THE CORRECTED OPERATOR SPLITTING APPROACH APPLIED TO A NONLINEAR ADVECTION-DIFFUSION PROBLEM

K. HVISTENDAHL KARLSEN, K. BRUSDAL, H. K. DAHLE, S. EVJE, AND K.–A. LIE

ABSTRACT. So-called corrected operator splitting methods are applied to a 1-D scalar advection-diffusion equation of Buckley–Leverett type with *general* initial data. Front tracking and a $2nd$ order Godunov method are used to advance the solution in time. Diffusion is modelled by piecewise linear finite elements at each new time level. To obtain correct structure of shock fronts independently of the size of the time step, a *dynamically defined* residual flux term is grouped with diffusion. Different test problems are considered, and the methods are compared with respect to accuracy and runtime. Finally, we extend the corrected operator splitting to 2-D equations by means of dimensional splitting, and we apply it to a Buckley–Leverett type problem including gravitational effects.

1. INTRODUCTION

The numerical solution of advective-diffusive transport problems arise in many important applications in science and engineering, e.g. oil reservoir flow, transport of solutes in ground water and surface water, the movement of aerosols and trace gases in the atmosphere, etc. The generic problem to be considered in the present paper is the initial-boundary value problem for a scalar advection-diffusion equation, that is,

$$\partial_t u + \partial_x f(u) = \varepsilon \partial_x \left[\nu(u)\partial_x u\right], \tag{1}$$

for $(x,t) \in (a,b) \times (0,T] \subset \mathbb{R} \times \mathbb{R}^+$, together with initial and boundary data imposed as follows

$$\begin{cases} u(x,0) = u_0(x), & \text{for } x \in (a,b), \text{ where } u_0 : \mathbb{R} \to \mathbb{R} \text{ is bounded,} \\ u(a,t) = u_a, & \text{for } t \in (0,T], \text{ where } u_a \in \mathbb{R}, \\ u(b,t) = u_b, & \text{for } t \in (0,T], \text{ where } u_b \in \mathbb{R}. \end{cases} \tag{2}$$

Our main motivation for studying (1) stems from the fact that this nonlinear partial differential equation constitutes an important part of a (two-phase flow) system of equations modelling displacement of oil by water in oil reservoirs [5]. In this context, (1) is often referred to as the Buckley–Leverett equation. Furthermore, the fractional flow (or flux)

Key words and phrases. Nonlinear advection-diffusion equation, operator splitting, front tracking, Godunov methods, modified method of characteristics.

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

function f has the usual s-shaped form, which we mimic using the analytic expression

$$f(u) = \frac{u^2}{u^2 + (1-u)^2}. \tag{3}$$

The capillary diffusion coefficient $\nu(u)$ is generally a nonlinear (bell-shaped) function of u, which we recreate using the simple expression

$$\nu(u) = 4u(1-u). \tag{4}$$

We note that the diffusion coefficient becomes zero at $u = 0, 1$, so that (1) is an example of a parabolic degenerate equation; see [36] for global existence, uniqueness, and stability results in BV space of the initial value problem for *general* $\nu(u) \geq 0$, and [41] for similar results concerning the initial-boundary value problem. Properties of BV solutions, such as regularity, have been studied in [42]. For further results on degenerate equations we refer to the recent survey paper [6] and the references therein.

The scaling parameter ε in front of the capillary diffusion term is usually small for reasonable flow rates. Consequently, this term can be neglected if the main problem is to trace fluid interfaces, in which case the problem is reduced to solving a hyperbolic conservation law. However, in many applications some detailed information on the structure of fronts is important, and the diffusion term cannot be neglected. Since we believe that it is important to obtain correct placement and structure of fronts, both these problems have to be addressed so that different balances of advection and diffusion can be modelled in an accurate and consistent way within the same application. This is however difficult to achieve numerically, especially when the process is dominated by a nonlinear advection term. Unless impractically small discretization parameters are used, conventional methods usually exhibit some combination of difficulties, from non-physical oscillations (centred difference and Galerkin finite element schemes) to excessive numerical diffusion (upwinding difference schemes).

A natural strategy to overcome these difficulties is to split the advection-diffusion equation (1) into a hyperbolic conservation law and a parabolic equation, each of which is solved separately, i.e., an advection step followed by a diffusion step. This may be done in a straightforward way, at least for the scalar problem considered here [26]. Furthermore, each step of such an algorithm may be fully discretized by efficient methods allowing for long, stable, and accurate time steps. The approach just described, which we often refer to as (viscous) *operator splitting* (OS), or at least certain variations on this approach, has indeed been taken by many authors [2, 11, 15, 16, 17, 19, 20, 26, 33]. A more general approach is the so-called ELLAM (Eulerian-Lagrangian Localized Adjoint Methods) framework for linear equations with various boundary conditions [4, 34, 37, 40]. Operator splitting has also been applied in terms of dimensional splitting for multi-dimensional conservation laws [7, 24] and for equations with source terms [8, 23, 29, 38, 39].

In the constant diffusion case, it is easy to see that viscous splitting errors, due to nonlinearity, may lead to front widths of size $\mathcal{O}(\sqrt{\varepsilon \Delta t})$ when the time step becomes too large, whereas physical front widths should be $\mathcal{O}(\varepsilon)$ [35]. One of the first attempts to overcome this difficulty was made by Espedal and Ewing [18], who proposed a slightly different splitting, taking into account the physically correct balance between diffusive forces and nonlinear advection. In order to reduce the splitting error, a residual or anti-diffusive flux term is constructed based on the wave structure of the solution from the advection step. This residual flux function is then accounted for when solving the diffusion part of the problem, so that correct balance between sharpening and diffusion is obtained.

The idea of a residual flux term was used in [18], and later by Dahle [10] (see also [3, 12, 13, 14, 21]) to simulate two-phase immiscible flow in oil reservoirs. The approach suggested by these authors depends on an *a priori* splitting of the flux function which presupposes solutions of the "well-established" shock-type. That is, their strategy works well in the case of a Riemann problem with solution containing a single shock [10].

Karlsen and Risebro [25] have recently suggested an extension of this approach, referred to as *corrected operator splitting* (COS), capable of solving problems with general flux functions and general initial data. The new aspect is that the residual flux term is no longer based on an *a priori* splitting of the flux function, but rather a dynamically defined splitting that depends on the solution of a certain conservation law (advection problem) that changes from time level to time level. Solutions of the advection problems are computed by the front tracking method [9, 22], which is based on solving Riemann problems so that an exact solution is obtained for a perturbed conservation law. Consequently, construction of *dynamically defined* residual flux terms becomes natural in the context of front tracking.

We emphasize that the main concern in [25] was to present the COS algorithm and to give rigorous convergence proofs for arbitrary non-smooth data. The present paper is concerned with applying a (slightly different) version of the COS approach to realistic flow equations and to compare COS with OS numerically. In addition to front tracking, a $2nd$ order Godunov method has been implemented. The nonlinear diffusion problems are solved by the method of "freezing" coefficients (Picard iteration) [21] and a Petrov-Galerkin finite element method [1]. See [25] for an alternative method yielding linear systems of equations that are symmetric.

Motivated by the work of Holden and Risebro [24] and the more recent work of Lie, Haugse, and Karlsen [31], both on dimensional splitting coupled with front tracking for 2-D scalar conservation laws, we extend the COS method (front tracking) to 2-D equations of the form

$$\partial_t u + \partial_x f(u) + \partial_y g(u) = \varepsilon\big(\partial_x[\nu(u)\partial_x u] + \partial_y[\nu(u)\partial_y u]\big), \qquad \text{for } (x,t) \in \Omega \times (0,T],$$

by means of a similar "alternating-direction" technique.

The remaining part of this paper is organized as follows: In section 2 we introduce a version of the semi-discrete COS method and explain how the residual flux terms are constructed. In section 3 we present two fully discrete COS methods: the first is based on front tracking (COS-F henceforth), while the second is based on a $2nd$ order Godunov method (COS-G henceforth). Section 4 is devoted to a comprehensive numerical investigation of these two methods. Comparisons are made between COS-F(G) and the splitting method based on the modified method of characteristics (MMOC henceforth) [18] whenever possible, and with the corresponding OS schemes [26]. In section 5 we present a 2-D extension of the COS-F method and apply it to a Buckley–Leverett type problem including gravitational effects. Finally, in section 6 we make some concluding remarks.

2. The Corrected Operator Splitting (COS) Approach

In this section we introduce the semi-discrete COS algorithm. For ease of presentation, consider the initial value problem with constant diffusion, i.e.,

$$\partial_t u + \partial_x f(u) = \varepsilon \partial_x^2 u, \qquad u(x,0) = u_0(x). \tag{5}$$

Fix $N \geq 1$ (integer) and choose Δt such that $N\Delta t = T$, for some fixed $T > 0$. Let u^n denote the approximate solution to (5) at time $t = n\Delta t$, for some $n = 0, \dots, N-1$. The splitting solution u^N is defined inductively; let $u^0 = u_0$ and construct u^{n+1} from u^n in the following steps.

Step 1 *(advection):* Let $v(x, \Delta t)$ be the solution at time $t = \Delta t$ to the hyperbolic conservation law

$$\partial_t v + \partial_x f(v) = 0, \qquad v(x,0) = u^n(x). \tag{6}$$

It is well-known that (6) does not possess a smooth solution, not even if the initial function is infinitely smooth. We therefore consider (piecewise smooth) entropy weak solutions to (6) in the sense of Kruzkov [27]. Let $x = y_i$, $i = 1, \dots, m-1$, be the locations of the local extrema of $v(x, \Delta t)$ so that $\langle y_{i-1}, y_i \rangle$, $i = 1, \dots, m$, are intervals[1] on which $v(x, \Delta t)$ is monotone. Here we have set $y_0 = -\infty$ and $y_m = \infty$. We make the convention that the location of a local extremum of a piecewise constant function is the midpoint of the interval on which the extremum is taken. Restricted to the ith monotone interval $\langle y_{i-1}, y_i \rangle$, assume that the discontinuities of $v(x, \Delta t)$ are located at $x = x_{ij}$, $j = 1, \dots, d_i$ (see Fig. 1)[2]. Furthermore, let v_{ij}^- and v_{ij}^+ be the left-hand and right-hand limits taken at these points, i.e.,

$$v_{ij}^- = \lim_{x \to x_{ij}-} v(x, \Delta t), \qquad v_{ij}^+ = \lim_{x \to x_{ij}+} v(x, \Delta t), \qquad j = 1, \dots, d_i.$$

Finally, let v_{min} and v_{max} be the global minimum and maximum of $v(x, \Delta t)$, respectively.

