NUMERICAL SOLUTION OF THE POLYMER SYSTEM BY FRONT TRACKING

V. HAUGSE, K. H. KARLSEN, K.–A. LIE, AND J. R. NATVIG

Abstract. The paper describes the application of front tracking to the polymer system, an example of a nonstrictly hyperbolic system. Front tracking computes piecewise constant approximations based on approximate Riemann solutions and exact tracking of waves. It is well–known that the front tracking method may introduce a blowup of the initial total variation for initial data along the curve where the two eigenvalues of the hyperbolic system are identical. It is demonstrated by numerical examples that the method converges to the correct solution after a finite time that decreases with the discretization parameter.

For multidimensional problems, front tracking is combined with dimensional splitting and numerical experiments indicate that large splitting steps can be used without loss of accuracy. Typical CFL numbers are in the range 10 to 20, and comparisons with Riemann free, high–resolution methods confirm that the high efficiency of front tracking.

The polymer system, coupled with an elliptic pressure equation, models two-phase, three-component polymer flooding in an oil reservoir. Two examples are presented, where this model is solved by a sequential time stepping procedure. Because of the approximate Riemann solver, the method is non-conservative and CFL numbers must be chosen only moderately larger than unity to avoid substantial material balance errors generated in near-well regions after water breakthrough. Moreover, it is demonstrated that dimensional splitting may introduce severe grid orientation effects for unstable displacements that are accentuated for decreasing discretization parameters.

1. Introduction

Front tracking has proved to be a very efficient numerical method for one–dimensional hyperbolic conservation laws, both for scalar problems [5] and for systems [17, 18]. By front tracking we mean an algorithm that is based on tracking waves from a series of initial Riemann problems and solving new Riemann problems when waves interact, unlike Glimm's [3] method that samples the solution before waves interact. The front tracking method is extended to multidimensions by dimensional splitting and gives a very efficient numerical method. This has been documented for scalar problems [15, 14], gas dynamics [6], and shallow water equations [4].

For *nonstrictly* hyperbolic systems, serious doubt [24] has been cast on the use of front tracking and other numerical methods based on piecewise constant approximations and

Date: March 11, 1999.

Key words and phrases. Front tracking, nonstrictly hyperbolic systems, polymer flooding, dimensional splitting.

Riemann solutions. The polymer system [12]

$$
\begin{aligned}
s_t + f(s,c)_x &= 0 \\
(sc + a(c))_t + \big(cf(s,c)\big)_x &= 0,
\end{aligned} \tag{1}
$$

is a nonstrictly hyperbolic where the two eigenvalues of the system are identical along a *transition curve* $s = s^T(c)$ in state space. For this system, blowup of the initial total variation was demonstrated for continuous initial data on the transition curve. These observations suggest that such piecewise constant approximations may be inappropriate from a *mathematical* point of view, e.g., when trying to prove existence along the lines of Glimm's existence proof.

In this paper, we discuss this special case from a *numerical* point of view, i.e., with a limited range of decreasing discretization parameters, and demonstrate that despite the large initial total variation, the approximate solutions converge to the same solution as the upwind method as time increases. In particular, the influence of the initial blowup in total variation decreases more rapidly in time with decreasing discretization parameters.

Risebro and Tveito [19] demonstrated that front tracking is more efficient than the upwind method and the random choice scheme for one-dimensional Cauchy problems. We have performed numerical experiments that confirm the high efficiency, except for the case with initial data on the transition curve, where the method is CPU intensive. These observations motivate the main objective of the paper: the extension of front tracking to multidimensions by dimensional splitting for the polymer system, as an example of a nonstrictly hyperbolic system.

The one-dimensional polymer system has been analysed by many authors. Isaacson [8] solved the Riemann problem for the case with no adsorption ($a(c) \equiv 0$) and Johansen and Winther [12] generalized the results by including adsorption. Temple [22] and Isaacson and Temple [9] proved the existence of a weak solution of the Cauchy problem with initial data of bounded variation in the case of no adsorption. The existence of a unique and stable solution of the same system was proved by Tveito and Winther [23] for initial data that are constant outside an interval and with c sufficiently smooth and by Klingenberg and Risebro [13] with no smoothness assumptions.

The polymer system arises as part of the fractional flow formulation of two–phase, three–component model of polymer flooding in an oil reservoir. Polymer flooding is an enhanced oil recovery process, where polymer is added to some of the injection water. Polymer increases the viscosity of the water, thereby reducing the mobility ratio between water and oil. Due to the additional cost, polymer is typically injected in slugs. The simplified mathematical model of polymer flooding is given by

$$
\begin{aligned}
s_t + U \cdot \nabla f(s,c) &= f(\hat{s},\hat{c})q, \\
(sc + a(c))_t + U \cdot \nabla (cf(s,c)) &= \hat{c}f(\hat{s},\hat{c})q, \\
\nabla \cdot (k\lambda(s,c)\nabla p) &= q,
\end{aligned} \tag{2}
$$

where (s,c) denotes water saturation and polymer concentration in the water, $f(s,c)$ is the fractional flow function of water, $a(c)$ is the adsorption function, and q is a source/sink term. The total velocity U is given by $U = -k\lambda(s,c)\nabla p$, where k is permeability, $\lambda(s,c)$ is total mobility, and p is pressure. For injection wells, $(\hat{s},\hat{c})$ denotes the injected saturation and concentration, whereas $(\hat{s},\hat{c})$ is the saturation and concentration at the well for a production well. Note that the net production, $\int q(x,y)\,dxdy$, must vanish as both fluids and the rock are assumed to be incompressible.

A common strategy to solve (2) is to use a sequential stepping procedure, where one assumes that the total velocity is slowly varying in time compared with the variation of the saturation and concentration. One time step in the algorithm consists of first solving the elliptic pressure equation with the total mobility given by the latest saturation and concentration. The hyperbolic system is then solved by assuming that the total velocity is constant during the time step. For increased efficiency, the hyperbolic equations may be solved several times before the pressure solution is updated.

It has earlier been documented that conventional upstream–weighted finite–difference schemes may be unsuited for the hyperbolic part of (2) due to large amounts of numerical diffusion, and in some cases, substantial grid orientation effects. Holing et al. [7] developed a second-order Godunov–type method for the polymer system. Their scheme reduces the smearing of fronts and grid orientation effects compared with standard upwind schemes. Front tracking combined with dimensional splitting is known to compute solutions with very little numerical diffusion. The second objective of our paper is therefore to investigate the performance of the method as part of a sequential stepping approach for the polymer flooding model (2). To this end, we present two numerical examples that assess numerical diffusion, grid orientation effects and other numerical characteristics of the scheme.

