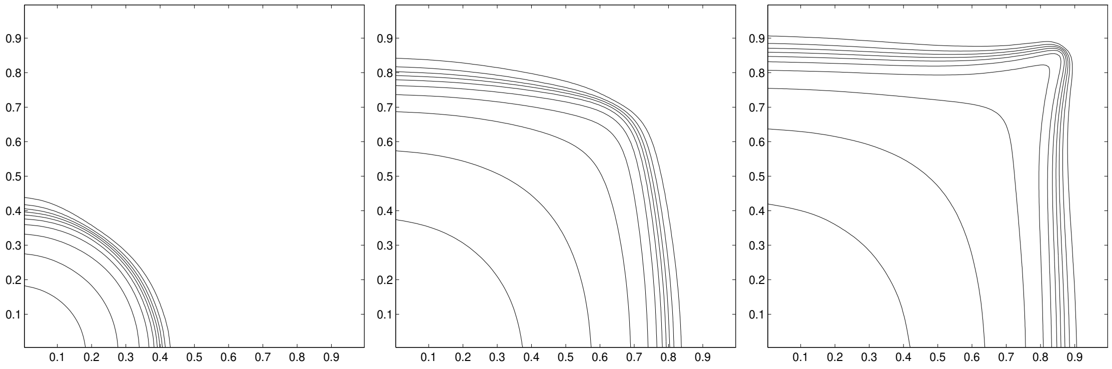


Figure 3.3. Example 1. OS solution; snapshots at times 0.1, 0.47, and 0.6 from left to right.

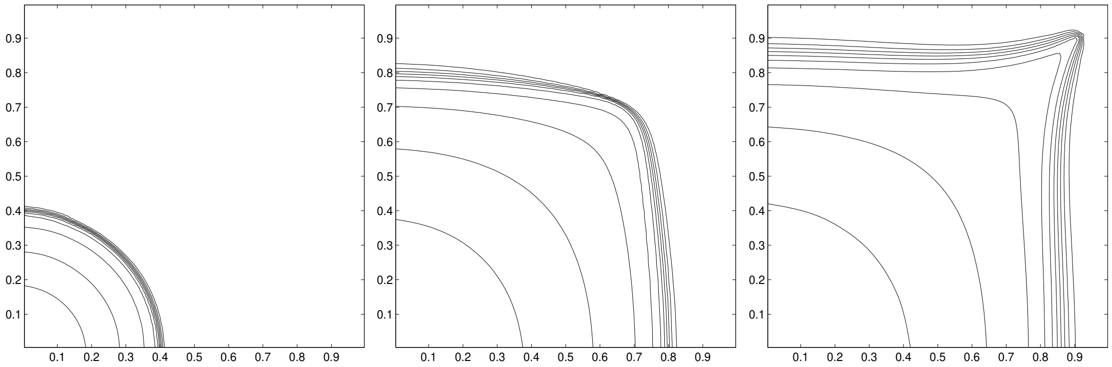


Figure 3.4. Example 1. COS solution; snapshots at times 0.1, 0.47, and 0.6 from left to right.

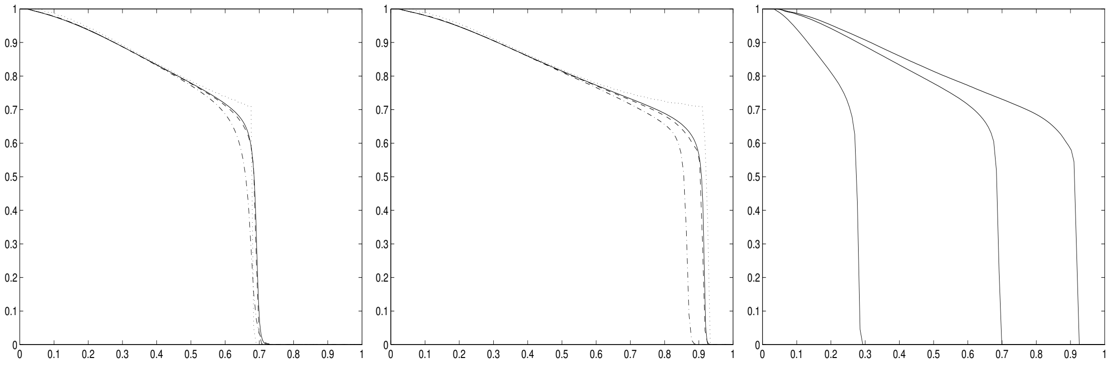


Figure 3.5. Example 1. (Left) Diagonal plot at times 0.47 (left) and 0.6 (middle); zero diffusion (hyperbolic) model solved by front tracking (dotted), reference solution (solid), OS (dashdot), and COS (dashed). (Right) COS with nonlinear diffusion coefficient (10) at times 0.1, 0.47, and 0.6. Note that there is only a slight "loss" of regularity at the "foot" of the front compared with the constant diffusion case (left and middle).

