OPERATOR SPLITTING METHODS FOR SYSTEMS OF CONVECTION-DIFFUSION EQUATIONS: NONLINEAR ERROR MECHANISMS AND CORRECTION STRATEGIES

K. HVISTENDAHL KARLSEN[a], K.–A. LIE[b,c],
J. R. NATVIG[c], H. F. NORDHAUG[a], AND H. K. DAHLE[a]

ABSTRACT. Many numerical methods for (one-dimensional) systems of convection-diffusion equations are based upon an operator splitting formulation, where convective and diffusive forces are accounted for in separate substeps. We describe the nonlinear mechanism of the splitting error in such numerical methods, a mechanism that is intimately linked to the local linearizations introduced implicitly in the (hyperbolic) convection steps by the use of an entropy condition. For convection-dominated flows, we demonstrate that operator splitting methods typically generate a numerical widening of viscous fronts, unless the splitting step is of the same magnitude as the diffusion scale. To compensate for the potentially damaging splitting error, we propose a corrected operator splitting (COS) method for general systems of convection-diffusion equations with the ability of correctly resolving the nonlinear balance between the convective and diffusive forces. In particular, COS produces viscous shocks with correct structure also when the splitting step is large. A front tracking method for systems of conservation laws, which in turn relies heavily on a Riemann solver, constitutes an important part of our COS strategy. The proposed COS method is successfully applied to a system modelling two-phase, multicomponent flow in porous media and a triangular system modelling three-phase flow.

1. INTRODUCTION

Mathematical models for fluid flow often involve systems of convection-diffusion equations as a main ingredient. When a mathematical model is used for qualitative or quantitative studies, approximate numerical solutions must be constructed for the nonlinear system. An important design principle for many successful numerical methods for convection-diffusion equations is *operator splitting* (OS). That is, one splits the time evolution into partial steps to separate the effects of convection and diffusion. In particular, OS methods are often used to solve convection-diffusion problems that are of convection dominated nature, see [12] (and the references therein).

The motivation for operator splitting methods lies in that it is easy to combine modern methods developed within the hyperbolic community for tracking discontinuous solutions with efficient methods (e.g., multigrid) for solving implicit discretizations of the parabolic

Date: January 13, 2000.
Key words and phrases. systems of convection-diffusion equations, numerical methods, front tracking, finite difference method, operator splitting, splitting error, two-phase multi-component flow, three-phase flow.

[a]Department of Mathematics, University of Bergen, Johs. Brunsgt. 12, N–5008 Bergen, Norway.
E-mail: kenneth.karlsen@mi.uib.no, hans.nordhaug@mi.uib.no, helge.dahle@mi.uib.no.
[b]Department of Informatics, University of Oslo, P.O. Box 1080, Blindern, N-0316 Oslo, Norway.
E-mail: kalie@ifi.uio.no.
[c]SINTEF Applied Mathematics (Numerical Simulation), P.O. Box 124 Blindern, N-0314 Oslo, Norway.
E-mail: Jostein.R.Natvig@math.sintef.no, Knut-Andreas.Lie@math.sintef.no.

diffusive step, thus giving a powerful and efficient numerical method designed for resolving sharp gradients. This can be done in a stepwise (plug-and-play) manner, starting with a simple solver for each subproblem and then replacing each solver independently of the other by a more advanced solver until a suitable level of sophistication is reached.

The obvious disadvantage of operator splitting methods is the temporal splitting errors. Consider a scalar, convection-dominated problem $u_t + f(u)_x = \varepsilon u_{xx}$. If the equation possesses a viscous shock profile, this profile will appear on a spatial scale of order ε and move on a time scale of order $|f'(u)|$. Recent studies [2, 20, 21, 23, 24] have shown that unless the splitting step is of order ε, the temporal splitting error in OS methods can be significant in regions containing viscous shocks. The resulting incorrect balance between convective and diffusive forces appears as too wide shock layers in the numerical solution. Thus, to resolve viscous shock profiles correctly, one must resort to very small splitting steps, thereby imposing a time step restriction that is not present in the underlying numerical methods for the convective and the diffusive step. In fact, the splitting step needed to resolve the shock layers correctly may be much less than the one needed to resolve the interaction of viscous waves.

Small splitting steps should be avoided (if possible) for two reasons: computational efficiency and spatial accuracy. Increasing the number of splitting steps usually means increasing the runtime. Moreover, for numerical methods having stability restrictions, the highest *spatial* accuracy is often obtained when the time step is close to the stability limit. For these two reasons, one should try to pick the splitting step as large as possible. To reduce the influence of temporal splitting errors in OS methods and allow for the use of large splitting steps, the *corrected operator splitting* (COS) method was introduced by Karlsen and Risebro [23]. The COS method was further developed and successfully applied by Karlsen *et al.* in a series of papers [20, 21, 22, 13]. The forerunner for the scalar COS method was the modified method of characteristics for nonlinear scalar parabolic problems introduced by Espedal and Ewing [11] and Dahle [2], and further developed and analysed by Dahle, Espedal, and their collaborators [2, 3, 4, 5, 6] in the context of reservoir simulation. The relation between the modified method of characteristics and COS is discussed in the lecture notes [12].

The main idea behind the scalar COS method is to take into account the unphysical entropy loss (due to Oleinik's convexification) produced by the hyperbolic solver in the convective step. The COS approach uses the wave structure from the convective step to identify where the (nonlinear) splitting error occurs. This potential error is then compensated for in the diffusive step or in a separate correction step.

The purpose of this paper is to propose a working COS algorithm for systems of convection-diffusion equations that generalizes the scalar COS algorithm [23]. To this end, we need to derive a thorough understanding of the nonlinear mechanisms behind the viscous splitting error, which is well understood in the scalar case. This is done in Sections 2.1 to 2.3, where we introduce the viscous splitting in more detail, discuss the nonlinear mechanisms behind the temporal splitting error, and introduce a general correction strategy. Then in Section 2.4, we suggest particular numerical methods for solving the split problems introduced by the COS method and describe its numerical realisation in detail. In Section 3 we demonstrate the novel COS algorithm by applying it to two 2×2 systems of convection-diffusion equations modelling one-dimensional flow in porous media. We present a two-dimensional extension of the COS method based on dimensional splitting in Section 4. Finally, we make some concluding remarks in Section 5.

2. Operator Splitting Methods for General Systems

To describe of our ideas in more detail, we consider the Cauchy problem for $\ell \times \ell$ ($\ell \geq 1$) systems of convection-diffusion equations

$$\partial_t U + \partial_x F(U) = D\,\partial_x^2 U, \qquad U(x,0) = U_0(x) \tag{1}$$

where $x \in \mathbb{R}$ and $t > 0$. Here $U = (u_1, \dots, u_\ell)^{\mathrm{T}}$ is the unknown state vector, $F(U) = (f_1(U), \dots, f_\ell(U))^{\mathrm{T}}$ is a vector-valued function of class C^2, and $D = \mathrm{diag}(\varepsilon_1, \dots, \varepsilon_\ell) > 0$ is a constant diagonal matrix. The linear diffusion operator is chosen for simplicity; all principles introduces below also apply to more complicated nonlinear (degenerate) diffusion operators. We always assume that the initial function $U_0(x)$ is of bounded total variation, i.e., $U_0 \in BV$. For a class of systems of the type (1), existence (and uniqueness) of classical solutions was established by Hoff and Smoller [17] using a finite difference scheme.

2.1. **Semi-Discrete OS.** Let $\mathcal{S}_t$ denote the solution operator taking the initial data $V_0(x)$ to a weak solution at time t of the purely hyperbolic problem

$$\partial_t V + \partial_x F(V) = 0, \qquad V(x,t) = V_0(x), \tag{2}$$

i.e., we write $v(x,t) = \mathcal{S}_t v_0(x)$ for this weak solution. For strictly hyperbolic $\ell \times \ell$ systems with initial data having small total variation, global existence of weak solutions of was proved by Glimm [15]. For stability and uniqueness of weak solutions, we refer to the recent paper by Bressan, Liu, and Yang [1].

Next, let $\mathcal{H}_t$ denote the operator taking the initial data $W_0(x)$ to a weak solution at time t of the purely parabolic problem

$$\partial_t W = D\partial_x^2 W, \qquad W(x,t) = W_0(x), \tag{3}$$

i.e., we write $W(x,t) = \mathcal{H}_t W_0(x)$ for this solution. For initial data of bounded total variation, the weak solution of (3) is a classical solution for $t > 0$, with the initial data U_0 taken in the strong L^1 sense on compact sets. This can be easily seen from the representation formula for the solution of the linear heat equation.

In what follows, we consider a fixed final computing time $T > 0$. For simplicity we also choose a fixed splitting step $\Delta t > 0$ and an integer N such that $N\Delta t = T$. Then we define the semi-discrete OS algorithm by

$$U_{\Delta t}(\cdot, n\Delta t) := \left[\mathcal{H}_{\Delta t} \circ \mathcal{S}_{\Delta t}\right]^n U_0(\cdot), \qquad n = 0, \dots, N. \tag{4}$$

In the scalar case, it can be proved that $U_{\Delta t}$ converges in L^1 on compact sets to unique classical solution U of (1) as the splitting step Δt tends to zero. The convergence proof is based on a uniform BV (space) estimate for the splitting approximation and Helly's theorem, see Karlsen and Risebro [24] for details. In general, it is not possible to prove BV estimates for systems and thus the Helly framework cannot be used to prove convergence of operator splitting methods. However, it is possible to prove a Lax–Wendroff type theorem for splitting methods for systems of parabolic equations:

Theorem 2.1. *If the splitting method* (4) *converges to a limit $U(\cdot,t) \in BV$, then this limit is the unique classical solution of* (1).

The proof goes along the lines of [18]. The theorem shows that (4) can indeed be used as a basis for constructing numerical solution algorithms for (1).

In applications, the exact solution operators $\mathcal{S}_t$ and $\mathcal{H}_t$ in (4) are replaced by numerical methods. We will use front tracking as defined by Risebro [27, 28, 29] as an approximate solution operator for the hyperbolic part. For the parabolic part, we here use a simple explicit central difference method. The methods will be introduced in more detail in the next section. Meanwhile, let us stress that the method for the diffusion part is deliberately chosen as simple as possible to focus on the main ideas of our COS strategy and generally should be replaced by a more sophisticated method. We mention that Dawson, Wheeler, and collaborators [7, 8, 9, 10, 32] are using operator splitting algorithms similar to (4). In their splitting algorithm, the hyperbolic problem (2) is solved by $M \geq 1$ local time steps (for each splitting step) with an explicit Godunov type method, while the diffusion equation (3) is solved implicitly.