The outline of the paper is as follows. Front tracking, dimensional splitting, and the sequential stepping method for (2) are presented in Section 2. Section 3 contains four numerical examples. The first example is a one-dimensional Cauchy problem with initial data on the transition curve. The second is a Riemann problem for the two-dimensional version of (1), where front tracking is compared with the upwind method and a second-order, non-oscillatory, central-difference scheme. The two last examples are of the full polymer model (2) and describe injection of water followed by polymer and miscible displacement with adverse mobility ratio. Both processes are examined on a five-spot well pattern. Finally, conclusions based on our numerical experiments are given in Section 4.

2. The Front Tracking Method

In this section we review the front tracking algorithm in one spatial dimension and present its extension to two dimensions by dimensional splitting. We also present a sequential stepping method for (2) using dimensional splitting for the hyperbolic system and finite differences for the pressure equation.

2.1. **Front Tracking in One Dimension.** Front tracking is an algorithm for computing piecewise constant approximations to a conservation law. The method is free of an underlying grid, is unconditionally stable, and has no intrinsic time step. The method was first introduced by Dafermos [2] to study scalar equations and later used as a computational tool by several authors. Holden, Holden and Høegh–Krohn [5] proved that the method is well-defined for an arbitrary flux function and suggested to use it as a reliable numerical method. Risebro [17] generalized the method to systems of hyperbolic conservation laws and Risebro and Tveito [19] applied the method to the one-dimensional polymer system.

The basic building block in the front tracking method is the solution of Riemann problems, i.e. initial value problems with data of the form

$$(s(x,0),c(x,0)) = \begin{cases} (s_L, c_L) & \text{for } x < 0, \\ (s_R, c_R) & \text{for } x > 0. \end{cases}$$

The Riemann solution is a similarity solution, i.e. a function of $\xi = x/t$ only. The Riemann problem for the polymer system is discussed in detail by Johansen and Winther [12]. The solution consists of a set of constant states connected by s–waves and c–waves. A s–wave may be composed of a shock and a rarefaction wave, while a c–wave is either a shock or a rarefaction wave. The concentration is constant in s–waves, and the hyperbolic part of the polymer system is reduced to a scalar conservation law. Both the saturation and concentration are varying in c–waves, and an ordinary differential equation must be solved to compute c–rarefaction waves. The complexity of the Riemann solution depends on where the left and right states are located in state space. The solution may consist of up to five constant states connected by s–waves and c–waves.

We seek piecewise constant solutions and must therefore approximate each Riemann solution by a step function in ξ. Shocks from the analytical solution are kept unchanged, while rarefaction waves are approximated by a series of small shocks. A rarefaction wave with left and right velocities ξ_L and ξ_R will be approximated with N shocks, where $N = \lceil(\xi_R - \xi_L)/\delta\rceil$. (The parameter δ is user–defined.) This way, each Riemann problem is approximated by a sequence of jump discontinuities that travel with a finite wave speed. Each jump discontinuity is called a front and a Riemann solution is represented by a list of fronts sorted by increasing waves speed.

For piecewise constant initial data, the global solution of the Cauchy problem is constructed by connecting the solution of the individual Riemann problems. The solution consists of constant states separated by space–time rays. There will be a first time when two or more space–time rays intersect. This defines a new Riemann problem that can be approximated as outlined above. Thus, the front tracking method solves the Cauchy problem by keeping track of all fronts and by solving the Riemann problems that arise. The method is well–defined as long as the number of waves is finite and is very efficient due to the highly accurate resolution of discontinuities. However, the approximation of rarefaction waves may introduce small material balance errors. If higher accuracy is needed in rarefaction regions, one can track rarefaction regions instead, as introduced by Wendroff [26] for chromatography. Notice also that the algorithm does not need a fixed grid, except for possibly specifying the piecewise constant initial data.

For general initial data, we project the initial function by a quadrature method onto the uniform grid with nodes $\{i\Delta x\}$ to define the piecewise constant initial data.

We refer to Risebro and Tveito [19] for further details on the implementation of front tracking for the polymer system.

2.2. **Front Tracking in Two Dimensions.** The two-dimensional version of (1) reads,

$$u_t + g(u)_x + g(u)_y = 0,$$

where $u = (s, sc + a(c))$ is the conserved variable and $g = (f(s,c), cf(s,c))$ is the corresponding flux. The Godunov form of dimensional splitting is

$$u^{n+1} = \pi\ \mathcal{S}^y(\Delta t^n)\ \pi\ \mathcal{S}^x(\Delta t^n)\ u^n,$$

where $\mathcal{S}^x$ and $\mathcal{S}^y$ are the front tracking solutions of (1) in x and y direction, respectively. The operator π is the usual grid block averaging operator. The underlying (fixed) grid is introduced to simplify the process of obtaining piecewise constant initial data for the y step from the solutions computed in the x step and vice versa.

2.3. **A Sequential Stepping Method.** Our numerical method for (2) uses an IMPES approach (Implicit Pressure, Explicit Saturation). To advance the solution to the next time step, the elliptic pressure equation is solved first with total mobility from the previous time step. Subsequently, the total velocity field from the pressure solution is used to solve the hyperbolic equations for saturation and concentration.

Let (p^n, s^n, c^n) denote the solution at time t^n and let Δt^n be the next time step. The numerical solution at time $t^n + \Delta t^n$ is then given by the following steps.

2.3.1. *Pressure Update.* The total mobility at time t^n is given by $\Lambda^n(x,y) = \lambda(s^n, c^n)$, and the pressure equation is then

$$\nabla \cdot (k(x,y)\Lambda^n(x,y)\nabla p) = q.$$

This equation is discretized by a finite–difference method on a block–centred grid. The transmissibilities in our pressure solver is based on the nine–point scheme of Yanosik and McCracken [25]. Upstream weighting was used in both the elliptic and hyperbolic equations by Yanosik and McCracken, while Potempa [16] developed a hybrid finite difference, finite–element method with upstream weighting only in the hyperbolic equation. We follow Potempa's approach, and use a harmonic average on the mobility. A simple well model is implemented, where each well is connected to the grid in only one grid block. All wells are controlled by specified injection/production rates.

The linear solver is based on the conjugate gradient method, preconditioned by a V–cycle multigrid method. Due to the preconditioner, our numerical results are performed on grids with $2^N + 1$ blocks in each direction. A conservative flux consideration is used to find velocities at the centre of the grid block boundaries.

2.3.2. *Saturation and Concentration Update.* The saturation and concentration are advanced to the next time level using the constant velocities determined in the pressure update. The front tracking and dimensional splitting approach introduced above can be extended to equations with variable coefficients, see [14]. Based on the velocities at the grid block boundaries, we reconstruct a velocity field $(U^n, V^n)(x,y)$, where U^n (V^n) is piecewise linear in x (y) and piecewise constant in y (x). The flow in areas with no production/injection is now given by

$$u_t + U^n(x,y)g(u)_x + V^n(x,y)g(u)_y = 0,$$

and the Godunov form of dimensional splitting is

$$u^{n+1} = \pi\, \mathcal{S}^{V^n}(\Delta t^n)\, \pi\, \mathcal{S}^{U^n}(\Delta t^n)\, u^n,$$

where $\mathcal{S}^{V^n}$ and $\mathcal{S}^{U^n}$ are the front tracking solutions of the equation

$$u_t + v(x)g(u)_x = 0,$$

with velocities V^n and U^n, respectively. The total production (from wells) is found by exact integration in time of water and polymer flux over the grid boundaries of each well.