Example 2 (Several injection and production wells). The next example is also water injection into a homogeneous horizontal oil field. But now we use five injection and five production wells distributed throughout the reservoir $\Omega = [0,2] \times [0,2]$ as shown in Figure 3.8. Each injection well is marked with a circle and each production well with a cross. The scaling parameter ε is 0.0025. We show the water saturation (as contour plots) at times $t = 0.32$ and 0.8. To reach final computing time 0.8 we have used a short initial time step of 0.01 followed by 5 time steps. The reference solution is shown in Figure 3.8. Notice that the problem is less advection dominated than in the previous example due to lower injection rates.

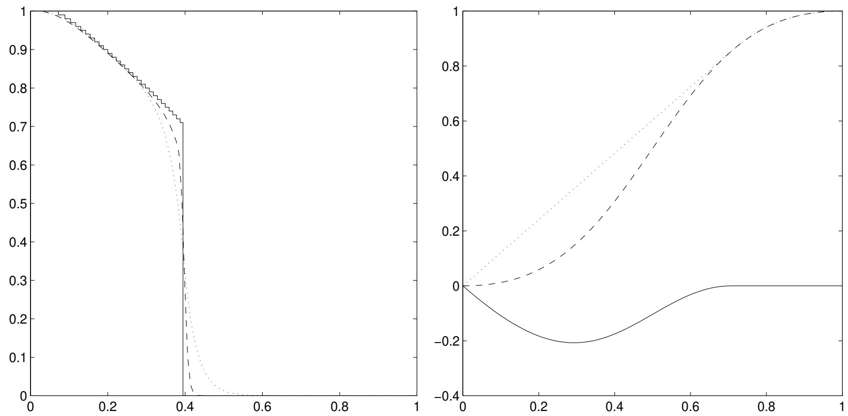


Figure 3.6. Example 1. (Left) Typical one dimensional solutions generated by front tracking (solid), OS (dotted), and COS (dashed). (Right) The flux function (dashed), the envelope function (dotted), and the residual flux function (solid) as defined by the front tracking solution. Note that the residual flux acts on the entire monotonicity interval [0,1]. We see that the effect of the residual flux term is significant (left); OS is several orders of magnitude more diffusive than COS.

Method	0.10	0.47	0.60
OS	3.4	2.1	1.8
COS	2.0	0.15	0.11
Ref	0.25	−0.025	0.021

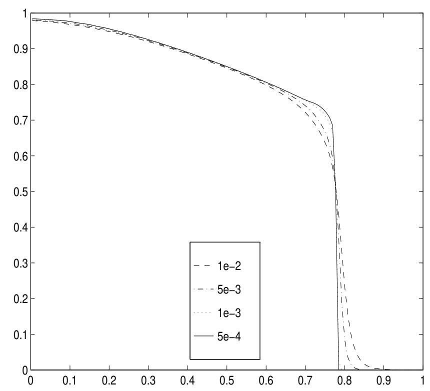


Figure 3.7. Example 1. (Left) Percentage mass balance error in the above simulations. (Right) Saturation profile at $y = 0.15$ for different values of the scaling parameter ε.