2.2. **Nonlinear Error Mechanisms.** In the introduction we stated that OS approximations can be too diffusive near viscous shocks when the splitting step Δt is large. Karlsen and Risebro [23] point out that this splitting error is simply a manifestation of the entropy condition that is imposed in the hyperbolic convection step. Let us consider the *scalar* case. The entropy condition introduces a local linearization of $f(\cdot)$ once a shock is formed in the convection step and this linearization represents the entropy loss associated with the formation of a shock in the hyperbolic solution. Thus the evolution of the *hyperbolic* solution is governed locally by some convex/concave envelope f_c of f between the left and right shock values. A similar linearization can be introduced locally for the *parabolic* problem; that is, the flux function f can be decomposed into a convective part f_c and a self-sharpening part $f - f_c$ that tends to counteract the diffusive forces. Loosely speaking, we say that f_c governs the local translation and $f - f_c$ the shape (or structure) of the viscous front. In the OS algorithm, the local residual flux $f - f_c$ is disregarded in the hyperbolic step and the corresponding self-sharpening effects are therefore not taken into account in the splitting, resulting in a splitting error. Figure 1 gives an illustration of f, f_c, and the residual flux $f_{\mathrm{res}} := f - f_c$ in the scalar case.

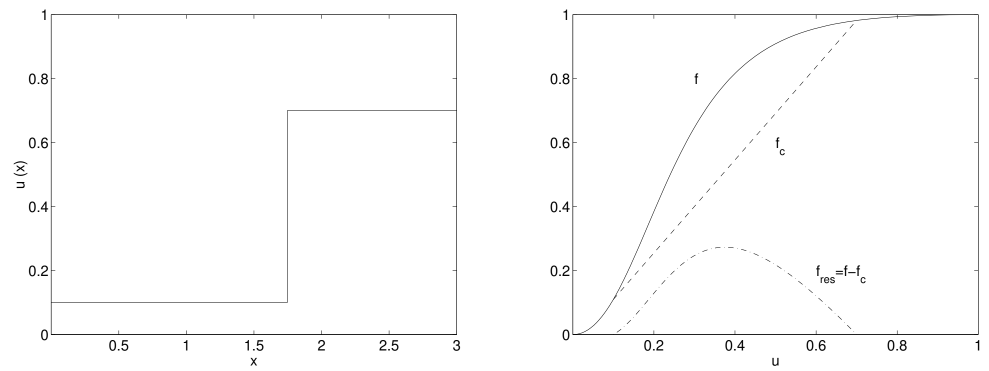

FIGURE 1. (Left) A single shock solution from a convection step. (Right) The corresponding residual flux function; flux function f (solid), convex envelope f_c, i.e., local linearization (dash), and residual flux f_{res} (dash-dot).

For a general system, the error mechanism is quite similar. To study it, we consider the propagation of a single viscous shock. Assume that the splitting step is sufficiently large so that a shock has developed in the hyperbolic substep (2), i.e., the solution $V(\cdot, t = \bar{t})$ consists of a single discontinuity at $x = \bar{x}$ with left and right shock values $V^l = (v_1^l, \ldots, v_1^l)^{\mathrm{T}}$ and

$V^r = (v_1^r, \dots, v_1^r)^{\mathrm{T}}$. Then the behaviour (forward and backward in time) of $V(x,t)$ locally around $(\bar{x}, \bar{t})$ is governed by the linearized hyperbolic problem

$$\partial_t V + \partial_x(\bar{\sigma} V) = 0, \qquad V(x, \bar{t}) = \begin{cases} V^l, & \text{for } x < \bar{x}, \\ V^r, & \text{for } x > \bar{x}, \end{cases} \tag{5}$$

where $\bar{\sigma}$ is the Rankine–Hugoniot shock speed satisfying

$$F(V^l) - F(V^r) = \bar{\sigma}(V^l - V^r).$$

We *claim* that a large part of the splitting error occurring locally around $(\bar{x}, \bar{t})$ in the standard OS algorithm can be understood in terms of the difference between the nonlinear system in (1) and the linearized system in (5) with right-hand side $D\partial_x^2 V$, or in other words, in terms of the difference $\partial_x(F(U) - \bar{\sigma}U)$. In (1), the diffusion caused by the second order operator is perfectly balanced by the self-sharpening effects due to the nonlinearity in the convective operator. In the OS strategy, this self-sharpening disappears once a shock develops because $F(U)$ is in effect replaced by $\bar{\sigma}U$ locally. Thus, one step in OS effectively amounts to solving $\partial_t U + \partial_x(\bar{\sigma}U) = D\partial_x^2 U$ and not (1).

2.3. **A Novel COS Strategy.** To compensate for the loss of self-sharpening effects, the *scalar* COS approach proposes to include the residual flux F_{res} in the diffusion step of the splitting. The COS method therefore replaces the purely parabolic split problem (3) by

$$\partial_t W + \partial_x F_{\text{res}}(W; x) = D\partial_x^2 W, \qquad W(x, 0) = W_0(x). \tag{6}$$

As an alternative, the residual flux can be included in a separate correction step, $\partial_t V + \partial_x F_{\text{res}}(V; x) = 0$, see [23, 21] for more details. Letting $\mathcal{P}_t$ denote the solution operator associated with (6), the COS solution may then be defined as

$$U_{\Delta t}(\cdot, n\Delta t) := \left[\mathcal{P}_{\Delta t} \circ \mathcal{S}_{\Delta t}\right]^n U_0(\cdot). \tag{7}$$

What we have done so far might seem a bit peculiar. We have taken the convection-diffusion equation (1) and replaced it by a hyperbolic equation (2) and another convection-diffusion equation (6), where the flux term in (6) is seemingly more complicated than the one in (1). However, we see that while F contains convective *and* self-sharpening effects, F_{res} only contributes self-sharpening effects. Thus, viscous shock fronts are moved to the correct location in the convective step and given a correct shape in the diffusive step. The solution process has also been simplified from a numerical point of view, i.e., with a fully discrete version of (7). Parabolic equations of the kind (1) and (6) will typically be solved by some implicit scheme, involving iterative solution of nonlinear systems of equations. The hyperbolic step can therefore be seen as some kind of preconditioner or a means for providing a good initial guess for the iteration, and the iteration process will converge faster for (6) than for (1). If the method used to solve the hyperbolic step is fast compared with the nonlinear iteration, we gain something in terms of efficiency, see, e.g., [4, 5, 11]. In the next subsection we introduce one such method, which employs a very fast, unconditionally stable, front-tracking method for the convective step.

When applied to systems of parabolic equations, the correction algorithm needs to be reformulated, since one cannot simply write down the solution of the hyperbolic step in terms of convex/concave envelopes. Instead, we identify the following term

$$\partial_x F_{\text{res}}(U) = \partial_x\big(F(U) - \bar{\sigma}U\big), \tag{8}$$

for each discontinuity in the solution from the hyperbolic step. Then, the parabolic subproblem (3) is modified locally by adding $F_{\text{res}}(U)$, giving the new split problem (6). By integrating (8) with respect to x, we get the *residual flux*

$$F_{\text{res}}(U) = \big(F(U) - F(V^l)\big) - \bar{\sigma}\big(U - V^l\big), \tag{9}$$

where we have chosen the constant of integration such that

$$F_{\text{res}}(V^l) = F_{\text{res}}(V^r) \equiv 0.$$

2.4. **A Fully Discrete (C)OS Method.** The operator splitting methods introduced above result in two different subproblems that each must be solved numerically. Therefore, before we describe the OS and COS approach in more detail, we introduce numerical methods for solving the subproblems.

Convection solver. In this section we describe the front tracking method [27, 28, 29] for solving systems of conservation laws (2)

$$\partial_t V + \partial_x F(V) = 0, \quad V(x,0) = V_0(x).$$

The initial function $V_0(x)$ is assumed to be of bounded variation. Front tracking is an algorithm for computing a piecewise constant approximation to $V(x,t)$. First, V_0 is approximated by a step function so that a Riemann problem can be associated with each jump in the approximate initial data. The solution of each Riemann problem is approximated by a step function. This is most accurately achieved by using the Lax construction for the exact Riemann solution, utilizing the local system of coordinates formed by the wave curves around two constant states in state space. The Riemann solution consists of a set of constant states connected by simple waves. In the front tracking approximation, rarefaction waves are approximated by step functions sampled along the wave curves (according to a pre-set, user-defined parameter δ), while the rest of the Riemann solution is left intact. This way, each Riemann problem produces a sequence of jump discontinuities (fronts) that travel with a finite wave speed. The Riemann solution is represented by a list of fronts sorted according to increasing waves speed.

A global solution (in x) is formed by connecting the local Riemann solutions and consists of constant states separated by space-time rays, i.e., a list of fronts sorted from left to right. There will be a first time at which two or more space-time rays intersect, i.e., two or more fronts collide. This collision defines a new Riemann problem with states given by the rightmost and leftmost of the colliding fronts. The colliding fronts are taken out of the solution. Then the Riemann problem is solved and the emerging fronts are inserted into the front list. This way, the algorithm proceeds from collision to collision. Notice that no computations are necessary between collision times. To reduce the number of wave interactions, it is customary to perform some data reduction, i.e., remove weak waves, see [29]. The numerical method is unconditionally stable and very fast.

Diffusion solver. The parabolic step is a Cauchy problem of the form

$$\partial_t W + \partial_x G(W) = \partial_x^2 D(W), \quad W(x,0) = W_0(x). \tag{10}$$

The initial function $W_0(x)$ is assumed to be of bounded variation. To approximate the solution $W(x,t)$, we introduce a mesh in the (x,t) plane where the spatial grid points are denoted by x_j and the time levels by t_n. We denote the spacing in the x and t variables by Δx and τ, respectively; i.e., $(x_j, t_n) = (j\Delta x, n\tau)$. The value of the difference approximation at (x_j, t_n)

is denoted by W_j^n. To solve this system, one can for instance use the explicit, central finite difference method

$$\frac{W_j^{n+1}-W_j^n}{\tau}-\frac{G(W_{j+1}^n)-G(W_{j-1}^n)}{2\Delta x}=\varepsilon\frac{D(W_{j+1}^n)-2D(W_j^n)+D(W_{j-1}^n)}{(\Delta x)^2}. \tag{11}$$

This scheme is stable provided the discretization parameters τ and Δx satisfy the following conditions

$$\tau \leq 0.5\Delta x^2/\varepsilon, \qquad \Delta x \max|\lambda_G| \leq 2\varepsilon,$$

where λ_G denotes the eigenvalues of G'. In the case when $D(W)=W$, i.e., linear diffusion, convergence and error estimates for this scheme are shown in [17].