Next, define the "envelope" function $\bar{f}^n : \mathbb{R} \times [v_{min}, v_{max}] \to \mathbb{R}$ (see Fig. 2),

$$\bar{f}^n(x,v) = \begin{cases} f(v_{ij}^-) + \frac{f(v_{ij}^+) - f(v_{ij}^-)}{v_{ij}^+ - v_{ij}^-}(v - v_{ij}^-), & x \in \langle y_{i-1}, y_i \rangle,\ v \in \operatorname{int}\{v_{ij}^-, v_{ij}^+\},\ \text{for some } i, j, \\ f(v), & x \in \langle y_{i-1}, y_i \rangle,\ v \notin \operatorname{int}\{v_{ij}^-, v_{ij}^+\},\ \text{for some } i, j. \end{cases}$$

Moreover, define the residual (or anti-diffusive) flux term $\tilde{f}^n$ (see Figs. 2 and 3) by

$$\tilde{f}^n(x,v) = f(v) - \bar{f}^n(x,v). \tag{7}$$

Observe that the residual flux $\tilde{f}^n(x,v)$ may be discontinuous as a function of x (see Fig. 3).

[1] The notation $\langle y_{i-1}, y_i \rangle$ means the half open interval $(y_{i-1}, y_i]$, if $i = 1, \dots, m-1$, and the open interval (y_{i-1}, y_i), if $i = m$.

[2] When applied to the solution in Fig. 1, the notation from step 1 reads: $m = 4$, $(y_0, y_1] \approx (-\infty, 0.46]$, $d_1 = 1$, $x_{11} \approx 0.46$, $v_{11}^- \approx 0.39$, $v_{11}^+ \approx 0.74$, $(y_1, y_2] \approx (0.46, 0.70]$, $d_2 = 0$, $(y_2, y_3] \approx (0.70, 0.74]$, $d_3 = 1$, $x_{31} \approx 0.74$, $v_{31}^- \approx 0.50$, $v_{31}^+ \approx 0.78$, $(y_3, y_4] \approx (0.74, \infty]$, $d_4 = 1$ and $x_{41} \approx 0.82$, $v_{41}^- \approx 0.71$, $v_{41}^+ \approx 0.0$.

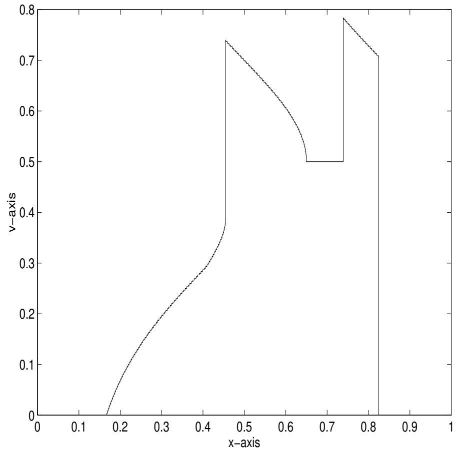

FIGURE 1. A non-monotone advection solution $v(x,\Delta t)$ (step 1).

Step 2 (diffusion): Using the residual flux function (7), let $w(x,\Delta t)$ be the (weak) solution at time $t=\Delta t$ to the parabolic equation

$$\partial_t w + \partial_x \left[\tilde{f}^n(x,w) - \varepsilon \partial_x w\right] = 0, \qquad w(x,0) = v(x,\Delta t). \tag{8}$$

We finally define $u^{n+1}(x) = w(x,\Delta t)$.

Thus the COS approximation at time $t=T$ is formally given by the composition

$$u^N(x) = \left[\mathcal{P}^{\tilde{f}^n}(\Delta t) \circ \mathcal{S}^f(\Delta t)\right]^N u^0(x), \tag{9}$$

where $\mathcal{S}^f(t)$ and $\mathcal{P}^{\tilde{f}^n}(t)$ are the solution operators associated with (6) and (8), respectively.

Several remarks are now in order:

Remark 2.1. Observe that in the presence of shock fronts, the advection part of (8) only has a sharpening effect, whose purpose is to balance the diffusion to the right order. It is instructive to compare COS with OS [26], which is similarly defined, except that the $\tilde{f}^n$-term is missing. Thus the diffusion step for OS consists of solving a heat equation that yields shock layers of size $\mathcal{O}(\sqrt{\varepsilon\Delta t})$ if the time step is large. Accordingly, we do not expect shock fronts to be properly resolved unless $\Delta t = \mathcal{O}(\varepsilon)$. The numerical experiments will demonstrate the importance of the $\tilde{f}^n$-term when the time step is large.

Remark 2.2. In order to obtain a fully discrete method, the numerical schemes replacing the exact operators $\mathcal{S}^f(t)$ and $\mathcal{P}^{\tilde{f}^n}(t)$ have to be carefully chosen (see below). COS is designed so that the greater part of the nonlinearity inherent in (5) should be resolved within the advection steps. Thus, a fully discrete algorithm depends heavily on an accurate numerical scheme for solving nonlinear conservation laws.

Remark 2.3. The splitting described above differs from the one introduced by Espedal and Ewing [18], by the fact that no *a priori* splitting of the flux function is required. However, we note that the splitting ideas suggested in [18] and [25] coincide in the case of

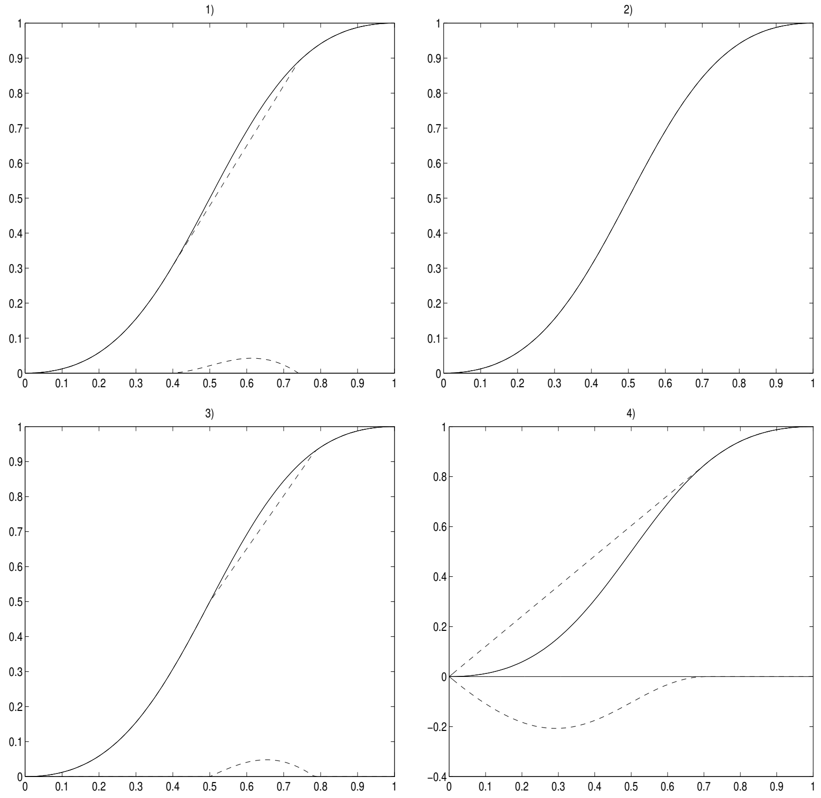

FIGURE 2. We have generated $\bar{f}^n$ and $\tilde{f}^n$ with respect to the solution shown in Fig. 1. Plots $1-4$ show $\bar{f}^n$ and $\tilde{f}^n$ (dashed lines) for: $x \in (-\infty, 0.46]$ (1st), $x \in (0.46, 0.70]$ (2nd), $x \in (0.70, 0.74]$ (3rd), and $x \in (0.74, \infty)$ (4th). The flux function f is drawn with solid line.

a well-established shock, and from this point of view, COS [25] can be considered as the next generalization of the "*a priori* splitting" method. The main new feature is the ability to handle general flux functions and general data in a consistent and systematic way.

3. Two Discrete COS Schemes

We now introduce two fully discrete methods, both based on the COS algorithm.

3.1. **COS-F (Front Tracking).** To begin with, replace equation (1) with a perturbed equation where the flux function (denoted by f_δ) is a piecewise linear and continuous approximation to the flux f in (1). The approximation f_δ is chosen simply as the linear interpolant of f at points $\{i\delta\}$. In the numerical experiments reported later, the number of interpolation points ($1/\delta$) will be the same as the number of spatial grid points (nodes). Assume that we have computed a piecewise linear approximation u^n to the solution of the initial-boundary value problem (1) at time $t = n\Delta t$. We next describe the steps leading to the approximation at time $t = (n+1)\Delta t$, u^{n+1}.

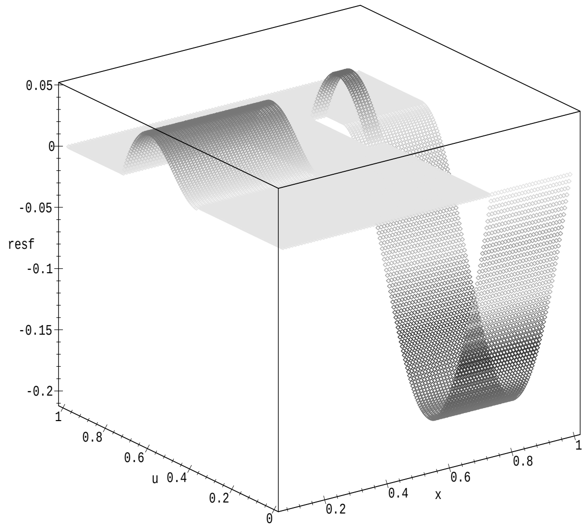

FIGURE 3. The residual flux term $\tilde{f}^n = \tilde{f}^n(x,u)$ (see Fig. 2) viewed as a function of x and u. Note that the residual flux depends discontinuously on the spatial variable, which is due to the non-monotone advection solution.