2.3.3. *Time Step Estimation.* The front tracking method is unconditionally stable, but too large time steps may produce inaccurate solutions due to splitting and material balance errors in the hyperbolic equations and decoupling of the elliptic and hyperbolic equations during each time step. Note that the velocity field changes slowly in cases where the variation in total mobility is small. The user–defined time step for the sequential stepping may therefore be chosen larger than the dimensional splitting step for the saturation/concentration equations. The latter time step is estimated from a user–defined CFL target. During the update of the hyperbolic equations, maximum speed of information in each direction is computed. The maximum speed in x–direction is given by

$$\mu_x^n = \max_{i,j}(U_{i,j}^n \xi_{i,j})$$

where $U_{i,j}^n$ is maximum velocity and $\xi_{i,j}$ is maximum wave speed in x–direction for grid block (i, j). In particular, ξ is a by-product of the Riemann solution process. The maximum speed is updated when determining possible wave interactions in the front tracking algorithm. A corresponding estimate is used in the y–direction. The length of the next time step is then given by

$$\Delta t^{n+1} = \mathrm{CFL} \cdot \min_{x,y} \left(\Delta x/\mu_x^n, \Delta y/\mu_y^n\right),$$

where CFL is the user-defined target.

3. Numerical Results

The front tracking algorithm has been tested on a variety of examples. Here we present four examples, starting with a one-dimensional Cauchy problem with initial data on the transition curve. This problem is a "worst case" for front tracking, since the total variation after the initial solution of Riemann problems approaches infinity as the number of constant states increases. Front tracking results are compared with results computed by the upwind scheme. The second example is two-dimensional with constant velocities, where a non–trivial Riemann problem is examined. The last two examples are two-dimensional with coupling between the elliptic and hyperbolic equations.

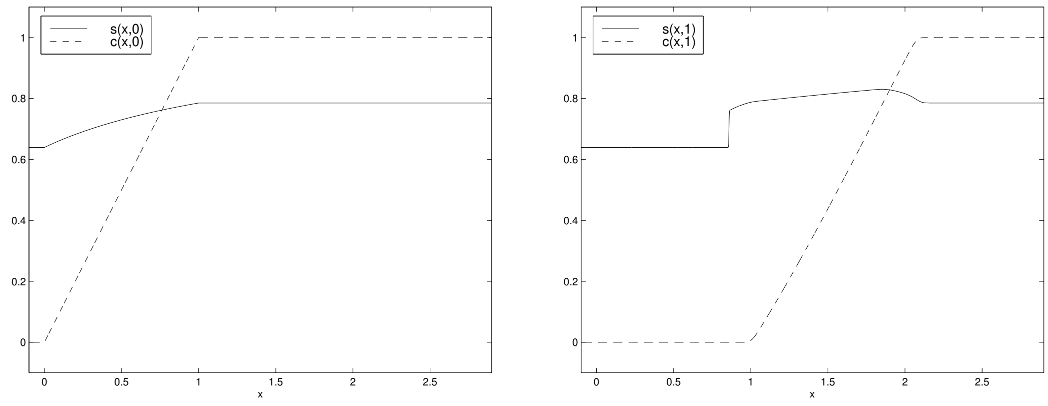


FIGURE 1. Initial saturation and concentration for the Cauchy problem with initial data on the transition curve (left). Numerical solution computed by the upwind method at $t=1$ with $\Delta x = 0.001$ and CFL $= 0.95$ (right)

3.1. **Data on the Transition Curve.** Tveito and Winther [24] argue that numerical methods based on a Riemann solver may have problems with initial data on the transition curve. The total variation of the numerical solution at time 0^+ will approach infinity as the number of constant states in the approximation of the initial data increases. We present numerical examples based on front tracking and an upwind finite–difference method, and show that the front tracking solutions converge to the same solution as the finite–difference solution.

Consider the following Cauchy problem:

$$f(s,c) = \frac{s^2}{s^2 + (1-s)^2(1+c)}, \qquad a(c) = \frac{0.2c}{1+c},$$

with initial data

$$c_0(x) = \begin{cases} 0 & \text{for } x < 0, \\ x & \text{for } 0 < x < 1, \\ 1 & \text{for } 1 < x, \end{cases} \qquad s_0(x) = s^T(c_0(x)).$$

We use a finite–difference approximation as a reference solution of the Cauchy problem. Godunov's method, which is identical to the upwind method for the polymer system, was chosen as numerical method. The discretization of the polymer system with this method is given by Johansen et al. [11]. Figure 1 shows initial data and solution at $t=1$ computed by the upwind method on a fine grid.

Consider now the numerical solution computed by front tracking. The solution of each of the initial Riemann problems consists of one s–shock and one c–rarefaction. The strength of the first s–shock is related to the difference in concentrations for the Riemann problem: $|\Delta s| = \mathcal{O}(|c_L - c_R|^{1/2}) = \mathcal{O}(|\Delta x|^{1/2})$, while the number of Riemann problems is $\mathcal{O}(|\Delta x|^{-1})$. Front tracking solutions for $\Delta x = 0.1$ and 0.01 are shown in Figures 2 and 3, respectively. The solutions before waves from neighbouring Riemann problem have collided contain

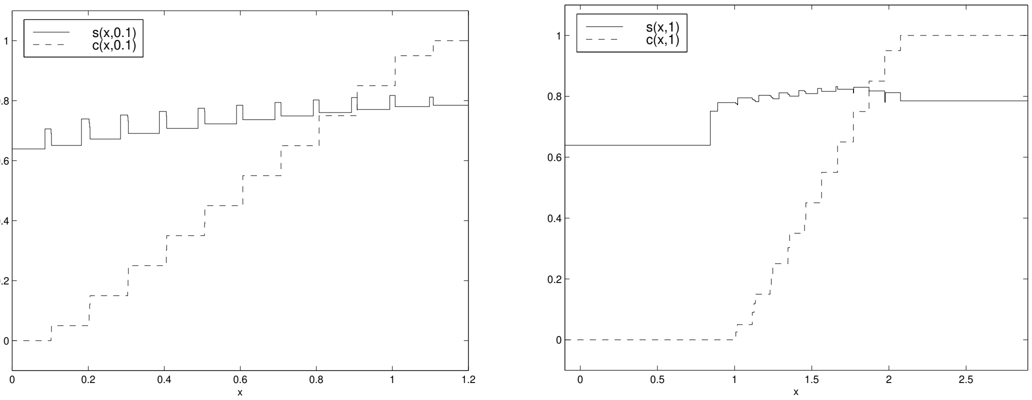


FIGURE 2. Front tracking solution for the Cauchy problem with initial data on the transition curve are shown after time 0.1 (left) and 1 (right) with initial discretization parameter $\Delta x = 0.1$.