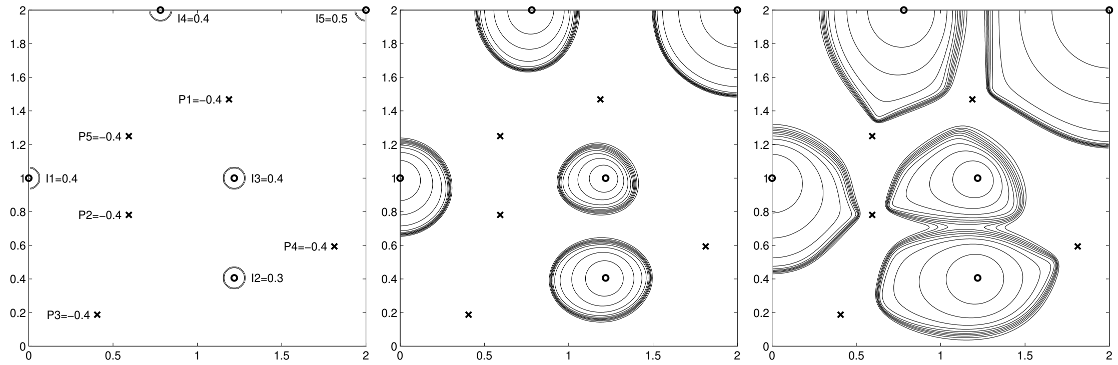


Figure 3.8. Example 2. (Left) Well configuration. The initial saturation is indicated by a few contour lines. (Middle and right) Reference solution at time 0.32 (left) and time 0.80 (right). Note that the injection rates are lower than in the previous example, so that the velocities are smaller than before and the problem becomes slightly less advection dominated.

In Figure 3.9 the OS and COS solutions are compared at time $t = 0.32$. It is evident that COS yields a more accurate picture of the flow than OS, which contains too much diffusion. We have also included a plot of the pointwise discrepancy between COS and OS to emphasise the different ability to resolve the water fronts. Similar plots are given for time $t = 0.8$ in Figure 3.10. This is immediately before water reaches some of the production wells. As before, the conclusion is that OS produces a significant amount of diffusion (splitting error) in areas of high Péclet numbers, whereas COS has virtually no viscous splitting error and therefore gives both correct placement and structure of the water fronts. When the spatial grid size is too big to represent the fronts accurately, we of course introduce an error in form of a numerical broadening corresponding to the grid size, but the solution is otherwise well behaved. However, a comparison with the zero diffusion model shows that the numerical diffusion from the projection operator is smaller than the observed diffusion in (almost) the whole reservoir.

Finally, we point out that the solution is non-monotone. This results in the generation of several residual flux functions. In Figure 3.11 (left) we have plotted a typical one-dimensional advection solution (in the x-direction) corresponding to row number 120 in the grid ($y \approx 1.88$). In the plot on the right we show the residual flux term defined by the advection solution (left). Notice that f_{res} is a function of both the space variable x and the saturation s because the solution is non-monotone. Here we must stress that the calculations leading up to the residual flux terms consume very little time compared with the rest of the algorithm. The necessary data are already contained in the data structure used by front tracking.

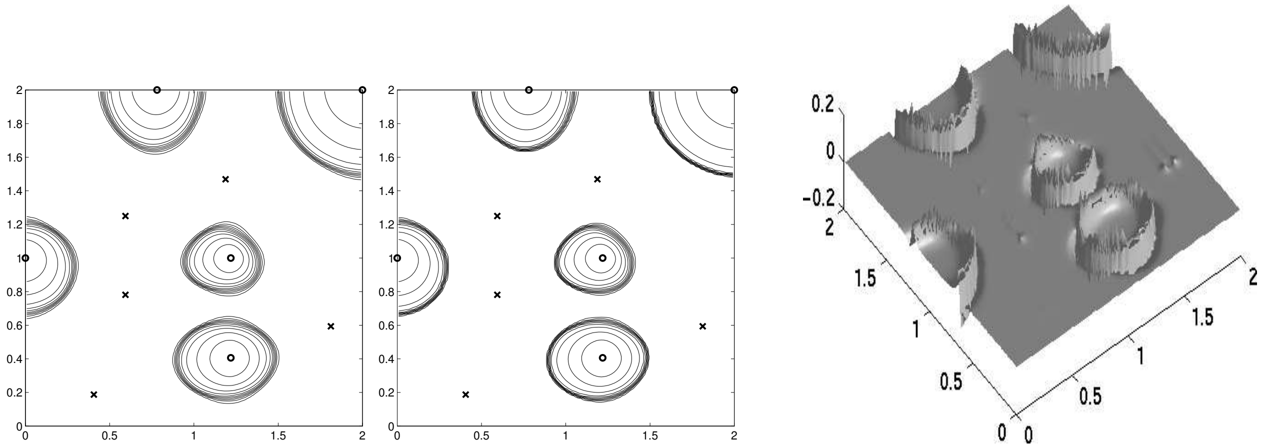


Figure 3.9. Example 2. (Left) OS solution at time 0.32. (Middle) COS solution at time 0.32. (Right) Three dimensional plot of the pointwise discrepancy between COS and OS. As before, we note that there is a difference near the water fronts, even though the problem is less advection dominated than in the previous example.

