UNCONDITIONALLY STABLE SPLITTING METHODS FOR THE SHALLOW WATER EQUATIONS

RUNAR HOLDAHL, HELGE HOLDEN, AND KNUT-ANDREAS LIE

Abstract. The front-tracking method for hyperbolic conservation laws is combined with operator splitting to study the shallow water equations. Furthermore, the method includes adaptive grid refinement in multidimensions and shock tracking in one dimension. The front-tracking method is unconditionally stable, but for practical computations feasible CFL numbers are moderately above unity (typically between 1 and 5). The method resolves shocks sharply and is highly efficient.

The numerical technique is applied to four test cases, the first being an expanding bore with rotational symmetry. The second problem addresses the question of describing the time-development of two constant water levels separated by a dam that breaks instantaneously. The third problem compares the front-tracking method with an explicit analytic solution of water waves rotating over a parabolic bottom profile. Finally, we study flow over an obstacle in one-dimension.

1. Introduction

The shallow water equations model free surface flow for a fluid under the influence of gravity in the case where the vertical scale is assumed to be much smaller than the horizontal scale. They can be derived from the depth averaged incompressible Navier–Stokes equations and express conservation of mass and momentum. The shallow water equations have applications to a wide range of phenomena other than water waves, e.g., avalanches [27] and atmospheric flow [7]. In the case of free surface flow when the shallow water approximation is not valid, it is common to model the surface waves using several layers of shallow water equations coupled via the pressure, [21, 24, 33, 36]. Furthermore, the shallow water equations coincide with the Euler equations of gas dynamics in the case of isentropic gas flow with $\gamma = 2$. Thus the 2×2 shallow water equations often serve as a simpler test case compared with the full 3×3 system of Euler equations.

Front tracking is a powerful tool to analyze systems of strictly hyperbolic conservation laws in one spatial dimension. It can briefly be described as follows for a general system. We make the standard approximation of the initial data by a piecewise constant function. Due to the finite speed of propagation in hyperbolic systems, the initial value problem is reduced to a family of Riemann problems, i.e., isolated jumps separating two constant states. If the two states are close in state space, the solution of the Riemann problem can

Date: December 23, 1998.
1991 *Mathematics Subject Classification.* 65M12, 65M50, 76B15, 35L65.
Key words and phrases. Front tracking, operator splitting, grid refinement, shallow water equations.
The third author has been supported by grant 100555/410 from the Norwegian Research Council.

be described by the Lax construction. Thus the solution consists of a combination of at most n elementary waves, i.e., shocks, contact discontinuities, or rarefaction waves. For the 2×2 shallow water equations in one spatial dimension, the Riemann solution consists of shocks and rarefaction waves only. However, when using dimensional splitting to solve multidimensional equations, transverse velocity components that are passively advected can be regarded as contact discontinuities.

In the front-tracking approximation shocks and contact discontinuities are unchanged while the rarefaction wave is approximated by sampling points along the wave. In this way we get a piecewise constant solution. We denote the discontinuities as fronts. When fronts collide, new Riemann problems are created that are solved in the same manner.

As the approximation of the initial data improves and the number of points on rarefaction waves increase, the approximate solution converges to a solution of the initial value problem, see Risebro [28]. This result holds when the system is strictly hyperbolic and the initial data has small total variation.

Recently the wave-front tracking method of Bressan *et al.*, closely related to the front-tracking method just described, has been shown to yield uniqueness and stability of solutions, a major breakthrough in the theory of systems of conservation laws. See [4] and [5].

Front tracking as we have described it so far, is restricted to one spatial dimension. The extension to multidimensional systems is accomplished by dimensional splitting where the solution is obtained in one spatial direction at the time, then projected onto a grid before the equation is solved in the other spatial directions, and so on. In the inhomogeneous case we use operator splitting to handle source terms.

For other numerical implementations of front tracking for hyperbolic systems we refer to [29, 30, 12, 13, 37, 8]. In particular, [8] involves similar numerical methods applied to the Euler equations of gas dynamics in two space dimensions. Our method is related to the large time stepping method of LeVeque, see [14]. Here we apply front tracking as a numerical technique to the shallow water equations. It is tested on four test cases.

The first case concerns the collapse of an elevated circular region of water. The rotational symmetry of the problem allows for testing of possible grid effects. Front tracking is compared with three second-order, unsplit methods; the non-oscillatory central schemes of Jiang and Tadmor [11], the composite CFLF4 method of Liska and Wendroff [22, 23], as well as the CLAWPACK code of LeVeque [15, 16, 17]. While Jiang and Tadmor's method and the CFLF4 method are based on central differences, the wave propagation method of CLAWPACK and front tracking are based on the solution of Riemann problems. Front tracking performs similarly to JT and CLAWPACK (with lower resolution at the dip, but better resolution at the origin) but is subject to slightly more distortions due to the Cartesian grid. However, front tracking runs at slightly higher CFL numbers and is also faster than other methods. See [26, 33, 34, 35] for alternative numerical methods applied to the same problem.

The second problem models the breaking of a dam, see [1, 2, 3, 6] for earlier treatments of this problem. More precisely, we have two different levels of water separated by a dam which is assumed to break instantaneously. Compared with methods described in [1, 2, 3, 6] front tracking is considerably less diffusive when run on the same 40×40 grid with a CFL number 1.5. On a finer grid, we observe improved accuracy by increasing the CFL number to around 4.0. We have also implemented one level of adaptive grid refinement, see [8, 20]. By including grid refinement for this problem, we achieve high accuracy with a reduced computational cost.