Step 1 *(advection):* We use front tracking to compute solutions to the (perturbed) advection equation

$$\partial_t v + \partial_x f_\delta(v) = 0, \qquad v(x,0) = v_0(x), \qquad x \in (a,b). \tag{10}$$

We assume that both the initial data v_0 and the solution $v(x,\Delta t)$ are consistent with the boundary conditions. Front tracking is based on replacing the initial data by a piecewise constant approximation, and to solve the resulting problem *exactly*. This is done by:

1. Solving all the Riemann problems defined by the piecewise constant initial data. This defines a sequence of discontinuities emanating from the origin of each problem, and between these discontinuities the solution is constant. The discontinuities defines the exact solution until some of them interact (collide).
2. Keeping track of collisions of neighbouring discontinuities. When discontinuities from neighbouring Riemann problems collide, they define a new Riemann problem with left and right states given by the values immediately to the left and right of the collision. The new problem is solved and the "tracking" continued, thus advancing the exact solution forward in time, until we reach final computing time ($t = \Delta t$).

The front tracking method was first defined by Dafermos [9] and later developed into a numerical method by Holden, Holden, and Høegh–Krohn [22]. In [22] it is proved that the solution remains piecewise constant for all time. Rarefaction waves are replaced by series of small ($\mathcal{O}(\delta)$ - sized) shocks. Observe that the δ - parameter determines the accuracy of the front tracking scheme in smooth parts of the (advection) flow. To obtain a step function approximation to the initial data, we have used a grid cell averaging operator defined on a uniform grid $\{\tilde{x}_i; \Delta x\}$, where Δx is a small discretization parameter controlling the spatial accuracy of the scheme, i.e.,

$$v_0(x) = \frac{1}{\Delta x} \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} u^n(\tilde{x})\, d\tilde{x}, \qquad \forall x \in [\tilde{x}_i, \tilde{x}_{i+1}).$$

Front tracking preserves the discontinuities in the true solution of (10) in a non-diffusive manner, which means that the residual flux term, denoted by $\tilde{f}_\delta^n$, is found as described in the semi-discrete algorithm (7). Moreover, since the front tracking solution is a step function taking a finite number of values, formula (7) is straightforward to realize.

Step 2 (diffusion): The nonlinear (perturbed) diffusion equation is linearized by the method of “freezing” the coefficients. This (Picard) iteration yields a sequence of linear equations, which subsequently are solved by the Petrov-Galerkin finite element method (see Appendix A). To be more precise, let $\{w^p(x, \Delta t)\}_{p=1}^\infty$ be the Petrov-Galerkin approximations at time $t = \Delta t$ to the linear variable-coefficient advection-diffusion equations

$$\partial_t w^p + \partial_x \left[b_\delta^n(x, w^{p-1}) w^p - \varepsilon \nu(w^{p-1}) \partial_x w^p \right] = 0, \qquad p = 1, 2, \dots, \tag{11}$$

with initial and boundary data: $w^p(x,0) = v(x, \Delta t)$, $w^p(a,t) = u_a$, and $w^p(b,t) = u_b$. In (11), we have for convenience rewritten $\tilde{f}_\delta^n(x, w)$ as $b_\delta^n(x, w)w$. The COS-F solution at time $t = (n+1)\Delta t$ is now taken to be $u^{n+1} = w^q$ for some q. The element approximation is a piecewise linear function on a grid with nodes $\{x_i\}$ chosen so that they coincide with the discontinuity points of the front tracking solution $v(x, \Delta t)$. In order to ensure convergence of our method, we have to add nodes whenever the spacing between two discontinuities becomes larger than Δx. Note that this results in a grid well-suited for resolving shock layers, in particular when the time step is large, i.e., when the front tracking solution contains “strong” shocks.

For later use, the scheme obtained by ignoring the b_δ^n-term in (11), which is more or less the one analysed in [26], will be denoted by OS-F. Finally, see [21] for a discussion of the convergence properties of the Picard iteration.

Remark 3.1. In the implementation of COS-F, the initial data for the Petrov-Galerkin solver is a piecewise constant function. Alternatively, accuracy could have been slightly improved by using piecewise linear data, obtained by a suitable linear interpolation (in smooth regions) of the front tracking solution.

Remark 3.2. Front tracking fits naturally into the COS concept primarily because the method provides “exact” information about the wave structure of the advection solution, thereby making it possible to find the residual flux term (7). In fact, this scheme constitutes an important part of the semi-discrete COS algorithm as it is defined and analysed in [25]. Front tracking is superior to other known methods for 1-D conservation laws when it comes to accuracy and runtime. On the other hand, it has the disadvantage of being nontrivial to

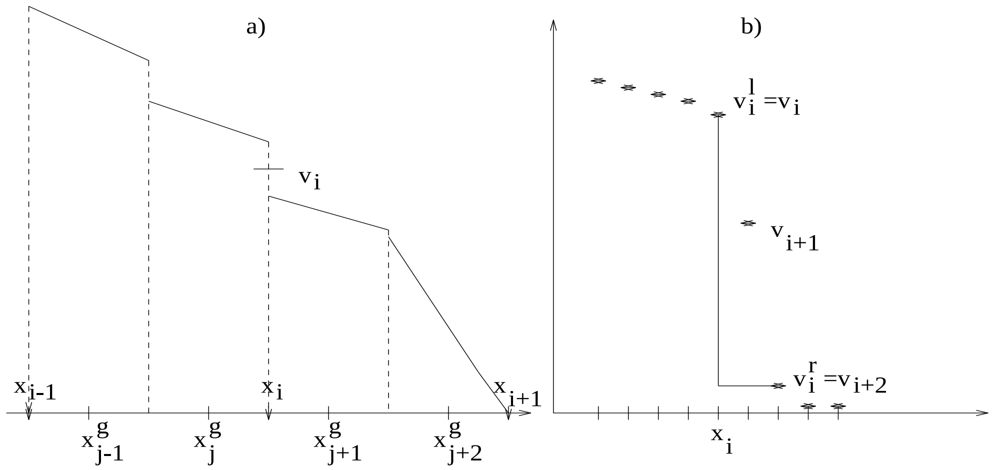

FIGURE 4. a) Based on the piecewise linear Godunov solution, we compute constant states $\{v_i\}$, which are used to define the residual flux term (here $M^g = 2$). b) Illustrates the method for finding the "shock" states $\{v_i^l, v_i^r\}$.

implement [28, 32] compared with difference schemes. This has motivated the application of a $2nd$ order Godunov (difference) method in the manner explained below.

3.2. **COS-G (A Godunov Method).** As before, let u^n denote the piecewise linear approximation at time $t = n\Delta t$. Then u^{n+1} is constructed as follows.

Step 1 *(advection):* Without going into details describing the $2nd$ order Godunov (slope limiter) method used here, we simply mention that the Godunov solution to (6), also denoted by $v(x, \Delta t)$, is a (discontinuous) piecewise linear function whose construction is explained in e.g. [30, p. 188]. The Godunov method needs a certain (fine) grid on which $v(x, \Delta t)$ is piecewise linear. Let therefore $\{x_j^g; \Delta x^g\}$ denote a uniform subdivision of $[a, b]$, and let $\Delta t^g = \Delta t / N^g$, where N^g is the smallest integer so that the CFL condition is satisfied. Here, $\Delta x^g < \Delta x$ and $\Delta t^g < \Delta t$ are small numbers specifying the accuracy of the Godunov scheme. The fine grid $\{x_j^g; \Delta x^g\}$ is related to the finite element grid $\{x_i; \Delta x\}$ by letting $x_{j+1/2}^g$ coincide with x_i for some j, and by dividing each element $[x_i, x_{i+1}]$ into M^g smaller elements so that $M^g \Delta x^g = \Delta x$; see Fig. 4a.

We restrict ourselves to a fixed monotonicity interval and explain how the (local) envelope function on this interval is found. The global envelope function $\bar{f}^n$, and thus the residual flux term $\tilde{f}^n$, is found by "gluing" together the local ones. Define the set $\{v_i\}$ by $v_i = (v(x_j^g, \Delta t) + v(x_{j+1}^g, \Delta t))/2$; see Fig. 4a. Let $\eta > 0$ be a predefined value (problem dependent) and set $\Delta v_i = |v_{i+1} - v_i|$. Suppose that $\Delta v_i > \eta$, which should indicate that a shock is located somewhere in the neighbourhood of the ith element $[x_i, x_{i+1})$, in which case we define $v_i^l = v_i$. Let $j \geq i + 1$ be the least integer so that $\Delta v_j \leq \eta$, and set $v_i^r = v_j$ Next, introduce the set $\{\bar{v}_j\}$ consisting of $\{v_i : \Delta v_i \leq \eta\}$ and the left and right (shock) states $\{v_i^l, v_i^r\}$; see Fig. 4b. The envelope function is found simply as the linear interpolant of f at points $\{\bar{v}_j\}$. Finally, as explained before, more than one envelope function is obtained if monotonicity changes.

Step 2 (diffusion): Coincides with step 2 in the COS-F algorithm, except that the finite element nodes $\{x_i; \Delta x\}$ are uniformly distributed, instead of being aligned with the discontinuities.

For later use, the scheme denoted by OS-G is obtained by ignoring the residual term present in COS-G.

4. Numerical Experiments

The discrete (C)OS methods will be compared with respect to both accuracy and efficiency. In particular, several plots are presented in order to demonstrate, among other things, the ability to handle sharp shock fronts, build-up of shock fronts, and general non-monotone initial data. Moreover, COS-G(F) is compared whenever possible with MMOC [18] (the version implemented in [14]) and always with OS-G(F).

We will also make an attempt to extract some information indicating the (asymptotic) rate at which OS-F converges. Note that OS and COS coincide when Δt becomes small, i.e., when the advection solutions do not contain shocks. We use OS-F instead of OS-G to obtain convergence rates, since their performance are almost equal with respect to accuracy, but with OS-F being notably faster than OS-G. In order to determine the rate of convergence, we shall measure the *error* at a fixed time T. The convergence results in [25, 26] were formulated in L_1. Therefore, we measure the error in a relative L_1-norm, i.e.,

$$E = \frac{||u^N - u_*(\cdot, T)||_1}{||u_*(\cdot, T)||_1},$$

where u_* is some reference solution. Since no exact solution is available, we have used a reference solution calculated by a standard difference scheme with a very fine discretization Let $h > 0$ be a small number, $\delta = \Delta x = h$ and $\Delta t = (\mathrm{CFL}/\|f'\|_\infty)h$. Here CFL denotes the usual Courant-Friedrichs-Lewy number, which will be specified later. We assume the OS-F error to be of the form $E = \mathcal{O}(h^\alpha)$. To determine the convergence rate α, we perform computations with $h = 2^{-4}, \dots, 2^{-9}$. Standard linear regression is used to determine α.