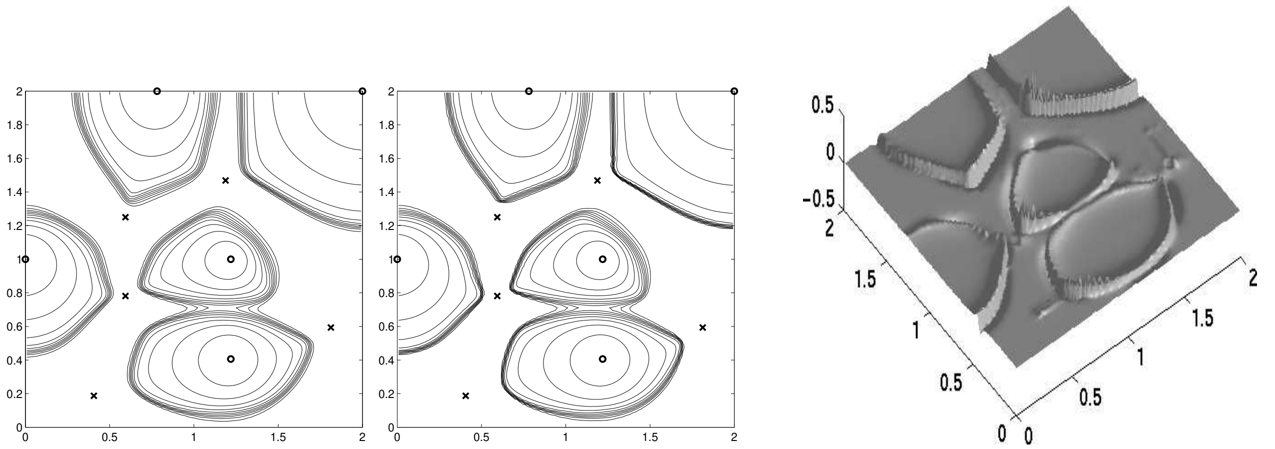
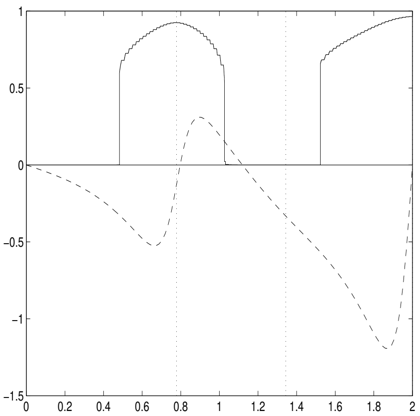


Figure 3.10. Example 2. (Left) OS solution at time 0.8. (Middle) COS solution at time 0.8. (Right) Three dimensional plot of the pointwise discrepancy between COS and OS.

4. Concluding Remarks

In reservoir simulation it is important to have a numerical strategy that can cope naturally with the different balances of advection and diffusion that will occur during a simulation. In a typical application, advective forces are highly dominant in certain parts of the reservoir whereas in other parts diffusive forces are more important. On one hand, there exists a variety of accurate numerical schemes for hyperbolic saturation equations obtained by ignoring capillary forces (advection dominated flow). On the other hand, if the flow is diffusion dominated virtually any finite element or finite difference scheme performs well.

Our starting point has been a desire to develop a numerical strategy that maintains the advantages of modern hyperbolic solvers and uses the algorithms developed when the equations are "almost hyperbolic". In addition, we wish to capture all balances of advective and diffusive forces ranging from advection to diffusion dominated problems. Numerical examples show that the usual operator splitting (OS) methodology is too simple in the sense that it does not resolve steep (self-sharpening) fronts. To amend this shortcoming, we have developed a corrected operator splitting (COS) methodology. COS models diffusion via a parabolic equation which contains a nonlinear residual flux term ensuring the correct amount of self-sharpening. The residual term turned out to be computationally easy to realize if we used Dafermos' method to solve the hyperbolic part of the problem. Numerical examples show that the COS strategy possesses the desired properties mentioned above. Finally, we have also managed to define a COS scheme in terms of a higher order Godunov scheme instead of front tracking. The computational results are satisfactory [24].