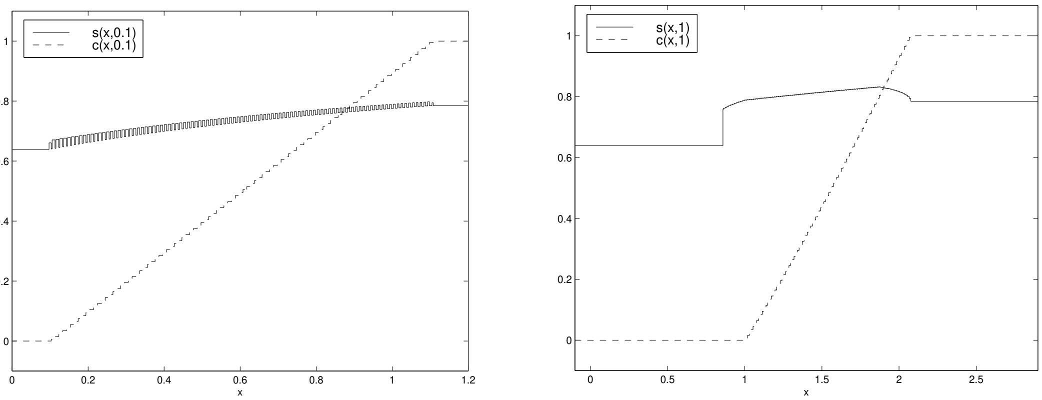


FIGURE 3. Front tracking solution for the Cauchy problem with initial data on the transition curve are shown after time 0.1 (left) and 1 (right) with initial discretization parameter $\Delta x = 0.01$.

oscillations and has large variation. However, after a few wave interactions, the total variation is dramatically reduced. In Figure 2, the total variation is reduced from time 0.1 to 1, but the saturation profile still contains oscillations at time 1. On the other hand, no oscillations are seen in the saturation profile at time 1 in Figure 3, and the front tracking solution is close to the finite–difference solution in Figure 1. Decreasing Δx further we observe the same tendency: the initial variation increases but disappears faster. These results seem to be little influenced by the parameter δ (determining the approximation of each c–rarefaction). Similar observations have been made for different orientation of the initial data along the transition curve. Moreover, if the front tracking algorithm is combined with repeated projections onto a grid, the numerical solution leaves the transition curve as a combined effect of interactions between fronts and projections.

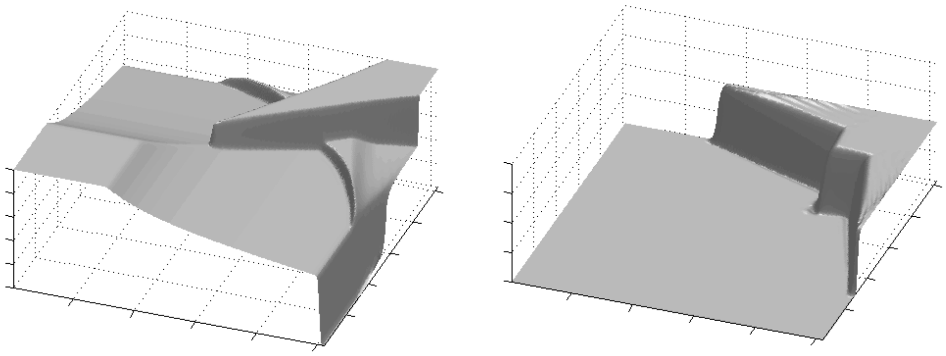


FIGURE 4. Solution at time $t = 0.5$ for the two-dimensional Riemann problem; water saturation (left) and polymer concentration (right).

3.2. **A Two-Dimensional Riemann Problem.** In this example we study the two-dimensional equation $u_t + g(u)_x + g(u)_y = 0$ on the unit square in the first quadrant with fractional flow and adsorption functions as in the previous example. The initial data is the Riemann problem

$$u_0(x) = (s,\, c)(x) = \begin{cases} (1.0,\, 0.0) & x < 0.3,\ y < 0.3, \\ (1.0,\, 0.1) & x > 0.3,\ y > 0.3, \\ (0.0,\, 0.0) & \text{otherwise.} \end{cases}$$

The physical setting consists of two quadratic water plugs, where the leading plug contains polymer with a concentration of 0.1. Since pure water flows more easily than water containing polymer, the pure water will overtake the polymer slug and overrun it along the sides, giving rise to a complicated wave pattern.

We compare the front tracking method with two finite–difference methods; the first-order upwind method and the second-order, non–oscillatory, central difference scheme of Jiang and Tadmor [10]. Since neither of these schemes use Riemann solvers, they should be rather fast and represent a fair match for the front tracking method.

Figure 4 shows the solution at time $t = 0.5$ computed by the Jiang–Tadmor scheme on a 256×256 grid. Notice that the water finger extending backwards from the polymer plug becomes very narrow as it approaches the stationary pure water plug and is therefore not represented completely on this grid.

To measure efficiency we plot the observed error versus runtime for a sequence of grids; $2^n \times 2^n$ for $n = 5, \dots, 8$. Figure 5 gives such a comparison for front tracking (with various CFL numbers), the upwind method, and the central difference scheme with MM_1 and UNO limiters (see [10]). For the finite–difference schemes we use CFL number 0.475. Errors are measured relative to a reference solution computed on a $2^{10} \times 2^{10}$ grid by the Jiang–Tadmor scheme.

We see from the plot that both runtime and error for the front tracking method decreases as the CFL number is increased above unity. This is in correspondence with observation

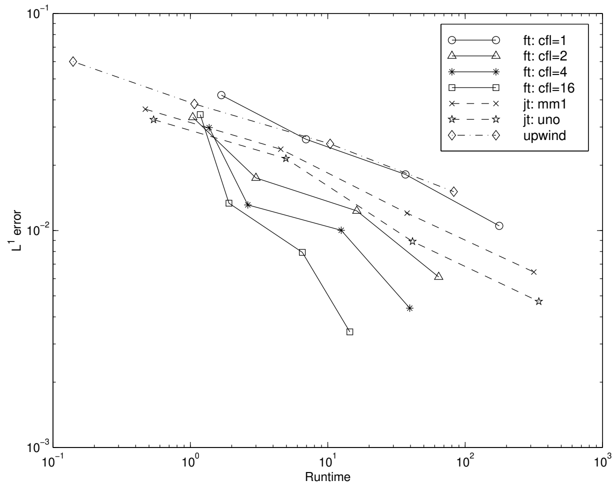


FIGURE 5. L^1 error in physical quantities (s, c) versus runtime.

made by Lie et al. [15] for scalar problems and by Lie [14] for scalar equations with a velocity field without sources/sinks; increasing the CFL number gives less numerical diffusion and lower runtime without a counterbalancing increase in splitting errors. In fact, extensive numerical experiments show that optimal CFL numbers in the range 10–20 are common when comparing numerical error versus runtime.