In the first four examples presented below we use flux function f given by (3) and constant diffusion coefficient $\nu(u) = 1$. In Example 5 we include gravitational effects in the flux function (see Fig. 5), namely,

$$f(u) = \frac{u^2}{u^2 + (1-u)^2}(1 - 5(1-u)^2). \tag{12}$$

In Example 2 we also present some results using the nonlinear diffusion coefficient (4).

All solutions are computed on $[a, b] = [0, 1]$ up to time $T = 0.2$ with $\varepsilon = 0.01$. Moreover, boundary values are chosen consistent with the initial data. All approximate solutions (and initial data) depicted on the next pages are computed on a grid consisting of 100 nodes, i.e. $\Delta x = \mathcal{O}(\varepsilon)$, with N time steps and 5 Picard iterations, unless otherwise stated. Moreover, the solutions (data) are shown as piecewise linear functions.

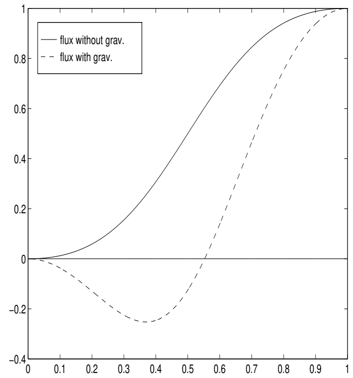

FIGURE 5. Flux function f given by (3) without gravitation, and (12) with gravitation.

4.1. **Example 1.** The first example is the so-called well-established shock case,

$$u_0(x) = \begin{cases} 1 - x, & 0 \le x \le 1 - \frac{1}{\sqrt{2}}, \\ 0, & 1 - \frac{1}{\sqrt{2}} < x \le 1. \end{cases}$$

See Fig. 6 for plots of approximate solutions generated by (C)OS-F, (C)OS-G, and MMOC. We present "log-log plots" of E versus h, E versus Δt, and E versus runtime in Fig. 7a, b, and c, respectively. Fig. 7c is generated by keeping $\Delta x = 2^{-11}$ fixed and letting $\Delta t = 0.2N^{-1}$, $N = 1, 2, 4, 8, 16, 32, 64$. The straight lines in Fig. 7 are regression lines obtained from least-squares fit. The (C)OS-G fine grid consists of 200 nodes; $M^g = 2$.

In order to emphasize on the effect of the residual flux terms, we compare OS-F and COS-F when $N = 1$ ($\Delta t = 0.2$); see Fig. 6a. We have already pointed out that time steps of size $\mathcal{O}(\varepsilon)$ should be used by OS schemes whenever shock fronts are present in the true solution. We see that this is indeed the case with OS-F, which produces a shock layer that is too much "smeared out" when $N = 1$. The COS-F method, however, obtains the correct balance between advection and diffusion, i.e., an $\mathcal{O}(\varepsilon)$ - shock layer. To obtain correct structure of the shock front with OS-F, approximately $15 - 20$ time steps have to be used; see Fig. 6b. Information on the effect of the residual flux terms can also be read from Fig 7b, which shows $\log(E)$ versus $\log(\Delta t)$ for (C)OS-F. We see that the temporal error is significantly smaller for COS-F than OS-F for large values of Δt, and that these methods begin to coincide, as expected, when the number of time steps reach $32 - 64$. Furthermore, with basis in Fig. 7b it is fair to say that COS-F is more or less independent of the size of the time step for this travelling "quasi-steady state" solution.

We see that COS-F, COS-G, and MMOC produce almost equal solutions in the "visual norm"; see Fig. 6a and c. However, for these "100 nodes" calculations, COS-F is notably faster than both COS-G and MMOC; the runtime for COS-F was 0.01 sec compared with 1.81 sec for COS-G, and 0.43 sec for MMOC. Very crude runtime estimates are also presented in Fig. 7c, which essentially say that COS-F is the most efficient implementation.

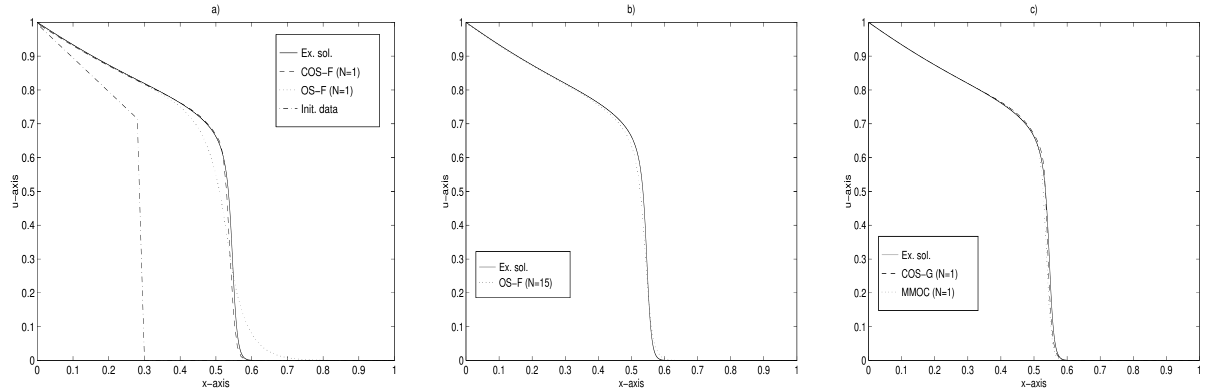

FIGURE 6. Example 1. Plots a) - c) show that COS-F(G) and MMOC need one time step to resolve the shock front, and that OS-F needs 15 time steps to do so. Runtime for: a) COS-F 0.01 sec. c) COS-G 1.81 sec, MMOC 0.43 sec.

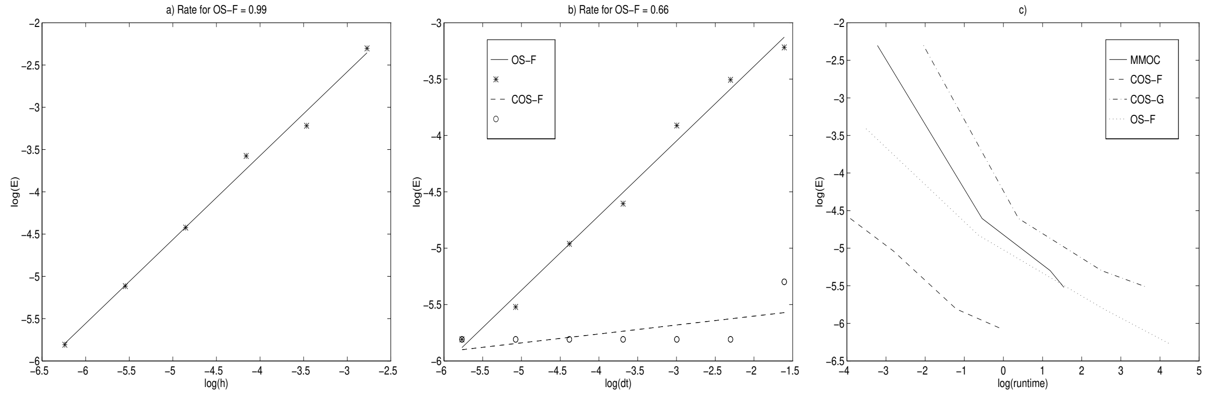

FIGURE 7. Example 1. "log-log plots" of: a) E vs h for OS-F (CFL = 3.2). b) E vs Δt for (C)OS-F. c) E vs runtime for MMOC, (C)OS-F(G).

Fig. 7a indicates that the asymptotic convergence rate[3] is linear with respect to h when both discretization parameters Δx and Δt are of order h. On the other hand, Fig. 7b seems to indicate that the temporal (splitting) error does not decrease linearly in Δt for large values of Δt (and a fixed small Δx). Based on a fit of data from $N = 1, \dots, 64$ - computations, we obtain the rate 0.66, which, more or less, coincides with the observations reported in [26]. We believe that this low convergence rate reflects the fact that the advection solutions are not sufficiently regular for large Δt, which supports, once more, our theme that one should not take too large time steps in OS schemes when the true solution contains shock fronts.

[3]One expects the error for OS-F(G) [15, 17] to behave like $E = \mathcal{O}(\Delta x^2 + \Delta t)$, at least when $\Delta x, \Delta t \to 0$. In experiments (not reported here) we have tested the "rate in Δx". These tests have shown the rate to be reasonably close to 2.

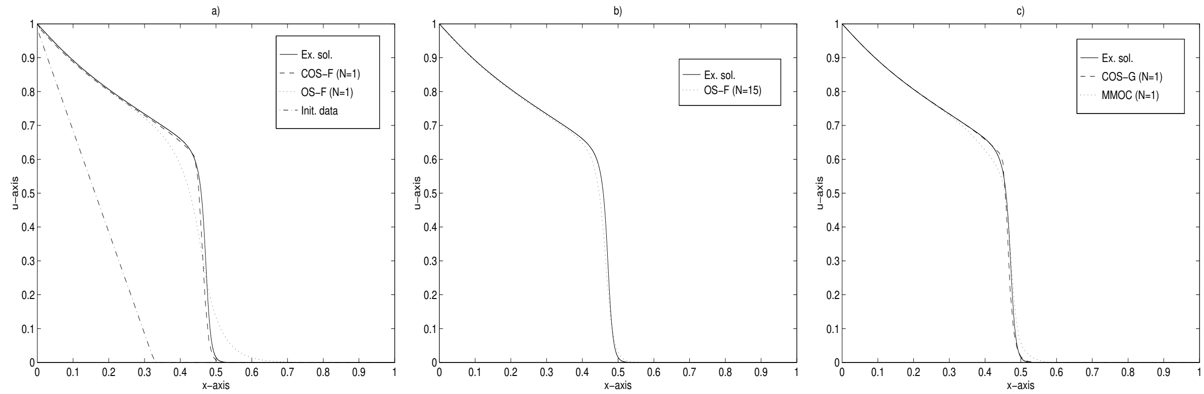

FIGURE 8. Example 2. Plots a) - c) show that COS-F(G) and MMOC need one time step to resolve the shock front, while OS-F needs more than 15 time steps to do so. Runtime for: a) COS-F 0.02 sec. c) COS-G 1.83 sec, MMOC 0.63 sec.