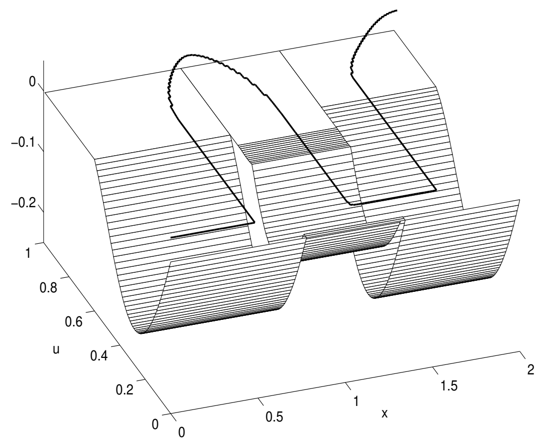


Figure 3.11. Example 2. (Left) One dimensional advection solution (solid) in the x - direction (grid row number 120). The velocity along this tube is shown as dashed. Note that the advection solution has three monotonicity intervals (indicated by the vertical dotted lines). (Right) The residual flux function (viewed as a function of space and saturation) defined by the front tracking solution (superimposed on the plot as a thick solid line). Note that the residual flux terms are relatively small (less than 0.2 in absolute value).

References

1. J. W. Barrett, K. W. Morton, *Approximate symmetrization and Petrov-Galerkin Methods for diffusion-convection problems*, Comput. Methods. Appl. Mech. Engrg. **45** (1984), 97–122.
2. J. T. Beale and A. Majda, *Rates of convergence for viscous splitting of the Navier-Stokes equations*, Math. Comp. **37(156)** (1981), 243–259.
3. J. B. Bell, G.R. Shubin, J. A. Trangenstein, *A method for reducing numerical dispersion in two-phase black oil reservoir simulation*, J. Comp. Phys. **65** (1986), 71–106.
4. J. B. Bell, G.R. Shubin, *Higher-order Godunov methods for reducing numerical dispersion in reservoir simulation*, Eight SPE Symposium on Reservoir Simulation (1985).
5. F. Bratvedt, K. Bratvedt, C. F. Buchholz, T. Gimse, H. Holden, L. Holden, N. H. Risebro, *Frontline and Frontsim: two full scale, two-phase, black oil reservoir simulators based on front tracking.*, Surv. Math. Indu. (1993), 185–215.
6. F. Bratvedt, K. Bratvedt, C. F. Buchholz, T. Gimse, H. Holden, L. Holden, R. Olufsen, N. H. Risebro, *Three-dimensional reservoir simulation based on front tracking*, North Sea Oil and Gas Reservoirs – III (Norwegian Institute of Technology, eds.), Kluwer Academic Publishers, 1994, pp. 247–257.
7. J. M. Burgers, *The nonlinear diffusion equation*, Reidel, Dordrecht, 1974.
8. G. Chavent, J. Jaffre, *Mathematical models and finite elements for reservoir simulation*, North Holland, Amsterdam, 1986.
9. C. M. Dafermos, *Polygonal approximations of solutions of the initial value problem for a conservation law*, J. Math. Anal. Appl. **38** (1972), 33–41.
10. H. K. Dahle, *Adaptive characteristic operator splitting techniques for convection-dominated diffusion problems in one and two space dimensions*, PhD. Thesis, Dept. of Math., University of Bergen, Norway (1988).
11. H. K. Dahle, M. S. Espedal, O. Sævareid, *Characteristic, local grid refinement techniques for reservoir flow problems*, International Journal for Numerical Methods in Engineering **34** (1992), 1051–1069.
12. C. N. Dawson, *Godunov-mixed methods for advective flow problems in one space dimension*, SIAM J. Num. Anal. **28(5)** (1991), 1282–1309.
13. J. Douglas, T. F. Russell, *Numerical methods for convection dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures*, SIAM J. Numer. Anal. **19** (1982), 871–885.
14. M. Espedal, R. E. Ewing, *Characteristic Petrov–Galerkin subdomain methods for two-phase immiscible flow*, Comput. Methods. Appl. Mech. Engrg. **64** (1987), 113–135.
15. R. E. Ewing, *Operator splitting and Eulerian-Lagrangian localized adjoint methods for multiphase flow*, The mathematics of finite elements and Applications VII MAFELAP (J. Whiteman, eds.), Academic Press, San Diego, 1990, pp. 215–232.
16. R. E. Ewing, T. F. Russell, *Efficient time stepping methods for miscible displacement problems in porous media*, SIAM J. Numer. Anal. **19** (1982), 1–67.
17. J. Glimm, B. Lindquist, O. A. McBryan, B. Plohr, S. Yaniv, *Front tracking for petroleum reservoir simulation*, SPE, Paper 12238, 41–49.
18. R. Hansen, M. S. Espedal, *On the numerical solution of nonlinear reservoir flow models with gravity*, International Journal for Numerical Methods in Engineering **38** (1995), 2017–2032.
19. H. Holden, L. Holden, R. Høegh-Krohn, *A numerical method for first order nonlinear scalar conservation laws in one-dimension*, Comput. Math. Applic. **15** (1988), 595–602.
20. H. Holden, N. H. Risebro, *A method of fractional steps for scalar conservation laws without the CFL condition*, Math. Comp. **60** (1993), 221–232.
21. C. Johnson, *Numerisk lösning av partiella differentialekvationer med finita element metoden*, Studentlitteratur, Lund, 1980.