The stability conditions above may put severe restrictions on the discretization parameters, especially on Δx for small values of ε. The second condition on Δx is removed by using an upwind discretization of the flux G. However, the most efficient method would generally be an implicit discretization combined with an efficient (non)linear solver. In Sections 3 and 4, however, we have chosen the explicit scheme with central and upwind discretization, respectively, of the flux terms.

The splitting method. Given numerical methods for the two substeps, we can now describe the (C)OS method for the convection-diffusion problem (1). The construction of the residual flux F_{res} is described in detail later on.

Letting $\Delta x > 0$ denotes the grid spacing, we introduce the grid $\{x_j = j\Delta x\}_{j\in\mathbb{Z}}$. Our approximate solutions will be piecewise constant functions with respect to the grid cells $\{I_j = [x_j, x_{j+1})\}_{j\in\mathbb{Z}}$. Therefore we need the scalar projection operator π defined by

$$\pi z(x) = \frac{1}{\Delta x}\int_{I_j} z(\xi)\;dx, \qquad \text{for all } x \in I_j, \tag{12}$$

for any scalar function $z(x) \in BV$, and the operator Π defined by $\Pi Z = (\pi z_1, \dots, \pi z_\ell)^{\mathrm{T}}$, for any vector-valued function $Z = (z_1, \dots, z_\ell)^{\mathrm{T}} \in BV$. We next consider a fixed final computing time $T > 0$, and choose a splitting step $\Delta t > 0$ and an integer N such that $N\Delta t = T$. Using Δ as a short-hand notation for the discretization parameters $(\Delta x, \Delta t, \delta)$, we define our piecewise constant COS approximation $U_\Delta : \mathbb{R}\times[0,T] \to \mathbb{R}$ by

$$U_\Delta(x,t) := U^n(x), \qquad (x,t) \in \mathbb{R}\times((n-1)\Delta t, n\Delta t], \quad n = 1,\dots,N, \tag{13}$$

where $U_\Delta(x,0) := U^0(x)$ and $U^0 := \Pi U_0$. For notational convenience, we have suppressed the dependency on Δ in U^{n+1}. We next explain how to inductively construct the piecewise constant function $U^{n+1}(x)$ from the knowledge of the piecewise constant function $U^n(x)$. The construction consists of two main steps:

Step 1 (Convection): Let $\mathcal{S}_t V_0$ be the front tracking solution of the hyperbolic problem

$$\partial_t V + \partial_x F(V) = 0, \quad V(x,0) = V_0(x).$$

where we have suppressed the dependency on the discretization parameters Δx and δ in $\mathcal{S}_t$. We then define the intermediate solution

$$U^{n+1/2} = \mathcal{S}_{\Delta t} U^n.$$

Step 2 (Diffusion): Introduce a local time step τ satisfying $\tau \leq 0.5\Delta x^2/\varepsilon$. Furthermore, we let N_τ be an integer such that $N_\tau \tau = \Delta t$. Let $\mathcal{P}_t W_0$ be the finite difference solution of

$$\partial_t W + \partial_x F_{\mathrm{res}}(x, W) = D\partial_x^2 W, \quad W(x,0) = \Pi W_0(x),$$

where we have suppressed the dependency on the discretization parameters Δx and τ in $\mathcal{P}_t$. The residual flux $F_{\mathrm{res}}(x,\cdot)$ depends on the hyperbolic solution $U^{n+1/2}$ and its construction is explained in detail below. Finally, the COS solution U^{n+1} is defined as

$$U^{n+1} = \mathcal{P}_{\Delta t} \circ \Pi U^{n+1/2},$$

or alternatively as

$$U^{n+1} = \Big[\mathcal{P}_{\Delta t} \circ \Pi \circ \mathcal{S}_{\Delta t}\Big] U^n.$$

Note that if the residual flux F_{res} is set to zero, the above algorithm reduces to a standard OS algorithm. In Section 3, we demonstrate numerically that the function U_Δ is a good approximation to the exact solution U of the convection-diffusion problem (1). In particular, when Δt is large, the COS method gives significantly more accurate treatment of viscous shocks than the corresponding OS method.

Remark. Although the exact solution of (1) is a smooth function, our approximation $U_\Delta(\cdot,t)$ is merely piecewise constant. Increased accuracy in space can be obtained by replacing $U_\Delta(\cdot,t)$ with a piecewise linear interpolant.

Construction of the residual flux. Given a piecewise constant, front-tracking solution $U^{n+1/2}$, we can now construct the residual flux $F_{\mathrm{res}}(x,\cdot)$ appearing in (6). We assume that the discontinuities of $U^{n+1/2}(x)$ are located at the points $\{x_i\}$. Let $U_i = (u_1^i, \dots, u_\ell^i)^{\mathrm{T}}$ and $U_{i+1} = (u_1^{i+1}, \dots, u_\ell^{i+1})^{\mathrm{T}}$ denote the values of $U^{n+1/2}(x)$ in the intervals $[x_{i-1}, x_i)$ and $[x_i, x_{i+1})$, respectively. Locally around the ith discontinuity emerging from (x_i, t_0) the nonlinear problem (2) is governed by the linearized problem

$$\partial_t V + \partial_x(\sigma_i V) = 0, \qquad V(x_i, t_0) = \begin{cases} U_i, & \text{for } x < x_i, \\ U_{i+1}, & \text{for } x > x_i, \end{cases}$$

where σ_i is the Rankine-Hugoniot shock speed satisfying $F(U_i) - F(U_{i+1}) = \sigma_i(U_i - U_{i+1})$. Motivated by the discussion in Section 2.2, we define the residual flux F_{res}^i associated with the ith discontinuity as

$$F_{\mathrm{res}}^i(U) = \begin{cases} \big(F(U) - F(U_i)\big) - \sigma_i\big(U - U_i\big), & U \in (u_1^i, u_1^{i+1}) \times \cdots \times (u_\ell^i, u_\ell^{i+1}), \\ 0, & \text{otherwise.} \end{cases}$$

Note that $F_{\mathrm{res}}^i(U_i) = F_{\mathrm{res}}^i(U_{i+1}) \equiv 0$.

We note that this is the same residual flux as Karlsen and Risebro [23] presented, since the shock speed σ_i has to fulfil the Rankine-Hugoniot relation. We also note that the constant of integration is of no importance since only derivatives of F_{res} are present in the COS equations.

Although a residual flux term can be identified for every discontinuity in the front tracking solution, they should not be included for discontinuities approximating rarefaction waves or for weak shocks. In practise we therefore only include residual terms for shock waves with strength exceeding a user-defined threshold γ. The process of identifying relevant residuals can be simplified by tagging fronts in the front tracker according to wave type (shock/rarefaction/contact).

Having defined the residual fluxes in state space $(u_1, \ldots, u_\ell)$, we need to specify where to apply them in physical space (i.e., intervals in x). This can be done in several ways. For *explicit* discretizations we apply the following strategy: We observe that in each spatial interval where the solution is monotone in all its components (henceforth called *monotonicity interval*), all residual fluxes are defined on disjoint sets in state space. Therefore, we set the residual flux to zero outside (a subset of) the associated monotonicity interval, i.e.,

$$F_{\text{res}}(x, U) = \sum_i F^i_{\text{res}}(U)\chi_{D_i}(x),$$

where $\chi_I(x)$ denotes the indicator function of the interval $I \subset \mathbb{R}$ and D_i is the (subset of) the monotonicity interval. Using the monotonicity of the solution, we can determine a unique residual flux also in regions of changing monotonicity. Although the monotonicity intervals may change throughout the diffusion step (as the discontinuity is smoothed out), they are always well-defined and easy to compute.

This approach works well for explicit schemes, but it does not apply to implicit discretizations when the discontinuity coincides with a change in monotonicity, i.e., when the left or right state of the discontinuity is a local extremum in one of its components, see e.g., Figure 15. For implicit discretizations we therefore use a much simpler approach where the user prescribes the length of the intervals where the correction is applied. To avoid overlap of residual flux domains, it might be necessary to clip the spatial correction intervals at each endpoint. To this end, one can use monotonicity information and for instance specify that the intervals should not exceed the midpoint between the discontinuities of two consecutive residual fluxes (i.e., not exceed $(x_i + x_{i+1})/2$).

Unfortunately, specifying a reasonable length for the correction intervals must be based on experience. For explicit discretizations, we have observed that the corresponding 'internal boundaries' introduced in the diffusion step may lead to unphysical effects in certain cases if for instance the length of the interval is underestimated by the user; one example is reflections breaking up the monotonicity of the viscous front. For explicit schemes we therefore advocate the approach based upon monotonicity intervals and for implicit schemes the approach based upon specified interval lengths. However, to show that both approaches work for explicit schemes, we construct residual fluxes based upon monotonicity intervals in Section 3.1 and based upon specified length in Section 3.2.

Remark. For scalar conservation laws, $f_{\text{res}}(\cdot, u)$ may be discontinuous for each fixed u, while $f_{\text{res}}(x, \cdot)$ is always continuous for each fixed x, see Figure 1. This is, however, not the case for systems of conservation laws. Each component $f^i_{\text{res}}(x, U)$ of $F_{\text{res}}(x, U)$ may be discontinuous for each fixed U and, more important, is always discontinuous for each fixed x, expect for the trivial case with no shock, see Figure 2 below.

3. Applications

In this section we present numerical realizations of the novel COS strategy for two particular 2×2 systems describing flow in porous media. Simulations for a two-phase, multicomponent model are reported in Section 3.1 and similar experiments for a triangular, three-phase model in Section 3.2.