For CFL numbers of 2 and larger, the front tracking scheme is more efficient than the second-order central scheme for this problem. For CFL number 1, front tracking and the upwind scheme performs almost similarly. Both front tracking and the upwind scheme are first order, but very different in nature. The upwind scheme is very simple and consists of a double loop with one function evaluation at each grid point. The front tracking scheme, on the other hand, uses a complicated Riemann solver which involves solution of nonlinear equations and ordinary differential equations. It thus gives a strong indication of the computational efficiency of front tracking that its worst performance is equal to the upwind method.

3.3. **Water Injection Followed by Polymer Injection.** The next example is closer to a real–life application. Pure water is injected until an accumulated amount of 0.5 pore volumes is reached, followed by injection of water containing a unity polymer concentration. The reservoir is homogeneous with a repeated five–spot well pattern. Positions of the producers and injectors are shown in Figure 6. Reported productions and material balance errors for parallel grids are always divided by two to be consistent with the diagonal grid. The flow functions

$$f(s,c) = \frac{s^2}{s^2 + (1-s)^2(1+4c)/2}, \qquad \lambda(s,c) = 2s^2/(1+4c) + (1-s)^2$$

model displacement of an oil with unit viscosity by water with viscosity equal 0.5 for pure water and 2.5 for maximum polymer concentration. All numerical results for this example

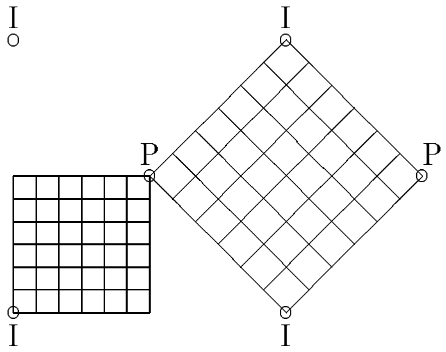


FIGURE 6. Diagonal (left) and parallel grid (right) of a five–spot repeated well pattern. The production and injection wells are denoted by P and I, respectively.

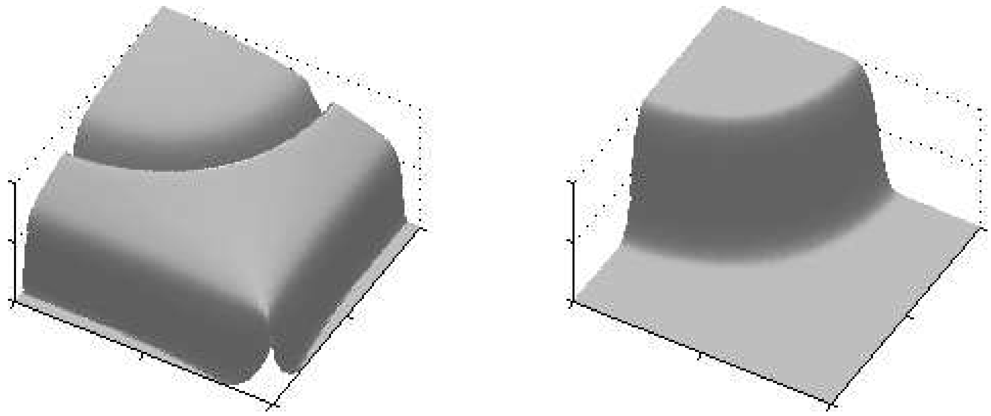


FIGURE 7. Saturation (left) and concentration (right) are plotted after injection of 0.8 pore volumes of water. Injection of 0.5 pore volumes of pure water was followed by injection of water containing polymer. This solution was calculated using a CFL number of 2 on a diagonal grid.

are computed on a 129×129 grid with a time step of 0.02 between the each update of the velocity field and a rarefaction approximation parameter $\delta = 0.01$.

Water saturation and polymer concentration after injection of 0.8 pore volumes are shown in Figure 7. Note that an oil bank develops in front of the injected polymer. The water cut of the producers are reduced when the oil bank reaches the wells, and the recovery is accelerated.

The previous example indicated that large CFL numbers can be used without loss of accuracy for the polymer system with constant velocity fields. A natural question is therefore whether this also is valid for the full model (2). Since an accurate reference solution is hard to compute, we will use the material balance error and the amount of oil remaining in the reservoir as a quality measure for the approximate solutions.

We ran three simulations with CFL numbers 1, 2, and 4 on the diagonal and on the parallel grid. Table 1 shows material balance errors and remaining amount of oil measured

TABLE 1. Water and polymer material balance and oil remaining in the reservoir after injection of two pore volumes for different CFL numbers and grid orientations.

Grid	CFL	Water material error	Polymer material error	Remaining oil
Diagonal	4.0	0.01519	0.01079	0.06449
Diagonal	2.0	0.00534	0.00280	0.06440
Diagonal	1.0	-0.00103	0.00032	0.06440
Parallel	4.0	-0.01602	-0.00510	0.06826
Parallel	2.0	-0.00540	-0.00044	0.06809
Parallel	1.0	-0.00031	0.00041	0.06809

after two pore volumes injected. The water material balance error is here defined as the difference in the net injection of water and net increase of water remaining in the reservoir, and a corresponding definition is used for polymer. Note that the material balance error is always positive for diagonal grids when large CFL numbers are used, while the opposite trend is observed for parallel grids. This may be explained by the development of the producing water fractional flow function in the wells after water breakthrough. Consider front tracking on one row (or column) of the grid near the boundary. When large time steps are used on a diagonal grid, the oil bank trapped close to the boundary (see Figure 7) will give rise to a flow of oil into the grid block that is connected to the well. Therefore the water fractional flow will decrease during the step and the water production is under–estimated. A similar argument yields that the calculated water production is too high for parallel grids.

The material balance error in Table 1 varies significantly with CFL, while the oil remaining in the reservoir is almost insensitive to CFL. This indicates that our method for calculating production is not accurate for large time steps. One problem is that saturations and concentrations between t^n and t^{n+1} are used to find flux functions in the producers, while the dimensional splitting method only defines saturations and concentrations at discrete time steps. We tested another method based on numerical fluxes into well blocks weighted with flux functions at the discrete time levels without success. A potential solution could be to include grid refinement around the producers, as indicated by Bratvedt et al. [1] for the commercial FRONTSIM simulator.

Oil production and water material balance for both types of grids are compared in Figure 8 for CFL $= 2$. Observe that the oil production is somewhat larger for the diagonal grid. A more detailed analysis shows that about 3/4 of the grid sensitivity is due to the material balance errors, the rest is due to the difference in oil remaining in the reservoir. The material balance error is negligible before water breakthrough (which occurs after about 0.55 pore volumes injected). This means that large CFL numbers may be used before breakthrough. However, the majority of the CPU time for the current problem is spend on the post breakthrough period.