22. K. Hvistendahl Karlsen, N. H. Risebro, *An operator splitting method for nonlinear convection-diffusion equations*, To appear in Numer. Math.
23. K. Hvistendahl Karlsen, N. H. Risebro, *Corrected operator splitting for nonlinear parabolic equations*, Preprint, University of Bergen (1997).
24. K. Hvistendahl Karlsen, K. Brusdal, H. K. Dahle, S. Evje, K-A. Lie, *The corrected operator splitting approach applied to a nonlinear advection-diffusion problem*, Preprint, University of Bergen (1997).
25. K. Hvistendahl Karlsen, K-A. Lie, *An unconditionally stable splitting scheme for a class of nonlinear parabolic equations*, Preprint, Institute of Mathematics and its Applications (IMA), University of Minnestota, Minneapolis, Minnesota (1997).
26. S. N. Kružkov, S. M. Sukorjanskiĭ, *Boundary value problems for systems of equations of two-phase porous flow type; statment of the problems, questions of sovability, justification of approximate methods*, Math. USSR Sbornik **33(1)** (1977), 62–80.
27. R. J. LeVeque, *Numerical methods for conservation laws*, Birkhäuser Verlag, Basel, 1992.
28. K.–A. Lie, Vidar Haugse, K. Hvistendahl Karlsen, *Dimensional splitting with front tracking and adaptive grid refinement*, Preprint (mathematics), Norwegian University of Science and Technology (1996).
29. O. Oleinik, *Discontinuous solutions of non-linear differential equations*, Amer. Math. Soc. Transl. **26** (1963), 95–172.
30. T. F. Russell, *Galerkin time stepping along characteristics for Burgers' equation*, Scientific computing (R. Stepleman, eds.), IMACS, North-Holland, 1983, pp. 183–192.
31. H. J. Schroll, A. Tveito, *Local existence and stability for a hyperbolic–elliptic system modeling two–phase reservoir flow*, Preprint RWTH Aachen **136** (1997).
32. J. S. Scroggs, *Shock-layer bounds for a singularly perturbed equation*, Quarterly of Applied Mathematics **LII(3)** (1995), 423–431.

A. Appendix

A.1. Front Tracking. We give a brief description of the front tracking method, which is an extension of Dafermos' method. Let $u(x)$ and $f(v)$ be Lipschitz continuous, piecewise linear functions. Consider a nonlinear conservation law of the form

$$\partial_t v + u(x)\partial_x f(v) = 0, \qquad v(x,0) = v_0(x), \qquad x \in \mathbb{R},\ t > 0, \tag{22}$$

where we assume that $v_0(x)$ is a step function with a finite number of jumps. The piecewise constant initial function yields a sequence of Riemann problems of the form

$$v(x,0) = \begin{cases} v_L, & \text{when } x < 0, \\ v_R, & \text{when } x > 0. \end{cases}$$

Each of these Riemann problems can be solved analytically, and since the flux function $f(v)$ is piecewise linear, the solution will be a step function. Rarefaction waves (smooth parts of the solution) are replaced by sequences of small shocks. Recall that a discontinuity (v_L, v_R) propagating with shock speed $s = (f(v_L) - f(v_R))/(v_L - v_R)$ is admissible provided the entropy condition

$$\operatorname{sign}(u(x))\,\frac{f(v) - f(v_R)}{v - v_R} \le \operatorname{sign}(u(x)) \cdot s \le \operatorname{sign}(u(x))\,\frac{f(v) - f(v_L)}{v - v_L}$$

is satisfied for all v between v_L and v_R. This condition, which is due to Oleinik [29], ensures that the solution is physically correct. To give the complete solution of the Riemann problem, assume first that $u(x)$ is positive and introduce the function

$$f_c(v\,; v_L, v_R) = \begin{cases} \text{the lower convex envelope of } f \text{ between } v_L \text{ and } v_R \text{ if } v_L < v_R, \\ \text{the upper concave envelope of } f \text{ between } v_R \text{ and } v_L \text{ if } v_L > v_R. \end{cases}$$