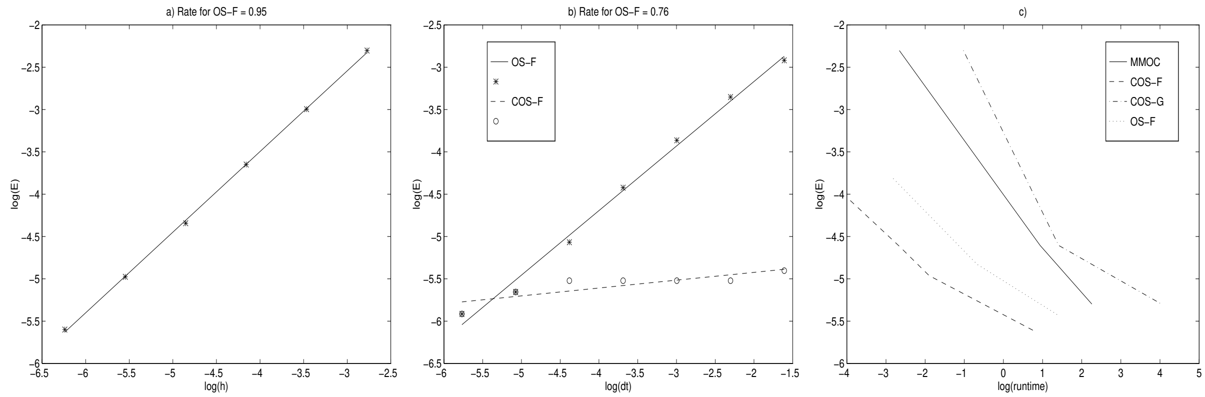

FIGURE 9. Example 2. "log-log plots" of: a) E vs h for OS-F (CFL = 3.2). b) E vs Δt for (C)OS-F. c) E vs runtime for MMOC, (C)OS-F(G).

4.2. **Example 2.** The second example is slightly more difficult in the sense that it involves the build-up of a shock front. The initial function is

$$u_0(x) = \begin{cases} 1 - 3x, & 0 \leq x \leq \frac{1}{3}, \\ 0, & \frac{1}{3} < x \leq 1. \end{cases}$$

The (C)OS-G fine grid is given by $M^g = 2$. Various data related to this example are presented in Figs. 8 and 9.

The preceding discussion (Example 1) carries more or less over to the present example. We see indeed that the COS schemes resolve the shock front within one time step, whereas OS-F needs more than 15 time steps to do so; see Fig. 8a and b. Furthermore, Fig. 8c shows that MMOC performs much better than OS-F, but not as well as COS-F(G). The reason is that the *a priori* splitting of the flux function is not consistent with the advection solution, as opposed to the previous example, and in some sense this is one of the simplest

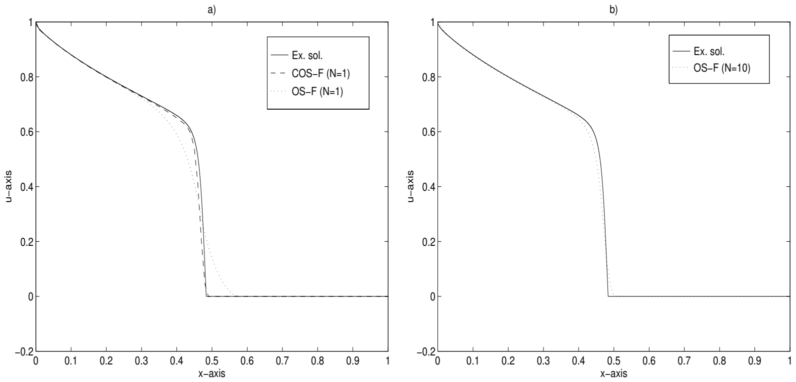

FIGURE 10. Example 2: (C)OS-F with a nonlinear (degenerate) diffusion coefficient. Note the "regularity loss" at the foot of the front.

examples in which an *a priori* splitting of the flux is not "optimal" and the dynamical splitting seems preferable.

In terms of runtime, COS-F is, as before, the scheme to prefer; see Fig. 9c for very crude runtime estimates.

Finally, according to Fig. 9a, the error decreases linearly in h for OS-F, and the decrease is slower (than linear) in Δt for large values of Δt; see Fig. 9b. These observations agree with those from Example 1.

To demonstrate that nonlinear diffusion coefficients causes no difficulties, let us consider the present problem, but with diffusion coefficient $\nu(u)$ given by (4). Solutions generated by (C)OS-F are shown in Fig. 10. The slight undershoot seen in Fig. 10 is of order 10^{-5}. Due to the degenerate behaviour at the foot of the front, we see that the true solution is less regular than in the constant diffusion (non-degenerate) case. This feature is captured by COS-F using one time step and 10 Picard iterations.

4.3. **Example 3.** This example is designed to explore the COS schemes potential for handling non-monotone data. Let therefore

$$u_0(x) = \begin{cases} 0, & 0 \leq x < \frac{1}{6}, \\ 12(x - \frac{1}{6}), & \frac{1}{6} < x \leq \frac{1}{4}, \\ 1, & \frac{1}{4} \leq x \leq \frac{1}{2}, \\ 1 - 12(x - \frac{1}{2}), & \frac{1}{2} < x < \frac{7}{12}, \\ 0, & \frac{7}{12} \leq x \leq 1. \end{cases}$$

Note that u_0 has two intervals of monotonicity. The (C)OS-G fine grid consists now of 300 nodes ($M^g = 3$). Various data are presented in Figs. 11 and 12. This is the first example on which schemes based on *a priori* splitting of the flux function (MMOC) do not apply. We also mention that the solution is smooth on the larger part of the domain, with shock layers forming around 0.48 and 0.85; see Fig. 11a.

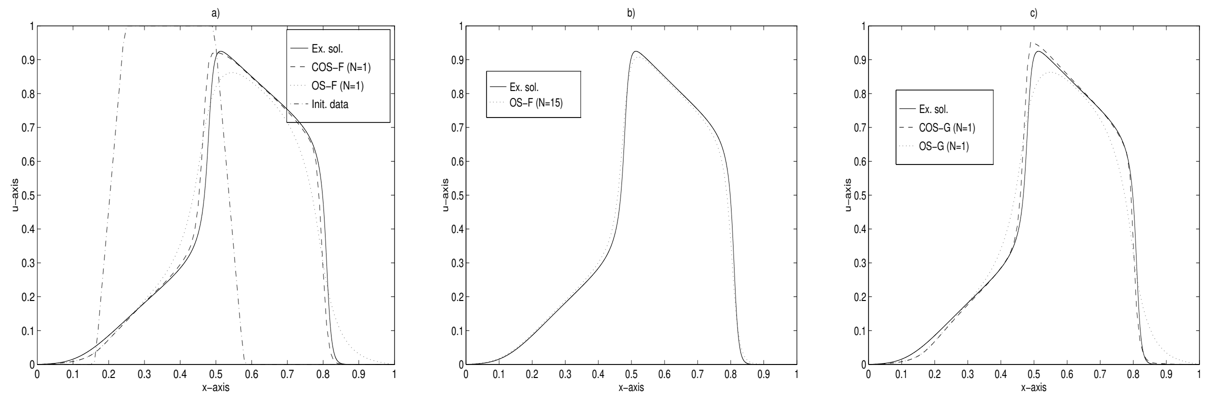

FIGURE 11. Example 3. Non-monotone data. Runtime for: a) COS-F 0.02 sec. b) OS-F 0.15 sec. c) COS-G 2.89 sec.

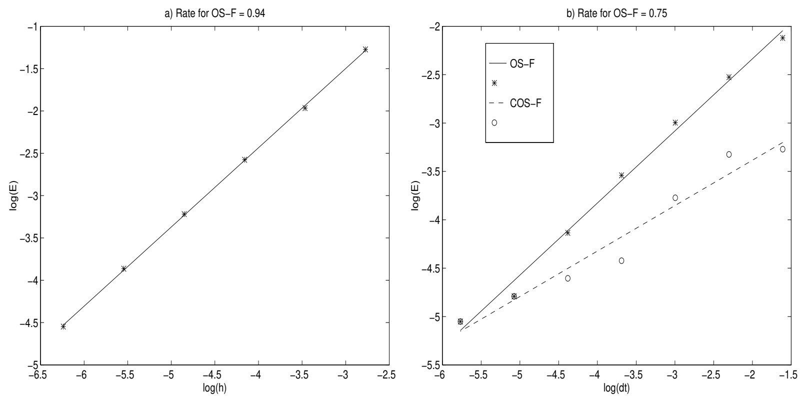

FIGURE 12. Example 3. "log-log plots" of: a) E vs h for OS-F (CFL = 6.4). b) E vs Δt for (C)OS-F.

Also here we see that both COS schemes perform reasonably well within one time step; see Fig. 11a and c, and that OS-F(G) requires 10–15 time steps to obtain the same degree of accuracy; see Fig. 11b. Moreover, Fig. 12b tells us that OS-F and COS-F more or less coincide when the number of time steps reaches 32.

When it comes to achieving a certain degree of accuracy, COS-F needs significantly less runtime than COS-G. Furthermore, COS-F is roughly $3-5$ times faster than OS-F for a fixed degree of accuracy.

Fig. 12a confirms that the error decreases linearly in h for OS-F, while Fig. 12b shows that the splitting error decreases like $\mathcal{O}(\Delta t^{0.75})$, at least when Δt varies in range $0.2/1, \ldots, 0.2/64$.