3.4. **Miscible Flooding with Adverse Mobility Ratio.** Our last example is related to injection of pure water after injection of a polymer slug. The problem is simplified by

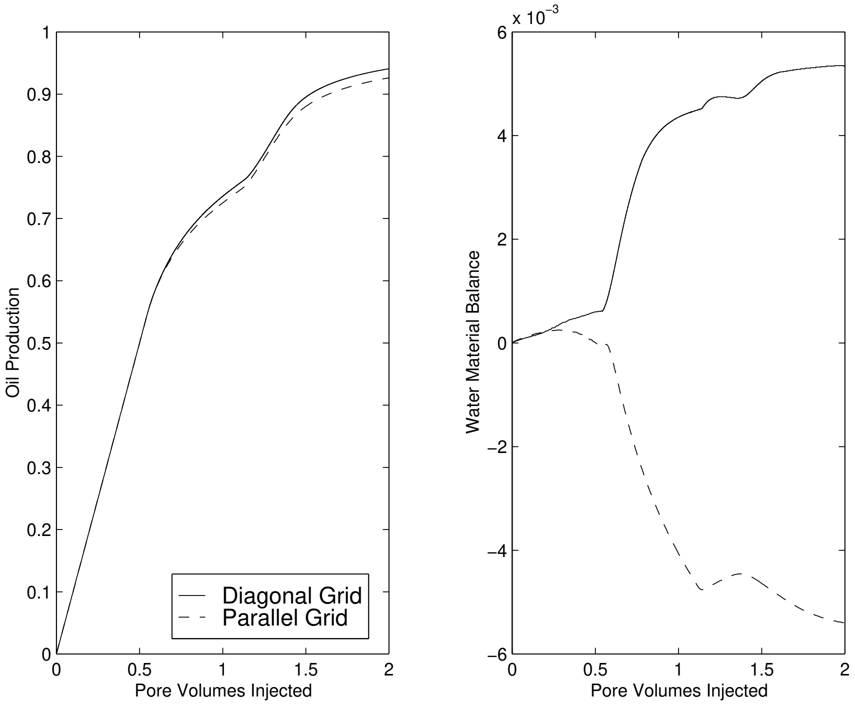


FIGURE 8. Oil production (left) and water material balance (right) for the diagonal and parallel grid are plotted versus the number of pore volumes injected. Both solutions were calculated by a CFL number of 2.

considering injection of pure water into a reservoir that is filled initially with water with a constant polymer concentration. The numerical difficulties created by this example are qualitatively similar to following the injection of polymer in the previous example by pure water.

The conservation equation of water is trivial in this case since the water saturation is unity for all times. A displacement with an adverse mobility ratio $M > 1$ is considered. The mobility ratio is defined as the ratio between the viscosity of the displaced fluid to the viscosity of the injected fluid. When the mobility ratio is larger than one, the displacement is unstable and the injected water tends to "finger" through the polymer. The mobility ratio is set to 10, the initial polymer concentration is unity, and the adsorption is neglected in this example. The total mobility is then $\lambda(c) = 1/(1 + 9c)$. Note that the conservation equation for polymer is linear for a miscible displacement processes. Front tracking has its worst performance for linear flux functions, for which there are no self–sharpening effects to counteract the numerical diffusion introduced by the projection operator; see [14].

Figure 9 shows the front tracking solutions for a diagonal and a parallel grid. In all numerical results for the five–spot, we interpolate results from the parallel grid onto a diagonal grid for plotting. Note that the two numerical solutions are totally different. The concentration calculated by the parallel grid shows an unstable finger of pure water from the injector towards the producer. The solution on the diagonal grid contains fingers

TABLE 2. The discrete L^1 norms between the numerical solutions on a diagonal and parallel grid after 0.4 pore volumes injected for different spatial discretizations. All simulations used CFL = 1 and velocity update at each time step.

$N_x = N_y$	17	33	65	129	257
L^1 difference	0.0273	0.0347	0.0571	0.0850	0.1062

of water flowing towards the injectors in addition to the finger towards the producer. The unphysical behaviour on the diagonal grid was not eliminated by using a finer grid. Table 2 shows that the discrete L^1 difference between the numerical solutions on a diagonal and parallel grid after 0.4 pore volumes injected increases when the spatial discretization parameters decrease. The increase in L^1 difference is caused by earlier breakthrough on the parallel grid and larger unphysical fingers on the diagonal grid as the grid is refined. Also, different discretizations of the pressure equation gave qualitatively the same displacement mechanisms. The numerical methods for the pressure equations that have been tested include different mobility evaluations in the Yanosik and McCracken scheme and the Shubin and Bell [21] scheme.

Two finite–difference methods have been applied to the same problem. Assuming a one-dimensional numbering of the grid blocks, the discretization of the pressure equation reads

$$\sum_j t_{i,j}^n (p_i^n - p_j^n) = q_i^n,$$

where the transmissibilities $t_{i,j}^n$ depend on permeability, total mobility, and the spatial discretization parameters. For symmetric discretizations of the pressure equation, $t_{i,j}^n = t_{j,i}^n$, we have updated the hyperbolic equations by

$$\frac{s_i^{n+1} - s_i^n}{\Delta t^n} + \sum_j f_{i,j}^n t_{i,j}^n (p_i^n - p_j^n) = f_i^n q_i^n,$$

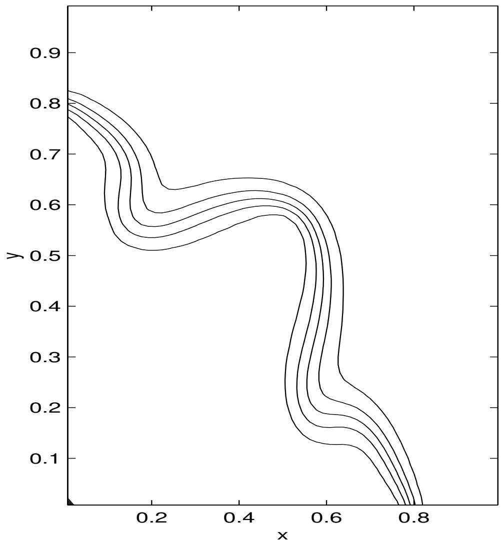
$$\frac{(sc + (a(c))_i^{n+1} - (sc + a(c))_i^n}{\Delta t^n} + \sum_j (cf)_{i,j}^n t_{i,j}^n (p_i^n - p_j^n) = (cf)_i^n q_i^n.$$

The fractional flow of water is given by an upwind method

$$f_{i,j}^n = \begin{cases} f(s_i^n, c_i^n) & \text{for } p_i^n > p_j^n, \\ f(s_j^n, c_j^n) & \text{for } p_j^n > p_i^n, \end{cases}$$

and a corresponding definition is used for the polymer flux.