For negative velocities f_c is defined similarly with 'lower convex' and 'upper concave' interchanged. Suppose now that $\{v_i\}$ are the breakpoints of $f(v)$, i.e., the points where $f'(v)$ is discontinuous. Since $f(v)$ is piecewise linear, then so is $f_c(v)$, and the breakpoints $\{\overline{v}_i\}$ of $f_c(v)$ form a subset of $\{v_i\}$. If we set $\overline{v}_1 = v_L$ and $\overline{v}_M = v_R$, then the solution of the Riemann problem is given by

$$v(x,t) = \begin{cases} v_L, & x < \overline{x}_0(t) \\ \overline{v}_i, & \overline{x}_i(t) \le x \le \overline{x}_{i+1}(t), \qquad i = 1, \dots, M-1 \\ v_R, & x > \overline{x}_{M-1}(t), \end{cases}$$

where $\overline{x}_i(t)$ satisfies the Rankine–Hugoniot condition

$$\frac{d\overline{x}_i}{dt} = u(\overline{x}_i)\frac{\bar{f}_{i+1} - \bar{f}_i}{\overline{v}_{i+1} - \overline{v}_i} = u(\overline{x}_i) \cdot \overline{s}_i, \qquad i = 0, \dots, M-1.$$

and $\bar{f}_i = f_c(\overline{v}_i; v_L, v_R)$. Suppose now that the velocity is given by $u(x) = a_j x + b_j$ on intervals $[x_j, x_{j+1}]$ for nonzero a_j; the intervals are given such that u has constant sign on each. The path of a shock with associated speed s starting from $x_0 \in [x_j, x_{j+1}]$ at time t_0 is then given by the explicit formula

$$x(t) = x_0 e^{a_j s(t-t_0)} + \frac{b_j}{a_j}(e^{a_j s(t-t_0)} - 1).$$

By connecting the solutions of the local Riemann problems we obtain the global solution of (22) up to the first time when two waves from neighbouring Riemann problems interact. This interaction defines a new Riemann problem with left and right states given by the values immediately to the left and right of the collision point. The new Riemann problem is then solved as outlined above, thereby giving the global solution until the next interaction occurs, and so on. Summing up, the front tracking algorithm goes as follows:

(1) Solve the Riemann problems defined by the piecewise constant initial data.
(2) Keep track of shock collisions and solve Riemann problems arising at the collision points.

For a more detailed treatment of the front tracking method for $u(x) \equiv 1$ we refer to [19].

Remark. The front tracking method for general conservation laws (arbitrary u, f, and v_0) consists in replacing u and f with piecewise linear approximations and v_0 with a piecewise constant approximation, and then to solve the resulting (perturbed) problem exactly according to the procedure described above.

A.2. The Petrov–Galerkin Finite Element Method. Consider the nonlinear parabolic equation (15), which can be written in conservative form as follows

$$\begin{aligned} \partial_t w + \partial_x \left[u_h(x) f_{\text{res}}(x,w) - \varepsilon\nu(w)\partial_x w\right] \\ = \partial_x u_h(x) f_{\text{res}}(x,w). \end{aligned} \tag{23}$$

By "freezing" the coefficients in this equation we obtain linear equations which can be solved by a Petrov–Galerkin finite element method. To be more precise, let w^p be the Petrov–Galerkin approximations at time $t = \Delta t$ to the linear, variable coefficients, advection–diffusion equation

$$\begin{aligned} &\partial_t w^p + \partial_x \left[u_h(x) b_\delta^n(x, w^{p-1}) w^p - \varepsilon\nu(w^{p-1})\partial_x w^p\right] \\ &= \partial_x u_h(x) f_{\text{res}}(x, w^{p-1}), \qquad p = 1, 2, \dots, \end{aligned}$$

where $f_{\text{res}}(x,w) = b_\delta^n(x,w)w$. The approximate solution to (23) is now taken to be w^q for some $q > 1$.