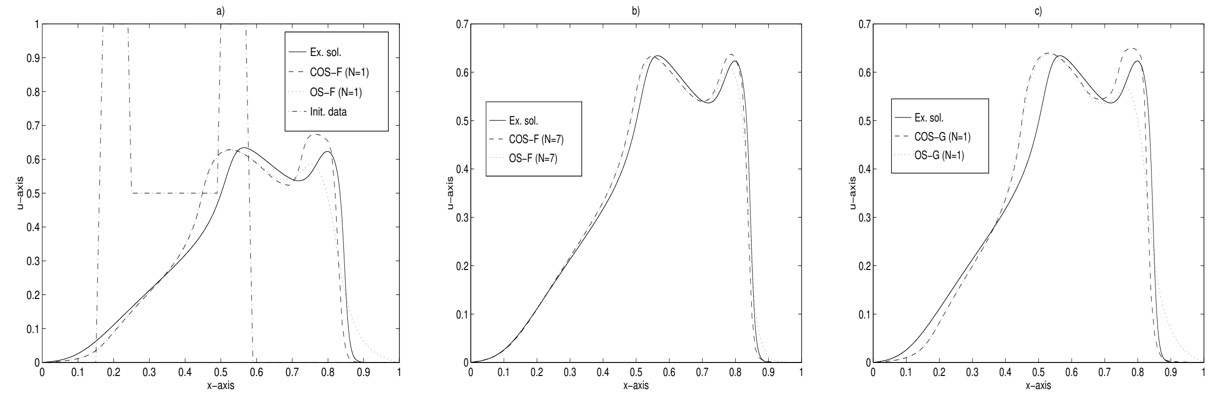

FIGURE 13. Example 4. "Highly" non-monotone data. Note that the two (large) waves interact during the time period $[0, 0.2]$. Runtime for: a) COS-F 0.04 sec. c) COS-G 9.3 sec.

4.4. **Example 4.** We now complicate the situation further by considering an initial function with four intervals of monotonicity,

$$u_0(x) = \begin{cases} 0, & 0 \leq x < \frac{1}{6}, \\ 1, & \frac{1}{6} < x \leq \frac{1}{4}, \\ \frac{1}{2}, & \frac{1}{4} \leq x \leq \frac{1}{2}, \\ 1, & \frac{1}{2} < x < \frac{7}{12}, \\ 0, & \frac{7}{12} \leq x \leq 1. \end{cases}$$

The (C)OS-G fine grid now consists of 600 nodes ($M^g = 6$). Various results are presented in Fig. 13. This and two more examples are included in order to demonstrate the COS schemes' potential for handling general initial data and flux functions. We do not present "log-log" plots and convergence rates for these examples.

Again the true solution is smooth (with slowly varying gradients) everywhere, except in the interval $[0.8, 0.9]$, where a shock front has formed. One cannot expect COS-F(G) (or any other scheme) to resolve the true solution within one or a few time steps, which has been the case in the previous examples. The reason is that the "dynamics" of the problem appears on time scales notably less than 0.2. This is evident from Figs. 13a and c, which also show that the least impressive behaviour of COS-F(G) is found in smooth parts of the flow. It seems that COS-F needs 5–7 time steps to produce reasonable accurate solutions; see Fig. 13b.

4.5. **Example 5 (Gravitational Effects).** We now consider a flux function f including gravitational effects (12), and initial data of the form

$$u_0(x) = \begin{cases} 0, & 0 \leq x < 1 - \frac{1}{\sqrt{2}}, \\ 1, & 1 - \frac{1}{\sqrt{2}} \leq x \leq 1. \end{cases}$$

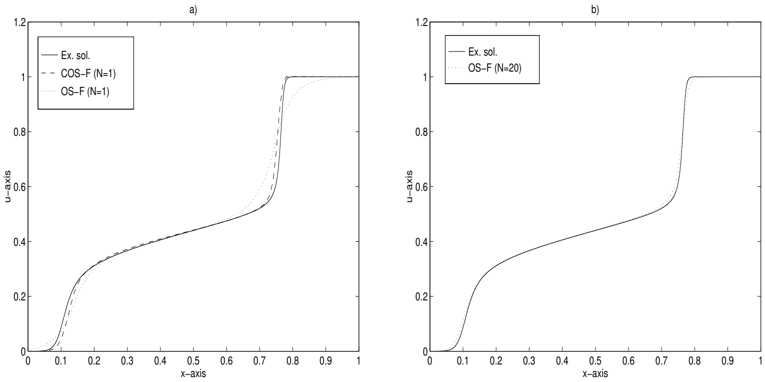

FIGURE 14. Example 5. Flux function with gravitation (12). COS-F resolve the true solution within one time step, while OS-F needs 15-20 time steps to do so. Runtime for: a) COS-F 0.02 sec. b) OS-F 0.16 sec.

The results can be found in Fig. 14. For this example we have only generated solutions with (C)OS-F, since the Godunov method implemented here does not handle a flux f where f' changes sign.

We see again that COS-F captures the correct structure of the solution using $N = 1$ (10 Picard iterations), compared with $N = 20$ required by OS-F. The slight overshoot seen in Fig. 14 is of order 10^{-3}. Keeping the number of iterations fixed at 10, and taking several (two or more) Euler steps, that is, several local time steps in the approximation of the time derivative in the (residual) diffusion equation, this overshoot can be greatly reduced, or even removed.

We have also computed solutions with Riemann data 1 and 0 to the left and right of $1 - 1/\sqrt{2}$, respectively. The quality of the approximations (not shown with plots) is the same as before, but the method of "freezing" the coefficients now needs approximately 20 iterations before it converges. The reason for the increased number of iterations is that the residual flux terms are more dominant than before, due to gravitation.

5. An Extension to 2-D Problems

Introduce the notation $\mathbf{f}(u) = (f(u), g(u))$ and $\nabla = (\partial_x, \partial_y)$. We shall in what follows be concerned with solving the 2-D problem

$$\partial_t u + \nabla \cdot \mathbf{f}(u) = \varepsilon \nabla \cdot \big[\nu(u)\nabla u\big], \qquad \text{in } \Omega \times (0, T], \tag{13}$$

$$u|_{t=0} = u_0, \qquad u|_{\partial\Omega} = c, \tag{14}$$

where Ω is an open bounded subset of $\mathbb{R}^2$, and c is a constant. For convenience we choose $c = 0$ and $\nu(u) = 1$.

Lie *et al.* [31] have recently observed that for 2-D scalar conservation laws, a particular dimensional splitting method based on front tracking is highly efficient due to the lack of a CFL condition. They report that CFL numbers typically in the range 5–15 can be used for many problems without loss of accuracy, thereby making the method very fast. The COS scheme based on front tracking (COS-F) also lacks a time step restriction, and large time steps are feasible without much loss of accuracy, as opposed to the OS-F scheme. This has motivated us to extend the COS-F algorithm to the initial-boundary value problem (13) using a similar alternating-direction technique.

Assume that the domain Ω is rectangular and consider a uniform Cartesian grid $\{z_{i,j}; \Delta x, \Delta y\}$, where each grid cell is of the form

$$z_{i,j} = \{(x, y) : x_i \leq x < x_{i+1} \text{ and } y_j \leq y < y_{j+1}\},$$

where Δx and $\Delta y > 0$ define the size of each 2-D grid cell. Let π denote the two-dimensional projection operator defined on $\{z_{i,j}\}$, i.e.,

$$(\pi u)(x, y) = \frac{1}{\Delta x \Delta y} \int_{z_{i,j}} u(\tilde{x}, \tilde{y})\, d\tilde{x}\, d\tilde{y}, \qquad \forall (x, y) \in z_{i,j}.$$

Moreover, let f_δ and g_δ be the piecewise linear and continuous approximations to f and g, respectively. Let u^n denote the piecewise constant splitting solution at time $t = n\Delta t$. We now proceed by explaining how to construct u^{n+1} from u^n.

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

$$\partial_t v + \partial_x f_\delta(v) = 0, \qquad v(x, 0; y) = (\pi u^n)(x; y). \tag{15}$$

Note that y acts only as a parameter in (15). Next construct the residual flux function $\tilde{f}_\delta^n(x, v; y)$ with respect to the constant values taken by $v(x, \Delta t; y)$. Let $w(x, \Delta t; y)$ be the solution at time $t = \Delta t$ to the parabolic equation

$$\partial_t w + \partial_x \left[\tilde{f}_\delta^n(x, w; y) - \varepsilon \partial_x w\right] = 0, \qquad w(x, 0; y) = v(x, \Delta t; y), \tag{16}$$

with boundary data $w|_{\partial\Omega} = 0$. The solution is obtained by "freezing" the coefficients and using the (1-D) Petrov-Galerkin method on a grid with nodes determined by the discontinuities of the front tracking solution $v(x, \Delta t; y)$.

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

$$\partial_t v + \partial_y g_\delta(v) = 0, \qquad v(y, 0; x) = (\pi w(\cdot, \Delta t; \cdot))(y; x). \tag{17}$$

Note that x acts as a parameter in (17). Furthermore, let the residual term $\tilde{g}_\delta^n(y, v; x)$ be given by the constant values taken by $v(y, \Delta t; x)$, and let $w(y, \Delta t; x)$ be the Petrov-Galerkin solution at time $t = \Delta t$ to the parabolic equation

$$\partial_t w + \partial_y \left[\tilde{g}_\delta^n(y, w; x) - \varepsilon \partial_y w\right] = 0, \qquad w(y, 0; x) = v(y, \Delta t; x), \tag{18}$$

with boundary data $w|_{\partial\Omega} = 0$. Then the solution at time $t = (n+1)\Delta t$ is defined as $u^{n+1} = \pi w(\cdot, \Delta t; \cdot)$.