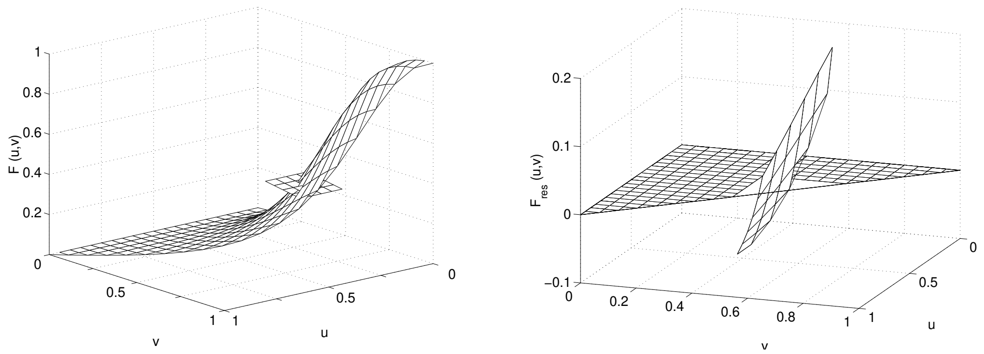

FIGURE 2. (Left) Shock plane and one component of the flux for a system of conservation laws. (Right) The corresponding residual flux component.

3.1. **The Polymer System.** We consider the initial value problem for the following 2×2 system of parabolic equations:

$$
\begin{aligned}
\partial_t s + \partial_x f(s,c) &= \varepsilon \partial_x^2 s \\
\partial_t [sc + a(c)] + \partial_x (c f(s,c)) &= \varepsilon \partial_x^2 [sc + a(c)],
\end{aligned} \tag{14}
$$

where (s, c) is the unknown state vector, $f = f(s,c)$, $a = a(c)$ are given functions

$$
f(s,c) = \frac{s^2}{s^2 + \mu(1+\nu c)(1-s)^2}, \quad a(c) = \frac{0.2c}{1+c}, \tag{15}
$$

and $\varepsilon > 0$ is a small scaling parameter. For all numerical examples in this section, $\mu = 0.5$ and $\nu = 2$.

The system (14) models a polymer process in enhanced oil recovery, see [30] and references therein for details. Existence, uniqueness and stability properties for a smooth solution of the Cauchy problem for (14) have been established by Tveito [30]. The Riemann problem for the corresponding inviscid, nonstrictly hyperbolic system is solved in [19] and a front tracking method is presented in [28]. One special feature of the inviscid system is that the eigenvalues $\lambda_s = f_s$ and $\lambda_c = f/(s + a_c)$ coincide along a curve T in state space.

Introducing the quantity $b = sc + a(c)$, the mathematical model (14) takes the form

$$
\begin{aligned}
\partial_t s + \partial_x f(s,c) &= \varepsilon \partial_x^2 s \\
\partial_t b + \partial_x (c f(s,c)) &= \varepsilon \partial_x^2 b.
\end{aligned} \tag{16}
$$

The Riemann problem. The solution of the Riemann problem for (16) (with $\varepsilon = 0$) is a composition of four simple waves; s- and c-rarefactions and s- and c-shocks. For Riemann problems with no jump in the concentration c, the solution reduces to that of the Buckley-Leverett equation, which is well known. Solutions of this kind will simply be termed s-waves, although they may be compositions of shocks and rarefactions.

For Riemann problems with a jump in c, the system behaviour is more complicated since the eigenvalues of (16) coincide along a curve T in phase space. In general, if $c^L > c^R$ the solution will consist of one c-shock and possibly some s-waves. If $c^L < c^R$, the solution will be made up of s-waves an c-rarefactions.

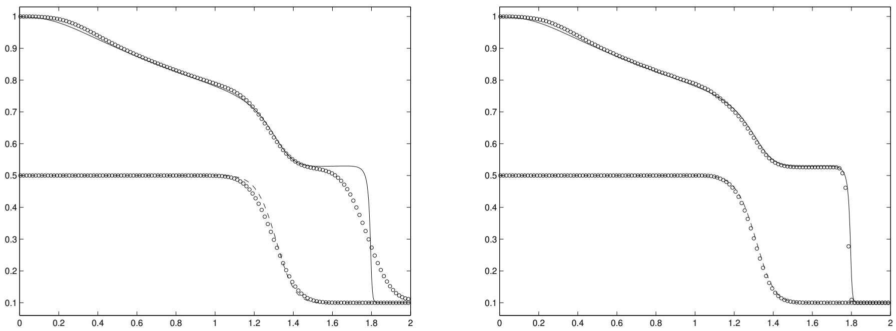

FIGURE 3. Example 1. Solution at time $t = 1.0$ computed using one OS step (left) and one COS step (right). The solution is plotted in primitive variables (s, c) at every second grid point. A reference solution computed by the central difference scheme (11) is included; the solid lines gives the s-component and the dashed line the c-component.

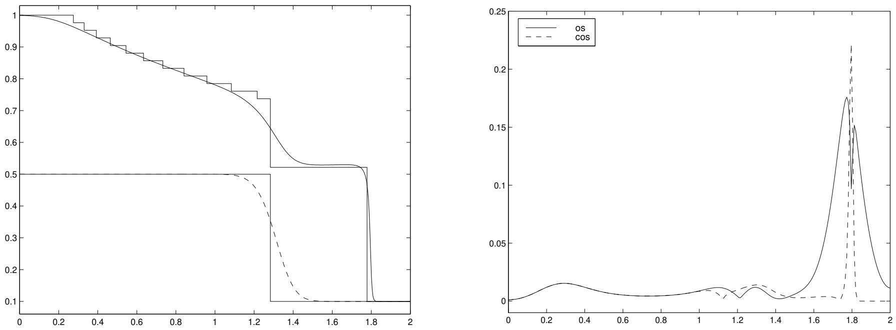

FIGURE 4. Example 1. (Left) The solution after the hyperbolic substep compared with the reference solution. (Right) Pointwise error in L^∞ norm relative to the reference solution.

For data completely on one side of T, the solution is a composition of one s-wave and one c-wave. However, when the left and right states $\mathbf{u}^L$ and $\mathbf{u}^R$ are on different sides of T, the solution can consist of up to five (or six) constant states, separated by simple waves. This case is shown in Example 2 of this section.

Due to the nonstrict hyperbolicity, there is one special case, called an overcompressive shock, where both the s- and c-characteristics go into the shock. The corresponding solution can be interpreted as the exact superposition of an s- and a c-shock, see Example 3 of this section.

Example 1. In our first example, we consider discontinuous initial data

$$(s_0, c_0)(x) = \begin{cases} (1.0, 0.5), & x \leq 0.1, \\ (0.1, 0.1), & x > 0.1. \end{cases}$$

In the inviscid case, the initial data correspond to a Riemann problem, which is solved by an s-shock, followed by a c-shock and an s-rarefaction wave.

Figure 3 shows the approximate solution at time $t = 1.0$ for $\varepsilon = 0.005$ computed using OS and COS with one time step on a 256 grid. Since the time step is much larger than the diffusion scale, OS gives a considerable smearing of the s-shock, whereas the c-shock contains little self-sharpening and is therefore represented quite accurately. When correction effects are included, both shock fronts are resolved almost perfectly. In this example, residual fluxes are defined for both the s-shock and the c-shock (see Figure 4), although the latter gives little effect. No fluxes are defined for the discontinuities in the rarefaction wave.

Figure 4 reports pointwise L^∞ errors for OS and COS. As expected, the major error contribution is around the s-shock. However, both OS and COS overshoot the s-component in the interval from $x = 0.1$ to $x = 0.5$. Because of mass conservation, the s-shock is therefore placed a bit to the left compared with the reference solution; hence the high peak in the pointwise error for COS.

To study the convergence of the splitting methods, we fix the spatial discretization to 2^{10} blocks on the interval $(-0.25, 2.25)$ and increase the number of splitting steps by powers of two. Errors for this convergence study are reported in Table 1 for $\varepsilon = 0.01$ and $\varepsilon = 0.001$. Here the error is defined as

$$E = \sum_{i=1}^{\ell} \frac{\|u_i - u_i^r\|_1}{\|u_i^r\|_1}, \tag{17}$$

where $(u_1, \ldots, u_\ell)$ denotes the splitting solution and $(u_1^r, \ldots, u_\ell^r)$ the reference solution.

TABLE 1. *Example 1. Estimated errors* (17) *and convergence rates for OS and COS with fixed spatial discretization; N_t denotes the number of splitting steps.*

	ε=0.01				ε=0.001			
N_t	OS	rate	COS	rate	OS	rate	COS	rate
1	4.42e-02	—	1.96e-02	—	1.88e-02	—	3.47e-03	—
2	2.90e-02	0.61	1.61e-02	0.29	1.26e-02	0.57	5.18e-03	-0.58
4	1.97e-02	0.56	1.22e-02	0.40	8.16e-03	0.63	4.30e-03	0.27
8	1.27e-02	0.63	8.53e-03	0.51	5.69e-03	0.52	3.27e-03	0.40
16	7.57e-03	0.75	5.65e-03	0.59	4.18e-03	0.44	2.52e-03	0.38
32	4.16e-03	0.86	3.58e-03	0.66	3.27e-03	0.35	2.22e-03	0.18
64	2.30e-03	0.86	2.45e-03	0.55	2.78e-03	0.23	2.17e-03	0.03
128	1.48e-03	0.63	1.53e-03	0.68	2.85e-03	-0.03	2.58e-03	-0.25
256	1.30e-03	0.19	1.31e-03	0.23	3.61e-03	-0.34	3.52e-03	-0.45

For $\varepsilon = 0.001$, the error for OS decreases up to $N_t = 64$, but then starts to increase as a result of increasing numerical diffusion introduced by the projection in the hyperbolic steps. For COS, the error increases when going from one to two splitting steps, because with two splitting steps the c-shock is not fully formed in the second hyperbolic step, and hence the residual flux is weaker. With four (or more) splitting steps, a residual flux for the c-shock is only formed in the first step.

For $\varepsilon = 0.01$, the OS error decreases for all N_t (but it increases for $N_t = 512$). By introducing corrections, we remove most of the splitting error around the s-shock, but not in the interval from $x = 0$ to $x = 1.5$, see Figure 5. Therefore, the COS error also decreases with increasing number of splitting steps. Since the c-shock is completely smoothed out by the diffusive forces, the corresponding residual flux is formed only in the first hyperbolic step.