An upwind scheme based on the standard five–point approximation of the Laplacian was used to calculate the concentration contours in Figure 10. Note that the qualitative behaviour of the five–point approximations is similar to the front tracking solution. However, the finite–difference solutions are more smeared than the corresponding front



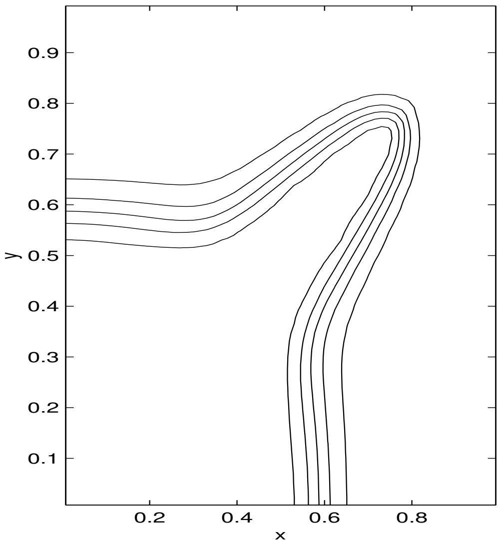


FIGURE 9. Concentration contours for the front tracking method after 0.4 pore volumes injected on diagonal (left) and parallel (right) grids with $N_x = N_y = 65$ and CFL $= 1$. The velocity field was updated at each time step.

tracking solution. Also, water breakthrough on the parallel grid is earlier for the five–point finite–difference scheme than for front tracking. The difference between the times to breakthrough is partly due to the smaller time steps in the finite–difference scheme.

An upwind method based on the nine–point approximation of the pressure equation was also applied to the same problem, and concentration contours for diagonal and parallel grids are shown in Figure 11. As expected, grid orientation effects are now small compared with the front tracking method and the five–point finite–difference method.

The grid sensitivity of the front tracking method is due to the dimensional splitting. For the diagonal grid, the fluid must first move in the x– or y–direction, i.e., in a direction making an angle of 45 degrees with the main flow direction. This initial movement of the injected fluid triggers unphysical fingers on the diagonal grid. Shubin and Bell [21] studied the effect of the numerical diffusion for finite–difference schemes applied to miscible displacement. They concluded that numerical methods with anisotropy in the numerical diffusion are plagued by grid sensitivity. Our results for the front tracking method based on dimensional splitting is consistent with this observation. The numerical diffusion for this method is introduced by the projection step after the one-dimensional solutions, and the diffusion is along the directions introduced by the dimensional splitting.

The FRONTSIM method by Bratvedt et al. [1] is quite similar to the front tracking method described in Section 2 in the case of miscible displacement. FRONTSIM uses a piecewise constant velocity field and a local grid refinement close to the wells. Our results indicate that this method is highly grid sensitive for adverse mobility flooding. The random choice method developed by Sethian et al. [20] is also likely to be sensitive to the orientation of the grid for these displacement processes.

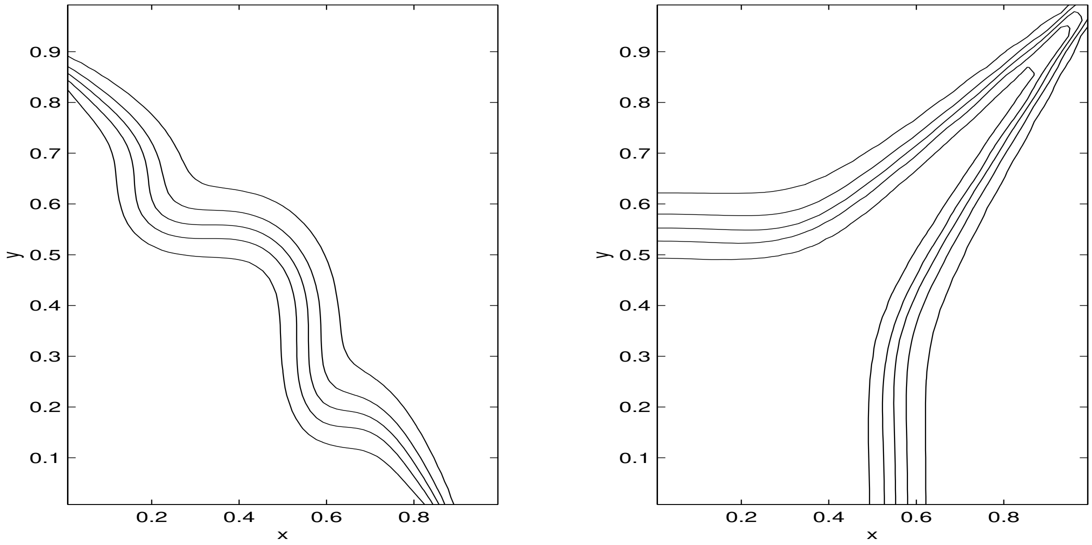


FIGURE 10. Concentration contours after 0.4 pore volumes injected for the upwind scheme based on a five–point discretization on diagonal (left) and parallel (right) grids with $N_x = N_y = 65$ and CFL $= 0.5$. The velocity field was updated at each time step.

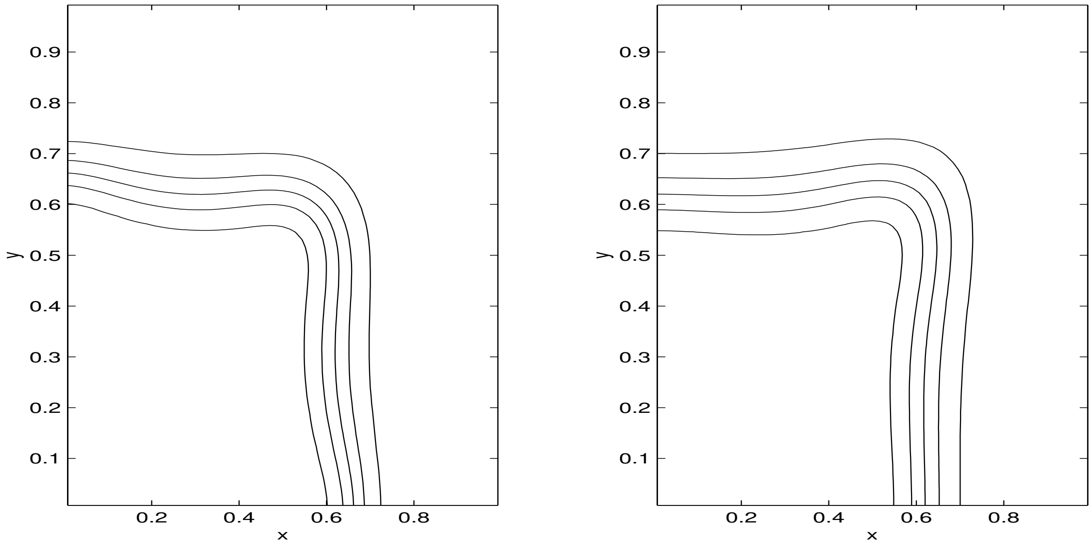


FIGURE 11. Concentration contours after 0.4 pore volumes injected for the upwind scheme based on a nine–point discretization on diagonal (left) and parallel (right) grids with $N_x = N_y = 65$ and CFL $= 0.5$. The velocity field was updated at each time step.