Thus the operator splitting solution at time $t = T$ is formally given by the composition

$$u^N(x, y) = \left[\pi \circ \mathcal{P}_\Delta^{g_\delta^n}(\Delta t) \circ \mathcal{S}^{g_\delta}(\Delta t) \circ \pi \circ \mathcal{P}_\Delta^{f_\delta^n}(\Delta t) \circ \mathcal{S}^{f_\delta}(\Delta t)\right]^N u^0(x, y), \tag{19}$$

where $u^0 = \pi u_0$, and $\mathcal{S}^{f_\delta}(t)$ and $\mathcal{S}^{g_\delta}(t)$ are the 1-D solution operators associated with (15) and (17), respectively. Furthermore, $\mathcal{P}_\Delta^{f_\delta^n}(t)$ and $\mathcal{P}_\Delta^{g_\delta^n}(t)$ are the approximate solution operator associated with (16) and (18), respectively, where $\Delta = (\Delta x, \Delta y)$. Note that u^N

is piecewise constant with respect to $\{z_{i,j}\}$. However, in applications we replace u^N by a proper piecewise linear function in order to obtain higher accuracy in space. When the solution is highly non-monotone it may be necessary to use local time stepping, that is, we replace (19) by

$$u^N(x,y) = \left[\left[\pi \circ \mathcal{P}_\Delta^{g_\delta^n}(\Delta t_y) \circ \mathcal{S}^{g_\delta}(\Delta t_y)\right]^{N_y} \circ \left[\pi \circ \mathcal{P}_\Delta^{f_\delta^n}(\Delta t_x) \circ \mathcal{S}^{f_\delta}(\Delta t_x)\right]^{N_x}\right]^N u^0(x,y),$$

where $N_x \Delta t_x = N_y \Delta t_y = \Delta t$. This is best utilized if the size of the time steps needed to resolve the dynamics is different in the two directions, due to a different degree of nonlinearity or monotonicity.

5.1. **A Numerical Example.** Consider (13) with flux functions of the form

$$f(u) = u^2/(u^2 + (1-u)^2),$$
$$g(u) = f(u)\big(1 - 5(1-u)^2\big),$$

and initial data

$$u_0(x,y) = \begin{cases} 1, & \text{for } x^2 + y^2 < 0.5 \\ 0, & \text{otherwise.} \end{cases}$$

The solution is computed on the domain $[-1.5, 1.5] \times [-1.5, 1.5]$ up to time $T = 0.5$. We use $\delta = 0.01$ for the flux approximations. The reference solution is computed with standard operator splitting (OS-F) using a $1600 * 1600$ grid and 441 time steps (CFL $= 2.0$).

Note that the above model includes gravitational effects in the y-direction, and that we therefore will need more Picard iterations (in that direction), cf. Example 5. In the present simulation we used up 10 iterations. However, the strong waves are not interacting, and we do not need to include local steps in either direction; $N_x = N_y = 1$.

Fig. 15a shows a contour plot of the solution obtained by OS-F using 5 time steps. The shock layer, but also the rarefaction area, is too wide. Note also the presence of a small artificial, vertical shock layer on the left-hand side of the peak. This is a result of the dimensional splitting, which is not able to resolve (completely) the dynamics of the problem. In Fig. 15b we have used 10 time steps. The artificial shock layer has now (nearly) disappeared, and the resolution of the physical shock layers is slightly improved.

Fig. 16a shows the solution obtained by the corrected operator splitting (COS-F) using 5 time steps. The shock layer is of correct size, but as in Fig. 15a the artificial shock layer is present. In Fig. 16b the number of time steps has been doubled, and the solution is now in good correspondence with the reference solution. For equal discretization parameters, the runtime for COS-F is three times as much as for OS-F, but COS-F gives a much sharper resolution of the shock layer. Trying to obtain an equally good resolution with the OS-F method, the runtime was increased beyond that of COS-F. However, comparing runtimes is a bit futile, since COS-F has not been optimized with respect to runtime.

Fig. 17 shows a 3-D plot of the solution obtained by COS-F and OS-F on a 100*100 grid. In addition we have included the solution obtained by a finite difference splitting

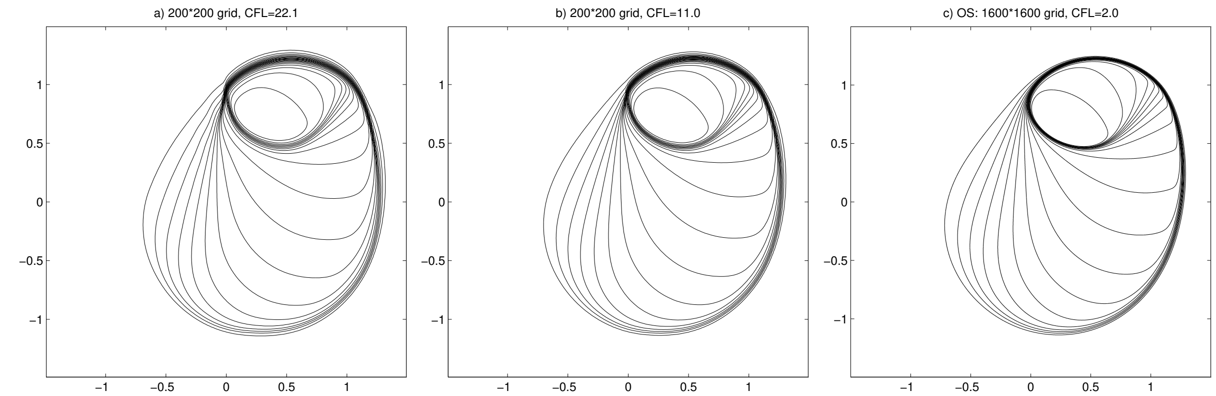

FIGURE 15. 2-D example. a) OS-F: 200*200 grid, CFL = 22.1. b) OS-F: 200*200 grid, CFL = 11.0. c) Reference solution on a 1600*1600 grid, CFL = 2.0.

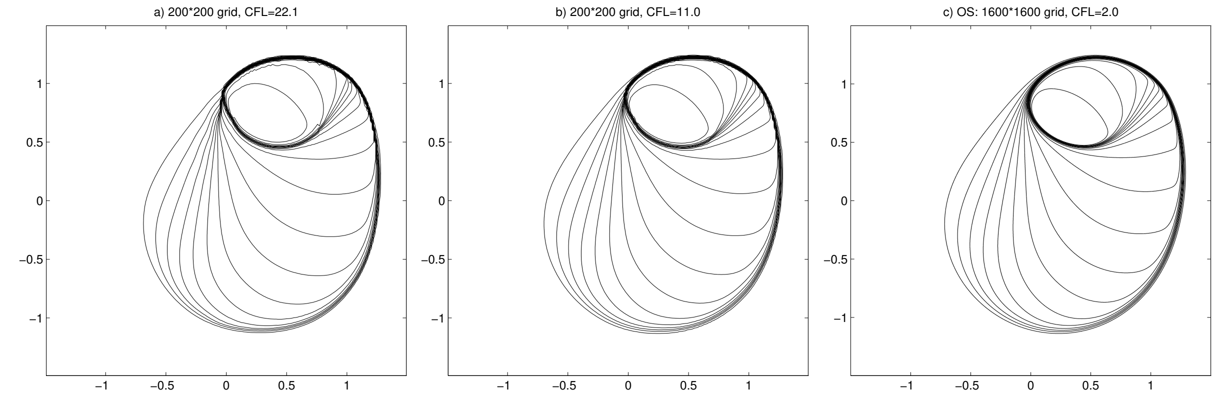

FIGURE 16. 2-D example. a) COS-F: 200*200 grid, CFL = 22.1. b) COS-F: 200*200 grid, CFL = 11.0. c) Reference solution on a 1600*1600 grid, CFL = 2.0.

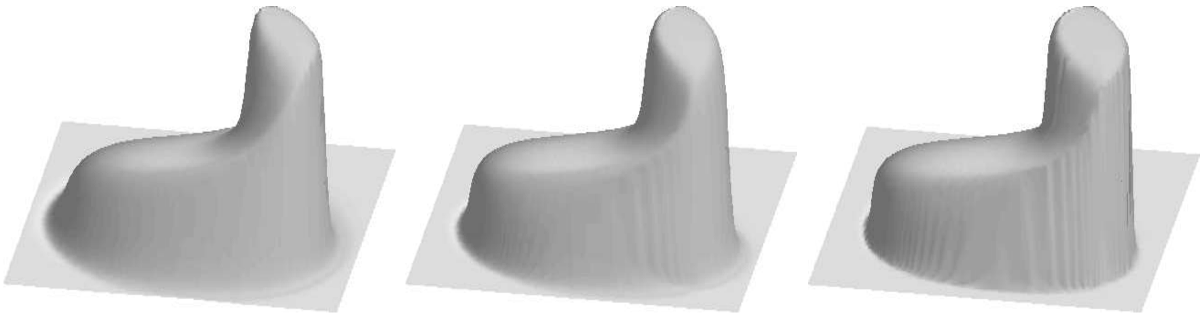

FIGURE 17. 2-D example. 3-D plot of approximate solutions. Left: Finite difference approximation on a 100*100 grid. Middle: OS-F on a 100*100 grid, CFL = 11.0. Right: COS-F on a 100*100 grid, CFL = 11.0.

method. The method uses Lax–Friedrichs for the advective part $u_t + f(u)_x + g(u)_y = 0$ and an implicit central difference approximation for $u_t = \varepsilon \Delta u$. It is evident that the shock layer is resolved much better by COS-F than OS-F. Not surprisingly, we also observe that the finite difference solution is inferior to the splitting methods.

6. Concluding Remarks

We have presented a corrected operator splitting (COS) algorithm which gives a methodology for handling general interaction between nonlinear advection and diffusion. This methodology is a major improvement over standard operator splitting (OS) in cases where both placement and structure of self-sharpening fronts are important. The main observations done in this paper may be summarized as follows:

1. OS schemes require (small) time steps of size $\mathcal{O}(\varepsilon)$ to reproduce the structure of self-sharpening shock fronts.
2. COS schemes obtain correct structure of shock fronts for large time steps. In fact, for travelling "quasi-steady state" solutions, only one or a few time steps are necessary to produce accurate solutions.
3. The most efficient and *natural* implementation is based on front tracking for handling the advection step (COS-F).
4. The implementation based on the $2nd$ order Godunov scheme (COS-G) is much slower than COS-F due to the CFL constraint inherited from the Godunov method and because of the high resolution required to recognize the wave structure of the advection solutions. However, in terms of accuracy the performance of these two schemes is equal.
5. For established shock fronts, the performance of MMOC (in terms of accuracy) compares with that of COS-F(G). However, with MMOC there is no obvious way to obtain residual flux terms dynamically so that general data can be handled.
6. A dimensional splitting extension of the COS-F algorithm is described and implemented for 2-D problems. The numerical results are promising, and in particular the 2-D scheme *appears* to be efficient with respect to runtime versus accuracy, which is due to the ability to resolve steep fronts with large time steps.

Acknowledgement

The authors thank N. H. Risebro and T. F. Russell for reading a preliminary version of the paper and for interesting discussions.