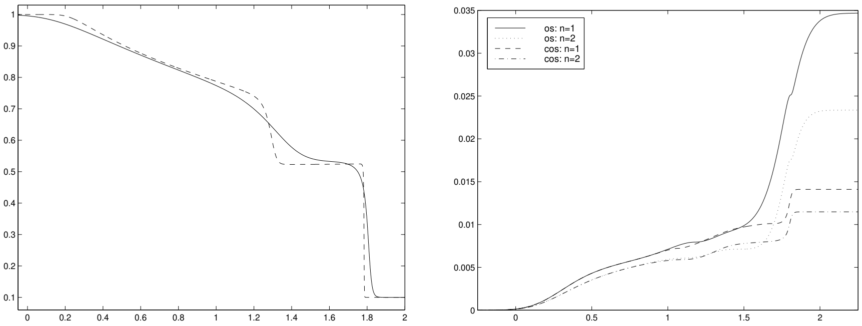

FIGURE 5. Example 1. (Left) The s-component of the reference solution for $\varepsilon = 0.01$ (solid) and 0.001 (dashed). (Right) Cumulative L^1 error of the s-component for OS and COS with one and two splitting steps for $\varepsilon = 0.01$.

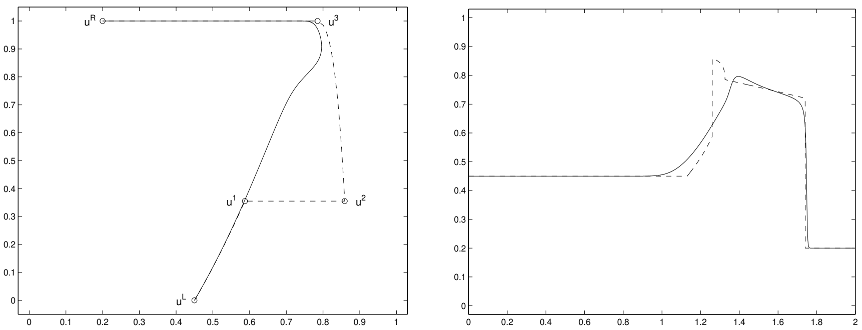

FIGURE 6. Example 2. (Left) The solution in (s, c)-space; solid line represents the solution for $\varepsilon = 0.0025$ at time $t = 1.0$ and the dashed line the inviscid solution. (Right) The s-component of the solutions.

Example 2. The next example demonstrates that the corrected splitting method also applies to non-monotone data. To this end, consider the Riemann initial data

$$(s_0, c_0)(x) = \begin{cases} (0.45, 0.0), & x \leq 0.1, \\ (0.2, 1.0), & x > 0.1. \end{cases}$$

In the inviscid case, this problem gives the maximum number of intermediate states in the Riemann solution, i.e., a solution of the form $u^L \overset{c}{\to} u^1 \overset{s}{\to} u^2 \overset{c}{\to} u^3 \overset{s}{\to} u^R$, where $\overset{s}{\to}$ denotes an s-wave and $\overset{c}{\to}$ a c-wave, see e.g., [19, Ex. 8.2]. The solutions in the inviscid case and for $\varepsilon = 0.0025$ are shown in Figure 6.

Figure 7 shows approximate solutions computed by OS and COS on a 2^{10} grid; estimated L^1 errors in each grid cell are shown in Figure 8. With one splitting step, COS resolves the leading s-wave accurately. However, since a residual flux is also identified for the s-shock at $x \approx 1.26$, COS produces an extra peak in the solution (resembling the inviscid solution). With two splitting steps, the OS and COS solutions coincide along the trailing edge. With four splitting steps, COS gives a fairly accurate approximation of the exact solution. Notice that four splitting steps corresponds to a CFL number around 300!

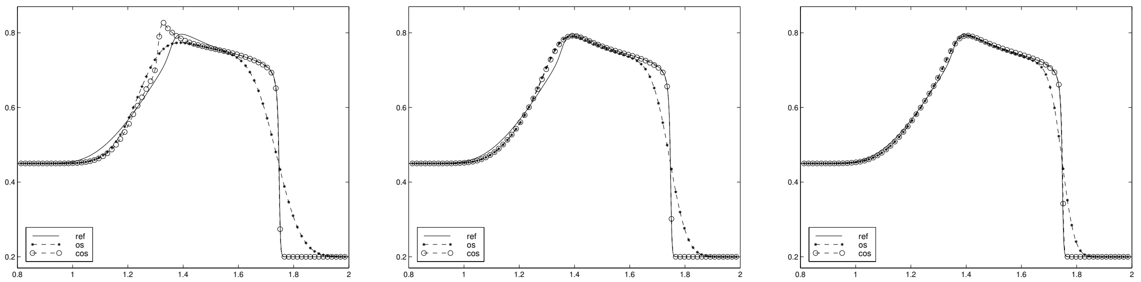

FIGURE 7. Example 2. Comparison of OS (small dots) and COS (large dots) with one, two, and four splitting steps, from left to right. The plots show the s-component at time $t = 0.5$ with markers at every eight grid point.

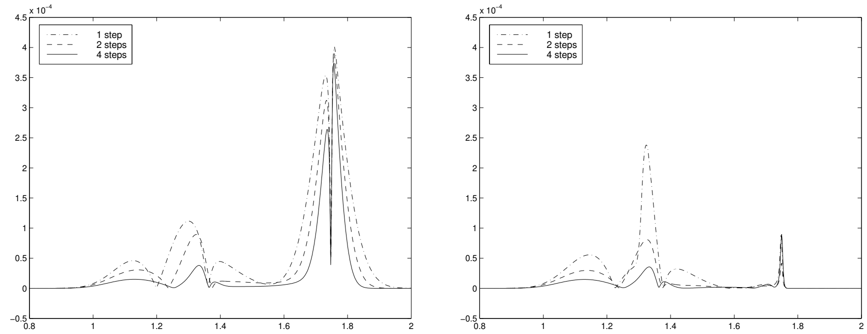

FIGURE 8. Example 2. L^1 error in each grid cell for OS (left) and COS (right).

Example 3. In the last polymer example, we consider a Riemann problem corresponding to an (over)compressive shock $u^L \overset{c}{\to} u^R$ in the inviscid case; compressive shock means that both the s- and the c-characteristics go into the shock and contribute to the self-sharpening.

The Riemann problem is given by

$$(s_0, c_0)(x) = \begin{cases} (0.75, 0.8), & x \le 0.25, \\ (0.839619\ldots, 0.4), & x > 0.25. \end{cases}$$

If this Riemann problem is perturbed slightly, the solution changes from a single shock to a composition of waves moving with almost the same speed, as shown in Figure 9. There are two possible results of a perturbation; either a monotone solution $u^L \overset{s}{\to} u^1 \overset{c}{\to} u^R$ or a non-monotone solution $u^L \overset{c}{\to} u^2 \overset{s}{\to} u^R$ (as shown Figure 9).

In the viscid case, the Riemann problem will be perturbed instantly. The result is a truly nonlinear phenomenon, where monotone data gives non-monotone solutions. Therefore, the solution of the inviscid case is a poor approximation to the true solution, and both the OS and COS strategy will fail to resolve the non-monotonicity in the first time step. Figure 10 shows the exact solution at time $t = 1.0$ along with the OS and COS approximations for one time step. The exact (reference) solution has a dip due to the presence of the diffusion term, which instantly perturbs the initial Riemann problem. The OS approximation is monotone and does not resolve the nonlinear dynamics between the convective and diffusive forces, thereby giving the usual smearing of the steep gradient. COS, on the other hand, does not produce

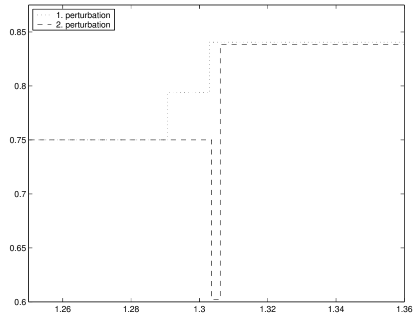

FIGURE 9. Example 3. Wave patterns in the s-component for two slightly perturbed right states ($s_1^R = s_0^R + 0.001$ and $s_2^R = s_0^R - 0.001$) for overcompressive shock case.

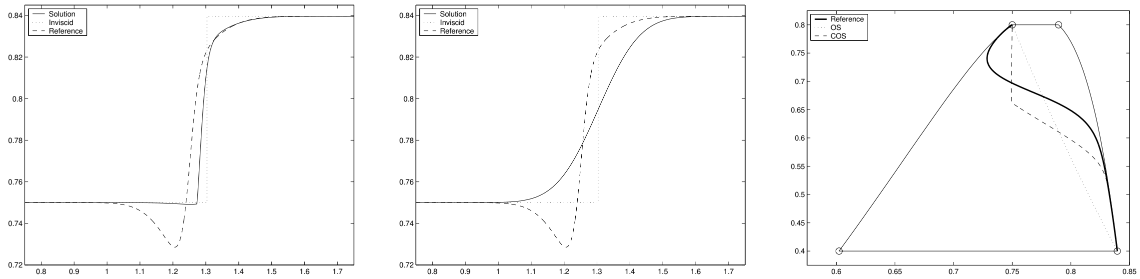

FIGURE 10. Example 3. Comparison of the s-component of splitting solutions at time $t = 1.0$ for $\varepsilon = 0.005$ computed with one step of COS (left) and one step of OS (middle). The right plot shows solutions in state space. The thin solid lines represents the waves of the slight perturbations (s_1^R and s_2^R).

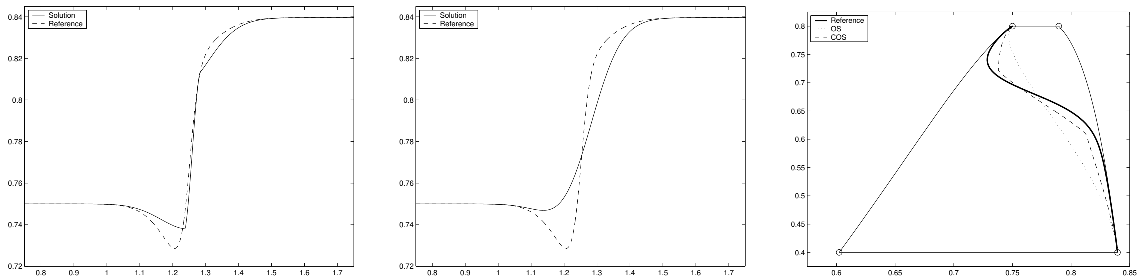

FIGURE 11. Example 3. Comparison of the s-component of splitting solutions computed with two steps of COS (left) and OS (middle). The right plot shows solutions in state space.

the same excessive smearing and gives a very small dip introduced by the nonlinear residual. In both cases the solution misses on the position of the shock because of mass conservation.