4. Concluding Remarks

The front tracking method presented in this paper is computationally a very efficient method to solve the polymer system, especially in one spatial dimension and for multidimensional examples with non-singular velocities. Numerical experiments in one dimension show that the large initial total variation for the front tracking method, as pointed out by Tveito and Winther [24], does not cause convergence problems for this method. The smearing of sharp fronts by the front tracking method is significantly reduced compared with standard finite–difference schemes.

In the dimensional splitting method for multidimensional problems there are two sources of errors – splitting errors and numerical diffusion caused by projections. At CFL numbers around unity, the latter is dominant. Increasing the CFL number to well above unity reduces the diffusion but does not increase the splitting errors significantly. Typically, we observed feasible CFL numbers in the range 10–20. The method is therefore very efficient compared to finite–difference methods. The performance is similar to the performance observed earlier for scalar equations [15, 14] and better than for the Euler and the shallow water equations [6, 4].

Unfortunately, the above situation changes when sources/sinks are included in the velocity field, as is the case in reservoir simulation. Numerical examples show that the method propagates water fronts very accurately before water breakthrough for fairly large CFL numbers. After breakthrough in a well, substantial material balance errors are introduced if too large time steps are used, at least when measured by our current method for calculating production in wells. Acceptable material balance errors are obtained by using a CFL number that is only moderately larger than unity. The CPU efficiency of the method for reservoir simulation applications is reduced by this constraint. Another drawback of the method is the grid sensitivity for unstable displacements. Anisotropy introduced by the dimensional splitting may lead to unphysical fingers in miscible displacements with unfavourable mobility ratios. Numerical experiments show that the grid sensitivity is accentuated as the grid is refined.

References

[1] F. Bratvedt, K. Bratvedt, C.F. Buchholz, T. Gimse, H. Holden, L. Holden, and N.H. Risebro. Frontline and Frontsim: Two full scale, two–phase, black oil reservoir simulators based on front tracking. *Surv. Math. Ind.*, pages 185–215, 1993.

[2] C. Dafermos. Polygonal approximations of solutions of the initial value problem for a conservation law. *J. Math. Anal. Appl.*, 38:33–41, 1972.

[3] J. Glimm. Solutions in the large for nonlinear hyperbolic systems of equations. *Comm. Pure Appl. Math.*, 18:697–715, 1965.

[4] R. Holdahl, H. Holden, and K.-A. Lie. Unconditionally stable splitting methods for the shallow water equations. *BIT*, to appear.

[5] H. Holden, L. Holden, and R. Høegh-Krohn. A numerical method for first order nonlinear scalar conservation laws in one-dimension. *Comput. Math. Applic.*, 15(6–8):595–602, 1988.

[6] H. Holden, K.-A. Lie, and N.H. Risebro. An unconditionally stable method for the Euler equations. *J. Comp. Phys.*, to appear.

[7] K. Holing, J. Alvestad, and J.A. Trangenstein. The use of second-order Godunov methods for simulating EOR processes in realistic reservoir models. In D. Guérillot and O. Guillon, editors, *2nd European Conference on the Mathematics of Oil Recovery*, pages 101–111, 1990.

[8] E. Isaacson. The global solution of a Riemann problem for a non–strictly hyperbolic system of conservation laws arising in enhanced oil recovery. Preprint, The Rockefeller University, 1981?

[9] E. Isaacson and B. Temple. Analysis of a singular hyperbolic system of conservation laws. *J. Differential Equations*, 65:250–268, 1986.

[10] G.-S. Jiang and E. Tadmor. Non-oscillatory central schemes for multidimensional hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 19:1892–1917, 1998.

[11] T. Johansen, A. Tveito, and R. Winther. A Riemann solver for a two–phase multicomponent process. *SIAM J. Sci. Statist. Comp.*, 19:541–566, 1989.

[12] T. Johansen and R. Winther. The solution of the Riemann problem for a hyperbolic system of conservation laws modelling polymer flooding. *SIAM J. Math. Anal.*, 19:541–566, 1988.

[13] C. Klingenberg and N.H. Risebro. Stability of a resonant system of conservation laws modeling polymer flow with gravitation. Preprint, 1999.

[14] K.-A. Lie. A dimensional splitting method for nonlinear equations with variable coefficients. Preprint Mathematics no. 17/1997, Norwegian University of Science and Technology, 1997.

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

[16] T. Potempa. Finite element methods for convection dominated transport problems. Ph.D. thesis, Rice University, 1982.

[17] N.H. Risebro. A front-tracking alternative to the random choice method. *Proc. Amer. Math. Soc.*, 117(4):1125–1139, 1993.

[18] N.H. Risebro and A. Tveito. A front tracking method for conservation laws in one dimension. *J. Comp. Phys.*, 101(1):130–139, 1992.

[19] 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.

[20] J.A. Sethian, A.J. Chorin, and P. Concus. Numerical solution of the Buckley–Leverett equations. In *SPE Reservoir Simulation Symposium*, pages 197–204, 1983. SPE 12254.

[21] G.R. Shubin and J.B. Bell. An analysis of the grid orientation effect in numerical simulation of miscible displacement. *Comp. Meth. Appl. Mech. Eng.*, 47:47–71, 1984.

[22] B. Temple. Global solution of the Cauchy problem for a class of 2×2 nonstrictly hyperbolic conservation laws. *Adv. in Appl. Math.*, 3:335–375, 1982.

[23] A. Tveito and R. Winther. Existence, uniqueness and continuous dependence for a system of conservation laws modelling polymer flooding. *SIAM J. Math. Anal.*, 22:905–933, 1991.

[24] A. Tveito and R. Winther. The solution of the nonstrictly hyperbolic conservation law may be hard to compute. *SIAM J. Sci. Comp.*, 16:320–329, 1995.

[25] J.L. Yanosik and T.A. McCracken. A nine point finite difference simulator for realistic predictions of adverse mobility ratio displacements. *SPE Journal*, pages 252–262, August 1979.

[26] B. Wendroff. An analysis of front tracking for chromatography. *Acta Appl. Math.*, 30(3):265–285, 1993.

(Vidar Haugse)
Department of Petroleum Engineering and Applied Geophysics
Norwegian University of Science and Technology
S.P. Andersens vei 15A
N-7491 Trondheim, Norway
E-mail address: vidarh@ipt.ntnu.no
URL: http://www.ipt.ntnu.no/~vidarh/

(Kenneth 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/

(Knut–Andreas 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/

(Jostein R. Natvig)
Department of Mathematical Sciences
Norwegian University of Science and Technology
N-7491 Trondheim, Norway