References

[1] J. Barrett and K. Morton. Approximate symmetrization and Petrov-Galerkin methods for diffusion-convection problems. *Comput. Methods Appl. Mech. Engrg.*, 45:97–122, 1984.

[2] J. T. Beale and A. Majda. Rates of convergence for viscous splitting of the Navier-Stokes equations. *Math. Comp.*, 37(156):243–259, 1981.

[3] M. A. Celia and P. Binning. Multiphase models of unsaturated flow: Approaches to the governing equations and numerical methods. In Russell et al., editor, *Mathematical Modeling in Water Resources*, volume 17, pages 257–272. Elsvier, New York, 1992.

[4] M. A. Celia, T. F. Russell, I. Herrera, and R. E. Ewing. An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation. *Adv. Water Resources*, 13:187–205, 1990.

[5] G. Chavent and J. Jaffre. *Mathematical models and finite elements for reservoir simulation*, volume 17 of *Studies in mathematics and it's applications*. North Holland, Amsterdam, 1986.

[6] Y. Chen. *Degenerate parabolic and elliptic equations, Partial differential equations in China (C. Gu et al., eds.)*, pages 1–18. Kluwer Academic Publishers, Netherlands, 1994.

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

[8] G. Crasta and B. Piccoli. Viscosity solutions and uniqueness for systems of inhomogenous balance laws. Preprint, SISSA, Italy, June 1995.

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

[10] 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, 1988.

[11] H. K. Dahle. ELLAM-based operator splitting for nonlinear advection-diffusion equations. Report 98, Department of Mathematics, University of Bergen, June 1995.

[12] H. K. Dahle, M. S. Espedal, R. E. Ewing, and O. Sævareid. Characteristic adaptive subdomain methods for reservoir flow problems. *Numerical Methods for PDEs*, 6:279–309, 1990.

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

[14] 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.*, 122:223–250, 1995.

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

[16] L. Demkowitz and J. T. Oden. An adaptive characteristic Petrov-Galerkin finite element method for convection-dominated linear and nonlinear parabolic problems in two space variables. *Comput. Methods Appl. Mech. Engrg.*, 55:63–87, 1986.

[17] J. Douglas and T. F. Russell. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Num. Anal.*, 19(5):871–885, Oct. 1982.

[18] 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.

[19] R. E. Ewing. Operator splitting and Eulerian-Lagrangian localized adjoint methods for multiphase flow. In Whiteman, editor, *The Mathematics of Finite Elements and Applications VII MAFELAP*, pages 215–232. Academic press, San Diego, CA, 1991.

[20] R. E. Ewing and T. F. Russell. Efficient time stepping methods for miscible displacement problems in porous media. *SIAM J. Num. Anal.*, 19:1–67, 1982.

[21] R. Hansen and M. S. Espedal. On the numerical solution of nonlinear reservoir flow models with gravity. *International Journal for Numerical Methods in Engineering*, 38:2017–2032, 1995.

[22] 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.

[23] H. Holden and N. H. Risebro. Conservation laws with a random source. To appear in Applied Mathematics and Optimization.

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

[25] K. H. Karlsen and N. H. Risebro. Corrected operator splitting for nonlinear parabolic equations. Preprint, University of Bergen, Norway, 1997.

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

[27] S. N. Kruzkov. First order quasi-linear equations in several independent variables. *Math. USSR Sbornik*, 10(2):217–243, 1970.
[28] J. O. Langseth. On an implementation of a front tracking method for hyperbolic conservation laws. *Advances in Engineering Software*, 26(1), 1996.
[29] J. O. Langseth, A. Tveito, and R. Winther. On the convergence of operator splitting applied to conservation laws with source terms. *SIAM J. Appl. Math.*, 33(3):843–863, June 1996.
[30] R. J. LeVeque. *Numerical methods for convservation laws.* Lectures in Mathematics. ETH Zürich. Birkhäuser Verlag, Basel, second edition, 1992.
[31] K.-A. Lie, V. Haugse, and K. H. Karlsen. Dimensional splitting with front tracking and adaptive grid refinement. To appear in Numerical Methods for PDEs.
[32] 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, Nov. 1991.
[33] T. F. Russell. Galerkin time stepping along characteristics for Burgers' equation. In Stepleman et al., editor, *Scientific computing*, pages 183–192. IMACS, North-Holland, 1983.
[34] T. F. Russell and R. V. Trujillo. Eulerian-Lagrangian localized adjoint methods with variable coefficients in multiple dimensions. In G. Gambolati et al., editor, *Proc. 8th Int. Conf. on Computational Methods in Water Resources*, pages 357–363. Computational Mechanics Publications, Southampton, UK, 1990.
[35] J. S. Scroggs. Shock-layer bounds for a singularly perturbed equation. *Quarterly of Applied Mathematics*, LII(3):423–431, Sept. 1995.
[36] A. I. Volpert and S. I. Hudjaev. Cauchy's problem for degenerate second order quasilinear parabolic equations. *Math. USSR Sbornik*, 7(3):365–387, 1969.
[37] H. Wang, H. K. Dahle, M. S. Espedal, R. E. Ewing, R. C. Sharpley, and S. Man. An ELLAM scheme for advection-dispersion equations in two dimensions. Preprint.
[38] H. Wang and R. E. Ewing. Optimal-order convergence rates for ELLAM for reactive transport and contamination in groundwater. *Numerical Methods for PDEs*, 11:1–35, 1995.
[39] H. Wang, R. E. Ewing, and M. A. Celia. Eulerian-Lagrangian localized adjoint methods for reactive transport with biodegradation. *Numerical Methods for PDEs*, 11:229–254, 1995.
[40] H. Wang, R. E. Ewing, and T. F. Russell. Eulerian-Lagrangian localized adjoint methods and their convergence analysis. *IMA J. Numer. Anal.*, 15, 1995.
[41] Z. Wu and J. Wang. Some results on quasilinear degenerate parabolic equations of second order. In *Proc. of the Beijing. Sym. on Diff. Geom. and Diff. Eqns.*, 1980.
[42] J. Zhao. Some properties of solutions of quasilinear degenerate parabolic equations and quasilinear degenerate elliptic equations. *Northeastern Math. J.*, (2):281–302, 1986.

Appendix A. The Petrov-Galerkin Finite Element Method

We will now give a very brief overview of the Petrov-Galerkin method as it is used in the present paper. The reader is referred to [10] for various details and aspects concerning the actual implementation.

Consider the linear variable-coefficient advection-diffusion problem

$$\partial_t u + \partial_x \left[b(x)u - \varepsilon\nu(x)\partial_x u \right] = 0, \qquad u(x,0) = \bar{u}(x),\ u(a,t) = u(a,t) = 0, \tag{20}$$

where we for simplicity have taken the boundary values to be zero. We are interested in computing approximations to (20) at time $t = \Delta t$. Let S_h be the standard piecewise linear finite element space spanned by hat functions $\{\theta_i(x)\}$ with nodes $\{x_i\}$, so that boundary conditions automatically are taken care of, and let $\Delta x_i = x_i - x_{i-1}$ be the width of each element. Hence, a finite element approximation $u_h \in S_h$ may be written

$$u_h = \sum_i u_i \theta_i,$$

for some proper coefficients $\{u_i\}$. The time derivative term in (20) is replaced by a single backward Euler step. Note that this implies that Δt must be reasonably small to ensure that the method of "freezing" the coefficients converges, see [21]. For larger Δt it may be necessary do several Euler steps (see Example 5).

Multiplying equation (20) by a test function $\psi_i(x)$ and subsequently doing an integration by parts, leads to the variational formulation

$$(u_h, \psi_i) + \Delta t\big(b(x)u_h - \varepsilon\nu(x)\partial_x u_h, \partial_x \psi_i\big) = (\bar{u}, \psi_i), \tag{21}$$

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

Since (21) is not completely symmetrized by the advection step because of the b-term, we have to choose a test space T_h different from the trial space S_h to stabilize equation (21). A theory for choosing an appropriate test space is given in [1], which consists in choosing a discrete test space that transforms the bilinear form (21) into an equivalent V-elliptic and symmetric bilinear form. Based on previous experience, e.g. [10], a good choice seems to be quadratic functions with support on $[x_{i-1}, x_{i+1}]$,

$$\psi_i = \begin{cases} \theta_i + c_{i-1/2}\sigma_i, & x_{i-1} \le x \le x_i, \\ \theta_i + c_{i+1/2}\sigma_i, & x_i \le x \le x_{i+1}, \end{cases}$$

where

$$\sigma_i(x) = \begin{cases} \frac{(x-x_{i-1})(x-x_i)}{\Delta x_i}, & x_{i-1} \le x \le x_i, \\ \frac{-(x-x_i)(x-x_{i+1})}{\Delta x_{i+1}}, & x_i \le x \le x_{i+1}, \end{cases}$$

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

Here,

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

is the mesh Péclet number on element $[x_{i-1}, x_i]$ and $b_{i-1/2}$, $\nu_{i-1/2}$ are average values over this element.

Some extra numerical diffusion may be introduced by mass lumping, i.e.,

$$(u_h, \psi_i) \approx \frac{(\Delta x_i + \Delta x_{i+1})}{2} u_i, \qquad (\bar{u}, \psi_i) \approx \frac{(\Delta x_i + \Delta x_{i+1})}{2} \bar{u}_i.$$

Note that this is consistent with the order of the finite element approximation. Using the above notation, the finite element approximation now reads:

Find $u_h(x) \in S_h$ so that (21) is satisfied for each $\psi_i(x) \in T_h$.

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

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

(K. Brusdal)
Department of Mathematics
University of Bergen
Johs. Brunsgt. 12
N-5008 Bergen, Norway
E-mail address: kari.brusdal@mi.uib.no

(H. K. Dahle)
Department of Mathematics
University of Bergen
Johs. Brunsgt. 12
N-5008 Bergen, Norway
E-mail address: helge.dahle@mi.uib.no

(S. Evje)
Department of Mathematics
University of Bergen
Johs. Brunsgt. 12
N-5008 Bergen, Norway
E-mail address: steinar.evje@mi.uib.no

(K.–A. Lie)
Department of Mathematical Sciences
Norwegian University of Science and Technology
N-7034 Trondheim, Norway
E-mail address: andreas@imf.unit.no