With two time steps, as in Figure 11, the difference is more distinct. Neither OS nor COS are able to resolve the dip in the solution properly. This is due to splitting errors from the coarse time discretization. However, when comparing OS and COS, we see that COS is much closer to the reference solution in the steep part (s-shock).

In Table 2 we report the estimated errors for a number of runs with decreasing splitting steps, along with convergence rates. COS and OS have more or less the same rates of convergence. We note, however, that COS in general has lower error than OS. This is caused by the sharpening effect of the residual flux.

When the number of splitting steps becomes large, the solutions deteriorate due to the many projections that are performed; see the discussion in Example 2.

TABLE 2. *Example 3. Estimated errors* (17) *and convergence rates for OS and COS with fixed spatial discretization* $\Delta x = 2^{-9}$*;* N_t *denotes the number of splitting steps.*

	ε=0.005				ε=0.01			
N_t	OS	rate	COS	rate	OS	rate	COS	rate
1	6.93e-03	—	4.31e-03	—	8.11e-03	—	5.05e-03	—
2	3.94e-03	0.57	2.95e-03	0.38	4.40e-03	0.61	3.60e-03	0.34
4	2.15e-03	0.61	2.67e-03	0.10	2.26e-03	0.67	2.15e-03	0.51
8	1.13e-03	0.64	9.94e-04	0.99	1.16e-03	0.67	1.02e-03	0.74
16	6.23e-04	0.60	5.84e-04	0.53	6.22e-04	0.62	5.86e-04	0.56
32	3.89e-04	0.47	3.77e-04	0.44	3.82e-04	0.49	3.71e-04	0.46
64	3.04e-04	0.25	2.97e-04	0.24	2.80e-04	0.31	2.76e-04	0.30
128	2.85e-04	0.07	2.82e-04	0.05	2.48e-04	0.12	2.46e-04	0.11
256	2.89e-04	-0.02	2.88e-04	-0.02	2.47e-04	0.00	2.46e-04	0.00

Remark. In the above examples we have not elaborated on how to choose the threshold parameter γ used to identify residual fluxes. When choosing this parameter there are two points to consider: First, γ should be sufficiently small, so that relevant residual fluxes are identified. Second, γ should be chosen so large that no residual fluxes are identified for small-scale oscillations appearing in the hyperbolic solutions (e.g., post-shock oscillations, or oscillations for data near the transition curve [31]). Using these rules of thumb, a suitable parameter can easily be identified in each case, without any special tuning necessary. In all the above runs, we used a parameter value of $\gamma = 0.1$ (as the size of the oscillations typically were below 0.05).

3.2. **A Triangular Three-Phase Flow Model.** We consider the initial value problem for the 2×2 system of parabolic equations

$$\begin{aligned} \partial_t u + \partial_x f(u) &= \varepsilon \partial_x^2 d(u), \\ \partial_t v + \partial_x g(u,v) &= \varepsilon \partial_x^2 d(v), \end{aligned} \tag{18}$$

where u and v are phase saturations (gas and water, respectively), f and g the fractional flow functions, d the diffusion function, and ε is a small scaling factor. The Jacobian matrix of the fractional flow functions is (lower) triangular, hence a triangular system. The system (18) models flow of oil, gas, and water in a porous medium. Generally, no systems arising from three-phase flow models are triangular, but by assuming that one of the phases, typically the gas phase, is decoupled from the other two we get the system above. In many cases this is a reasonable assumption. Here we use flux functions [25]

$$f(u) = \frac{u^2}{u^2 + (1-u)^2/10},$$

$$g(u,v) = \frac{(1-u)^2 + u^2/10}{10u^2 + (1-u)^2} \cdot \frac{v^2}{v^2 + (1-v)^2/10},$$

and the diffusion function is given as

$$d(x) = 4x(1-x) + 0.01,$$

which thus has the usual bell shape typically seen in reservoir models. For simplicity, a small constant has been added to avoid degeneration of the system. Alternatively, we could have treated the degeneracy by using upwind discretizations of the convective terms.

We present two numerical examples, one simple Riemann problem and a Cauchy problem with non-monotone solution. In both cases, we compare COS and OS solutions against a reference solution computed with a central difference scheme on a very fine discretization.

The Riemann solver. The essential part of the hyperbolic solution operator in our operator splitting method is the Riemann solver whose construction depends heavily on the triangular nature of the system. The procedure for constructing the solution of the Riemann problem is found in Gimse [14]. Let us very briefly recall the basic steps in this procedure.

We solve the Riemann problem for (18) with linearized flux functions (the linearization parameter is denoted by $\delta > 0$). The scalar decoupled equation for u (the first equation in (18)) is easily solved, and the solution u consists of a finite number (say N) of constant states. Rarefaction waves are approximated by step functions. Then we solve the (coupled) equation for v (the second equation in (18)). In each region in the (x,t) plane where u is constant, the coupled equation reduces to a scalar equation. In order to solve the coupled equation, we must therefore solve N scalar equations and connect together the solutions from the neighbouring regions. To find legal connection values, i.e., legal v-values on each side of the interface between two regions, we must construct so-called H-sets. We mention that the construction of the H-sets is the most time consuming part of the Riemann solver. For further details on the Riemann solver, we refer to [14].

There are three wave types in the solution of the triangular Riemann problem. From the scalar Riemann problem, we have rarefaction waves and shock waves. In addition, there are connection waves, i.e., waves connecting the v-regions in the (x,t) plane. The connection waves will always have the same speed as the corresponding discontinuities in u.

In the numerical examples below, the flux functions are linearized with $\delta = 0.01$. This sets a limit on the resolution of the front tracker and hence the expected accuracy of the solution.

Example 1. As for the polymer system, we consider first a simple Riemann problem:

$$(u_0, v_0)(x) = \begin{cases} (0.4, 0.6), & x \leq 1.0, \\ (0.0, 0.0), & x > 1.0. \end{cases} \tag{19}$$

The diffusion coefficient ε is set equal to 0.1, and we seek solutions at time $t = 0.5$. The finite difference operator uses a grid with $\Delta x = 0.01$.

Figure 12 shows solutions computed using one step of OS and COS compared with a reference solution. The COS method sharpens the shock front in u nicely. In addition, the COS method incorrectly sharpens the quite smooth front in v. As a result of this sharpening, the error in v close to the shock is larger for COS than for OS. The left plot in Figure 13 shows local L^1 error in each grid cell as a function of the position of the cell center. The remedy for this erroneous sharpening is to decrease the time step. Figure 13 (right plot) shows cumulative L^1 errors for COS and OS with one and two steps. We observe that already when two time steps are used instead of one, the erroneous sharpening in v (at $x \approx 1.5$) is

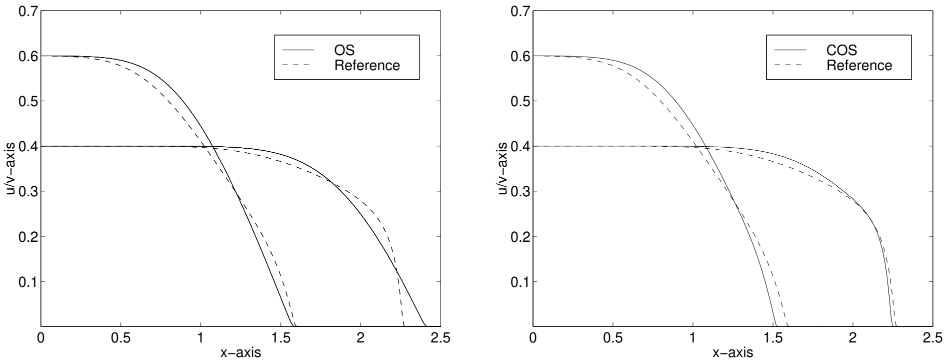

FIGURE 12. Example 1. Solution of Riemann problem (19) at time $t = 0.5$ for $\varepsilon = 0.1$ computed using one step of OS (left) and one step of COS (right).

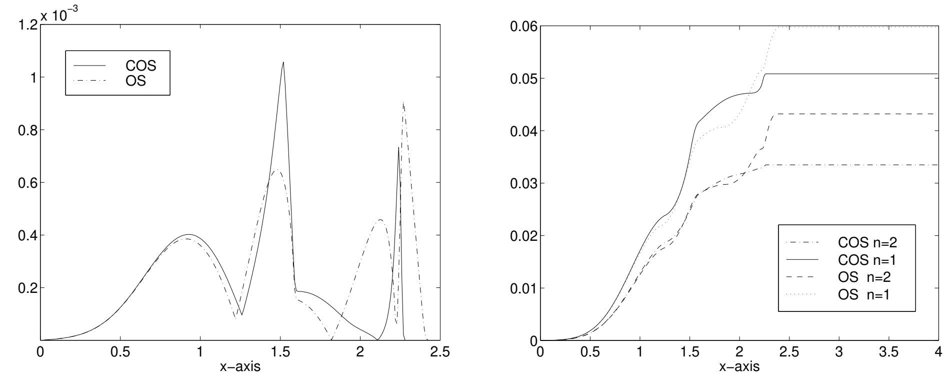

FIGURE 13. Example 1. Error measurements for Riemann problem (19) at time $t = 0.5$ for $\varepsilon = 0.1$. (Left) L^1 error in each grid cell for COS and OS with one time step. (Right) Cumulative L^1 error for COS and OS with one and two steps.

almost gone, and COS outperforms OS. This is because the front in v is smoother in the second time step and therefore no residual flux is defined.

Example 2. In the second example, we solve the Cauchy problem for (18) with initial data

$$u_0(x) = v_0(x) = \begin{cases} 0.0, & x < 0.4, \\ x - 0.4, & 0.4 < x < 0.8, \\ 0.4, & x > 0.8. \end{cases} \tag{20}$$

Final computing time is $t = 1.0$ and we set $\varepsilon = 0$. The finite difference operator uses a grid with $\Delta x = 0.01$.