The rest of this section is devoted to describing the Petrov–Galerkin scheme for problems of the form

$$\partial_t w + \partial_x \left[b(x)w - \varepsilon d(x)\partial_x w\right] = r(x), \qquad (x,t) \in (a,b) \times (0, \Delta t], \tag{24}$$

with initial and, say, homogeneous boundary data; $w|_{t=0} = \overline{w}$ and $w|_{x=a,b} = 0$, respectively.

We are interested in computing approximations to (24) at time $t = \Delta t$. Let S_h be the standard piecewise linear finite element space spanned by hat functions $\{\theta_i(x)\}$ with nodes $\{x_i\}$, so that boundary conditions automatically are taken care of, and let $\Delta x_i = x_i - x_{i-1}$ be the size of each element. Hence, a finite element approximation $w_h \in S_h$ may be written

$$w_h = \sum_i w_i \theta_i, \qquad w|_{t=0} = \sum_i \overline{w}_i \theta_i,$$

for some proper coefficients $\{w_i\}$ and $\{\overline{w}_i\}$. The time derivative in (24) is replaced by a single backward Euler step. Note that this implies that Δt must be reasonably small to ensure that the method of "freezing" the coefficients converges, see [18]. For larger Δt it may be necessary do several Euler steps.

Multiplying equation (24) by a test function $\psi_i(x)$ and subsequently doing an integration by parts, leads to the variational formulation

$$(w_h, \psi_i) + \Delta t \left(b(x)w_h - \varepsilon d(x)\partial_x w_h, \partial_x \psi_i\right) = (\overline{w} + r\Delta t, \psi_i), \tag{25}$$

where $(\cdot,\cdot)$ denotes the usual L_2 inner product on (a,b).

Since (25) is not completely symmetrized by the advection step because of the b - term, we have to choose a test space T_h different from the trial space S_h to stabilize equation (25). A theory for choosing an appropriate test space is given in [1]. This consists in choosing a discrete test space that transforms the bilinear form (25) into an equivalent V - elliptic and symmetric bilinear form. Based on previous experience (e.g. [10]) a good choice seems to be quadratic functions with support on $[x_{i-1}, x_{i+1}]$,

$$\psi_i(x) = \begin{cases} \theta_i + c_{i-1/2}\sigma_i, & x \in [x_{i-1}, x_i]\,, \\ \theta_i + c_{i+1/2}\sigma_i, & x \in [x_i, x_{i+1}]\,, \end{cases}$$

where

$$\sigma_i(x) = \begin{cases} (x - x_{i-1})(x - x_i)/\Delta x_i, & x \in [x_{i-1}, x_i]\,, \\ -(x - x_i)(x - x_{i+1})/\Delta x_{i+1}, & x \in [x_i, x_{i+1}]\,, \end{cases}$$

and

$$c_{i-1/2} = 3\left(\frac{2}{\beta_{i-1/2}} - \coth\left(\frac{\beta_{i-1/2}}{2}\right)\right).$$

Here,

$$\beta_{i-1/2} = \frac{b_{i-1/2}\Delta x_i}{\varepsilon d_{i-1/2}}$$

is the mesh Péclet number on element $[x_{i-1}, x_i]$, and $b_{i-1/2}$ and $d_{i-1/2}$ are average values over this element. Using the above notation, the finite element approximation now reads;

Find $w_h(x) \in S_h$ so that (25) is satisfied for each $\psi_i(x) \in T_h$.

Note that a mesh Péclet number has to be computed for each component of the velocity field on each element.

(Kenneth Hvistendahl Karlsen)
Department of Mathematics
University of Bergen
Johs. Brunsgt. 12
N–5008 Bergen, Norway
E-mail address: kenneth.karlsen@mi.uib.no

(Knut–Andreas Lie)
Department of Mathematical Sciences
Norwegian University of Science and Technology
N–7034 Trondheim, Norway
E-mail address: andreas@math.ntnu.no

(Nils Henrik Risebro)
Department of Mathematics
University of Oslo
P.O. Box 1053, Blindern
N–0316 Oslo, Norway
E-mail address: nilshr@math.uio.no

(Johnny Frøyen)
Department of Mathematics
University of Bergen
Johs. Brunsgt. 12
N–5008 Bergen, Norway
E-mail address: johnny.froyen@mi.uib.no