The performance of COS in this example is very good. In Figure 14, OS and COS solutions computed with one time step are compared with the reference solution. The OS solution is nowhere near catching the shock fronts in u and v. On the other hand, the correction achieved by adding the residual fluxes enables the COS method to resolve the shock fronts almost perfectly.

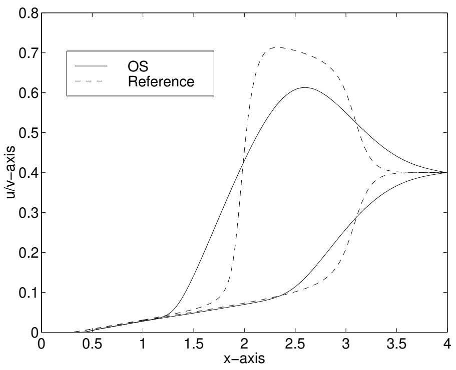

FIGURE 14. Example 3. Solutions of Cauchy problem (20) at time $t = 1.0$ for $\varepsilon = 0.1$ computed using one step of OS (left) and one step of COS (right).

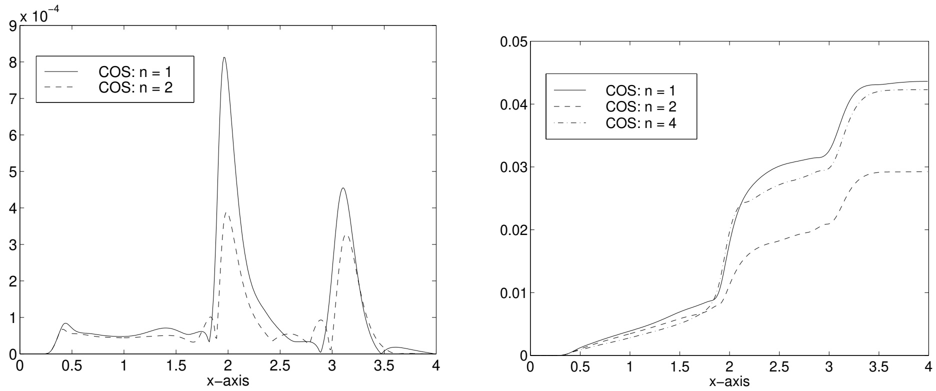

FIGURE 15. Example 3. Error measurements for COS for Cauchy problem (20) at time $t = 1.0$ with $\varepsilon = 0.1$; (left) L^1 error in each grid cell, (right) cumulative L^1 error.

When going from one to two time steps in COS, we observe a nice decrease in the error, see Figure 15 (left plot). However, when increasing the number of splitting steps to four, the error increases almost to the level observed for one step, see Figure 15 (right plot). The reason is the way the residual fluxes are defined. If the distance between two shocks is less than the (prescribed) length of the correction intervals, the residual fluxes are only defined to the midpoint between the two shocks. When the time step decreases, the distance between the two v-shocks also decreases. As a result, the correction interval for each residual flux also decreases, and the sharpening effect is reduced. Thus the improvement of the OS solution is smaller, but COS still gives a lower error than OS with the same number of time steps.

To study the convergence of OS and COS, we fix the spatial discretization Δx to 0.01 on the interval $(0, 4)$ and increase the number of splitting steps by powers of two. Table 3 gives errors for this convergence study for $\varepsilon = 0.1$. The error is defined as

$$E = \sum_{i=1}^{\ell} \|u_i - u_i^r\|_1, \tag{21}$$

where $(u_1, \dots, u_\ell)$ denotes the splitting solution and $(u_1^r, \dots, u_\ell^r)$ the reference solution. Table 3 shows errors and convergence rates for COS and OS. The error for OS decreases up to $N_t = 64$, but then starts to increase as a result of numerical diffusion. The error for COS first decreases and then increases for $N_t = 4, 8$ as a result of decreasing distance between the two v-shocks (as explained above). For N_t larger than 8, COS performs as OS.

TABLE 3. *Example 3. Estimated errors* (21) *and convergence rates for OS and COS with fixed spatial discretization* $\Delta x = 0.01$*;* N_t *denotes the number of splitting steps.*

	ε=0.1			
N_t	OS	rate	COS	rate
1	2.76e-01	—	4.36e-02	—
2	1.75e-01	0.65	2.92e-02	0.58
4	9.76e-02	0.84	4.23e-02	-0.53
8	5.51e-02	0.82	5.59e-02	-0.40
16	3.19e-02	0.79	2.97e-02	0.91
32	2.15e-02	0.56	2.11e-02	0.49
64	1.80e-02	0.26	1.80e-02	0.24
128	1.99e-02	-0.14	1.99e-02	-0.15
256	2.30e-02	-0.20	2.31e-02	-0.21

Remark. The COS method gives very good results. Using an implicit diffusion solver would make the runtimes go down and thus make the COS method very efficient. We have not done this here, since we were merely interested in testing whether COS reduces the temporal splitting error compared to OS. In the COS method we have chosen to prescribe the length of the correction intervals as $K\varepsilon$ with $K = 10$. Moreover, as for the polymer system in Section 3.1, we chose $\gamma = 0.1$ as the value for the threshold parameter used to pick out the shocks that define the residual fluxes.

4. A Two-Dimensional Extension

In this section we present a two-dimensional extension of the COS idea for the polymer system using dimensional splitting. The same idea can be applied to other systems and higher dimensions. To add more realism to the equation, we also include a driving velocity field. That is, we consider

$$\begin{aligned} \partial_t s + V(x) \cdot \nabla f(s,c) &= \varepsilon \Delta s \\ \partial_t b + V(x) \cdot \nabla (cf(s,c)) &= \varepsilon \Delta b, \end{aligned} \tag{22}$$

where f is given by (15) with $\mu = 0.25$ and $\nu = 9$, $x = (x_1, x_2)$, $V = (V_1, V_2)$, $\nabla = (\partial_{x_1}, \partial_{x_1})$, and $\Delta = \partial_{x_1}^2 + \partial_{x_2}^2$. The velocity field is given by the pressure equation combined with Darcy's law:

$$-\nabla(\lambda(s,c)\nabla p) = 0, \quad V = -\lambda(s,c)\nabla(p), \tag{23}$$

where λ denotes the total mobility.

A common strategy to solve the overall system (23) and (22) is to decouple the equations by operator splitting: First solve the pressure equation (23) with coefficients given by the initial fluid distribution. Then the velocity field is computed using Darcy's law, and this velocity is held fixed while the saturation s and polymer concentration c are advanced forward according to (22). Then the process is repeated. In the following we will assume that the velocity field is a given quantity and concentrate on the solution of (22).

Let $\Delta x > 0$ and $\Delta t > 0$ denote the spatial and temporal discretization parameters associated with our discrete splitting method. Let $\mathcal{S}_t^{x_j}$ denote the front tracking solution operator associated with the following one-dimensional system of conservation laws

$$\begin{aligned} \partial_t s + V_j(x)\partial_{x_j} f_j(s,c) &= 0 \\ \partial_t b + V_j(x)\partial_{x_j}(c f_j(s,c)) &= 0. \end{aligned} \tag{24}$$

The front tracking algorithm can easily be modified to incorporate varying velocity fields, see e.g., Lie [26] or Haugse *et al.* [16]. Moreover, residual fluxes are identified as outlined above for $V \equiv 1$. Now, let $\mathcal{P}_t^{x_j}$ denote the finite difference solution operator of the following parabolic system

$$\begin{aligned} \partial_t s + V_j(x)\partial_{x_j} f_1^{\mathrm{res},j}(x,s,c) &= \varepsilon \partial_{x_j}^2 s \\ \partial_t b + V_j(x)\partial_{x_j} f_2^{\mathrm{res},j}(x,s,c) &= \varepsilon \partial_{x_j}^2 b. \end{aligned} \tag{25}$$

Note that x_i, $i \neq j$, act only as parameters in (25). We then introduce the following operator splitting solution

$$(s,b)(x,n\Delta t) \approx (s^n, b^n) \equiv \left[\mathcal{P}_{\Delta t}^{x_2} \circ \Pi \circ \mathcal{S}_{\Delta t}^{x_2} \circ \mathcal{P}_{\Delta t}^{x_1} \circ \Pi \circ \mathcal{S}_{\Delta t}^{x_1}\right]^n \Pi(s_0, b_0). \tag{26}$$

Example 1. In the first example, we consider a standard test case from reservoir simulation, the quarter five-spot case, which consists of a unit square with an injection well in the lower left corner and a production well in upper right. A no-flow condition is specified along the boundaries. We consider a reservoir initially filled with pure oil into which pure water with a polymer concentration 0.1 is injected. The diffusion coefficient ε is set to 0.005. To solve the parabolic systems (25) we use a componentwise upwind method.

Figure 16 shows the saturation component of the solution at time $t = 0.4$ (corresponding to 0.4 pore volumes of fluid injected) computed by COS and OS. A reference solution computed on a 257×257 grid is given in Figure 17. Due to very large velocities near the wells, we use small splitting steps (CFL number 2.0) up to time $t = 0.02$ to stabilize the profile and then four splitting steps to reach final time $t = 0.4$. The splitting steps have been intentionally chosen large to demonstrate the error mechanisms in the shock layers, hence the lack of symmetry due to dimensional splitting errors. Much of the dimensional splitting error is removed by increasing the number of splitting steps to e.g., 16. Furthermore, we see that OS and COS coincide except near the diagonal, where the velocity field is sufficiently strong to form discontinuities in the hyperbolic steps and hence residual fluxes.

Example 2. In the second example, we add a heterogeneous permeability field to the quarter five-spot. The permeability field is realised as a log-Gaussian random field. Due to the heterogeneity, the water front will contain viscous fingers. To resolve the fingering properly, one has to reduce the splitting steps compared with the example above. Figure 18 shows saturation and polymer concentration profiles at time $t = 0.3$ for three different values of ε. The length of the splitting steps vary according to a CFL target 4.0 for the hyperbolic steps. Although residual fluxes are seldom identified in these simulations (for $\varepsilon = 0.05$ and 0.005), the mechanism is embedded and is automatically invoked whenever necessary to prevent viscous splitting errors.

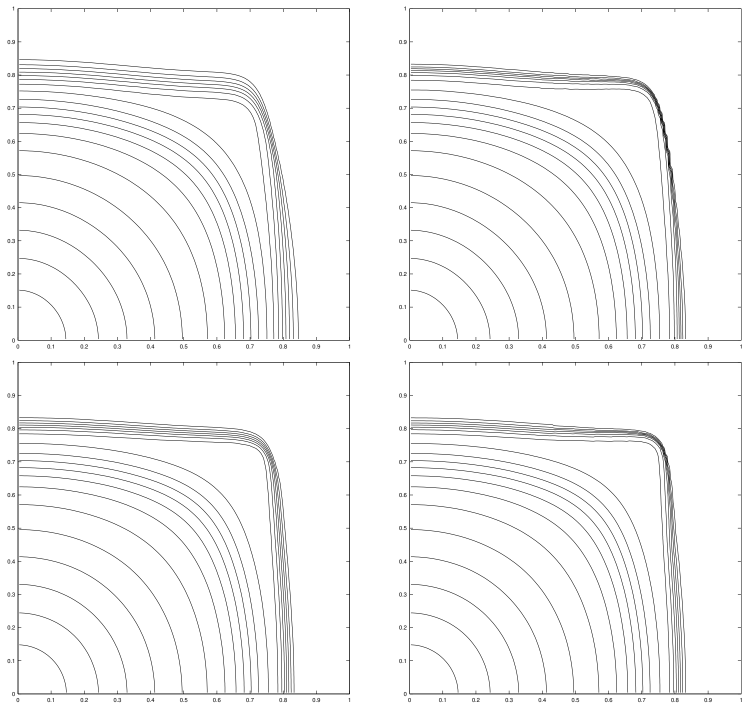

FIGURE 16. Example 1. Saturation component of the solution at time $t = 0.4$ computed by OS (left) and COS (right) with 4 steps (upper row) and with 16 steps (lower row).

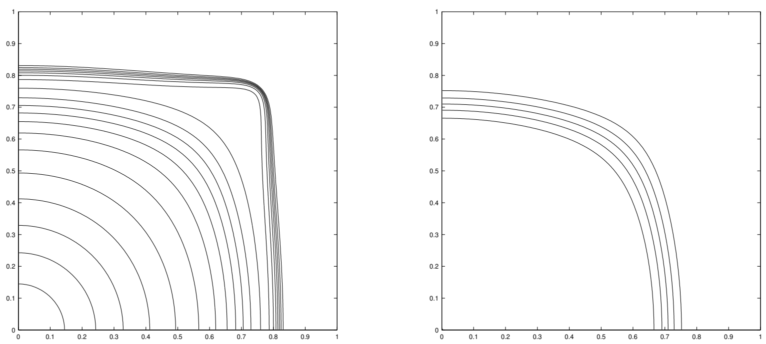

FIGURE 17. Example 1. Saturation (left) and polymer concentration (right) computed on a 257×257 grid with CFL number 2.0.

5. Concluding Remarks

We have demonstrated numerically that operator splitting (OS) methods for systems of convection-diffusion equations in one space dimension have a tendency to be too diffusive near viscous shock waves. In the scalar case [23, 20], this is related to the fact that the entropy condition (Oleinik's convexification criterion) forces the hyperbolic solver in the convection step to throw away information about the structure of the viscous shock waves, at least when the splitting step is large, thereby creating an entropy loss. To reduce this temporal splitting

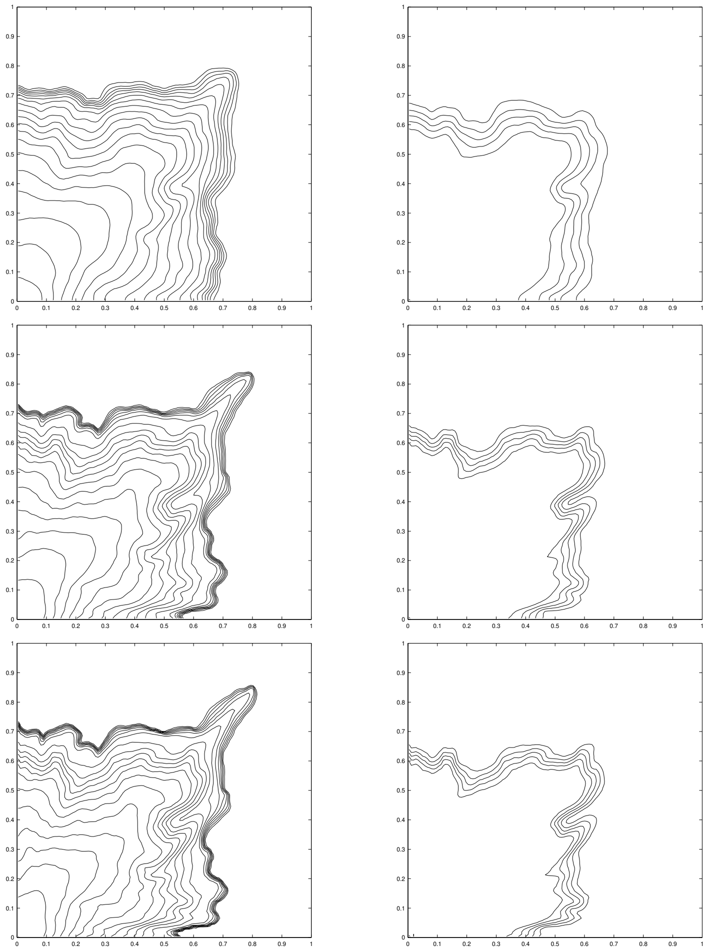

FIGURE 18. Example 2. Saturation and concentration profiles at time $t = 0.3$ computed on a 129×129 grid. The diffusion coefficient is $\varepsilon = 0.05$, 0.005, and 0.0005 from top to bottom.

error, Karlsen and Risebro [23] (see also [20, 12]) introduced the corrected operator splitting (COS) method for scalar convection-diffusion equations. The idea behind the scalar COS method is to use the wave structure from the convection step to identify where the nonlinear splitting error (or entropy loss) occurs. The potential error is then compensated for in the diffusion step (or in a separate correction step).

In the present paper, we have outlined the mechanisms behind the splitting error for systems of convection-diffusion equations. Similarly to the scalar case, the splitting error is intimately related to the local linearizations introduced implicitly in the convection steps due to the use of an entropy condition. Moreover, we have proposed a working COS method for systems. A front tracking method [27, 28, 29] for systems of conservation laws, which in turn relies on a Riemann solver, is an important part of this COS method. The proposed COS method has been applied to a 2×2 system modelling two-phase, multicomponent flow and a triangular 2×2 system modelling three-phase flow. The numerical examples demonstrate that the COS method is significantly more accurate than the corresponding OS method when the splitting step is large and the solution consists of (moving) viscous shock waves. We have extended the COS method to two-dimensional systems of convection-diffusion equations by means of dimensional splitting and applied it to a polymer system with a driving velocity field. In closing, we mention that the COS method will be applied to other systems in future work.

References

[1] A. Bressan, T.-P. Liu, and T. Yang. L^1 stability estimates for $n \times n$ conservation laws. Preprint, S.I.S.S.A., Trieste, 1998.

[2] H. K. Dahle. *Adaptive characteristic operator splitting techniques for convection-dominated diffusion problems in one and two space dimensions.* PhD thesis, Department of Mathematics, University of Bergen, Norway, 1988.

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

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

[5] H. K. Dahle, M. S. Espedal, and O. Sævareid. Characteristic, local grid refinement techniques for reservoir flow problems. *International Journal for Numerical Methods in Enginering*, 34:1051–1069, 1992.

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

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

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

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

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

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

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

[13] S. Evje, K. H. Karlsen, K.–A. Lie, and N. H. Risebro. Front tracking and operator splitting for nonlinear degenerate convection–diffusion equations. To appear in *Parallel solution of partial differential equations*, Eds. P. Bjørstad and M. Luskin, *IMA Vol. Math. Appl.*, Vol. 120, pp. 209-228, Springer Verlag.

[14] T. Gimse. A numerical method for a system of equations modelling one-dimensional three-phase flow in a porous medium. In *Nonlinear hyperbolic equations—theory, computation methods, and applications (Aachen, 1988)*, pages 159–168. Vieweg, Braunschweig, 1989.

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

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

[17] D. Hoff and J. Smoller. Error bounds for finite-difference approximations for a class of nonlinear parabolic systems. *Math. Comp.*, 45(171):35–49, 1985.

[18] H. Holden, K. H. Karlsen, K.–A. Lie, and N. H. Risebro. Numerical solution of nonlinear partial differential equations using operator splitting methods: An L^1 convergence theory. Preprint (in preparation).

[19] 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.

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

[21] K. H. Karlsen and K.–A. Lie. An unconditionally stable splitting scheme for a class of nonlinear parabolic equations. *IMA J. Numer. Anal.*, 19(4):609–636, 1999.
[22] K. H. Karlsen, K.–A. Lie, N. H. Risebro, and J. Frøyen. A front-tracking approach to a two-phase fluid-flow model with capillary forces. *In Situ (Special issue on reservoir simulation)*, 22(1):59–89, 1998.
[23] K. H. Karlsen and N. H. Risebro. Corrected operator splitting for nonlinear parabolic equations. *SIAM J. Numer. Anal.* To appear.
[24] K. H. Karlsen and N. H. Risebro. An operator splitting method for convection-diffusion equations. *Numer. Math.*, 77(3):365–382, 1997.
[25] J. A. Kok. Front tracking for three phase flow in porous media. Final report of the postgraduate program 'Mathematics for industry', Eindhoven, Netherlands, 1994.
[26] K.-A. Lie. A dimensional splitting method for quasilinear hyperbolic equations with variable coefficients. *BIT*, 39(4):683–700, 1999.
[27] N. H. Risebro. A front-tracking alternative to the random choice method. *Proc. Amer. Math. Soc.*, 117(4):1125–1139, 1993.
[28] 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.
[29] N. H. Risebro and A. Tveito. A front tracking method for conservation laws in one dimension. J. Comp. Phys., 101(1):130–139, 1992.
[30] A. Tveito. Convergence and stability of the Lax-Friedrichs scheme for a nonlinear parabolic polymer flooding problem. *Advances In Applied Mathematics*, 11:220–246, 1990.
[31] 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.
[32] M. F. Wheeler, W. A. Kinton, and C. N. Dawson. Time-splitting for advection-dominated parabolic problems in one space variable. *Comm. Appl. Numer. Methods*, 4(3):413–423, 1988.