The third problem addresses the question of describing the motion of water waves on a parabolic bottom, comparing with an explicit solution due to Thacker [32]. The solution describes a planar water surface that rotates. Here the equations are inhomogeneous, and the source term is treated using operator splitting. The front-tracking method shows excellent agreement with the exact solution in the conserved variables. However, due to the use of dry-bed Riemann problems, rather large errors are observed in the velocities near the shore line. A convergence study verifies first-order convergence of the method in L^1 norm both for conserved and primitive variables and in L^∞ norm for conserved variables.

While the previous three problems are two-dimensional, the last problem treats flow over an obstacle in one spatial dimension. To treat the resulting source term in the equations we employ an operator splitting. The simulations are compared with a composite scheme of Liska and Wendroff, see [22]. Sharp resolution of shocks (within *one* grid cell) is obtained by including explicit shock tracking.

2. Front Tracking and Operator Splitting

2.1. **Front Tracking.** Let the strictly hyperbolic conservation law be given by

$$U_t + f(U)_x = 0, \quad U|_{t=0} = U_0. \tag{1}$$

When we approximate U_0 by a piecewise constant function $U_{0,\delta}$, where δ measures the approximation, the initial value problem (1) is reduced to a family of Riemann problems that will be solved separately. Consider a Riemann problem with left state U_L and right state U_R. The solution can be written as a combination of at most n elementary waves. Shock waves and contact discontinuities remain unchanged. Assume that a rarefaction wave from the jth family connects states U_l and U_r, that is,

$$U_r = \mathcal{R}_j(U_l; \epsilon_j),$$

for some ϵ_j, where $\mathcal{R}_j(U; \cdot)$ is the jth rarefaction curve from (left) state U. Define $M = \lceil(\lambda_j(U_r) - \lambda_j(U_l))/\delta\rceil$, where $\lambda_j(U)$ denotes the jth eigenvalue of $f'(U)$. Furthermore, let $\tilde{\delta} = (\lambda_j(U_r) - \lambda_j(U_l))/M$. We replace the rarefaction wave by the step function

$$U(x,t) = \begin{cases} U^0 = U_l & \text{for } x \leq \xi_1 t, \\ U^1 & \text{for } \xi_1 t \leq x \leq \xi_2 t, \\ \vdots & \\ U^k & \text{for } \xi_k t \leq x \leq \xi_{k+1} t, \\ \vdots & \\ U^{M-1} & \text{for } \xi_{M-1} t \leq x \leq \xi_M t, \\ U^M = U_r & \text{for } x \geq \xi_M t. \end{cases}$$

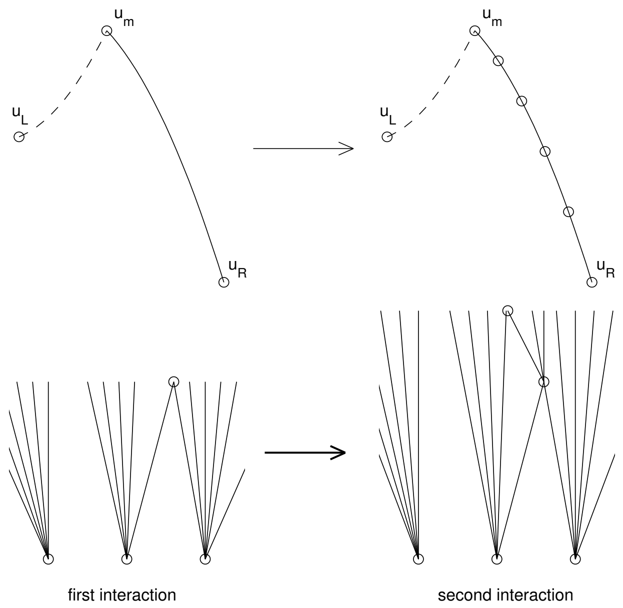

FIGURE 1. Illustration of the front-tracking construction. (Top) Approximation in phase plane. The Riemann problem (u_L, u_R) consists of a shock wave connecting u_L to an intermediate state u_m and then a rarefaction connection u_m to u_R. (Bottom) Tracking fronts in the (x,t)-plane; the circles indicate positions where Riemann problems are solved.

Here

$$U^k = \mathcal{R}_j(U_l; k\tilde{\delta}), \qquad \xi_k = \frac{1}{2}(\lambda_j(U^k) + \lambda_j(U^{k-1})).$$

The resulting approximate solution is a piecewise constant function which we denote by U_δ. Moreover, we denote discontinuities by *fronts*. When fronts collide, new Riemann problem appear that are solved in the same manner. The solution process is illustrated in Figure 1. In general there will be a build-up of the number of fronts. To avoid an infinite number of fronts in finite time, we perform a data reduction by removing small waves. We measure the size of a wave by the variation of the wave divided by the number of components that are non-constant over the wave. The cut-off level of small waves is predetermined by the user, see [12]. Furthermore, a Riemann problem is ignored if the distance in state space between the two constant states is smaller than a certain lower threshold.

In the implementation of the front-tracking method, each front is represented as a data object. The objects are linked in a spatial list (from left to right) and colliding fronts are inserted in a collision list. The tracking part thus consists of operating on the first element of the collision list. That is, we take out the first collision from the list and remove the corresponding fronts from the spatial list. Then we solve the new Riemann problem, insert the new fronts into the spatial list, compute possible collisions with the nearest neighbors, and insert the collision into the correct place in the list.

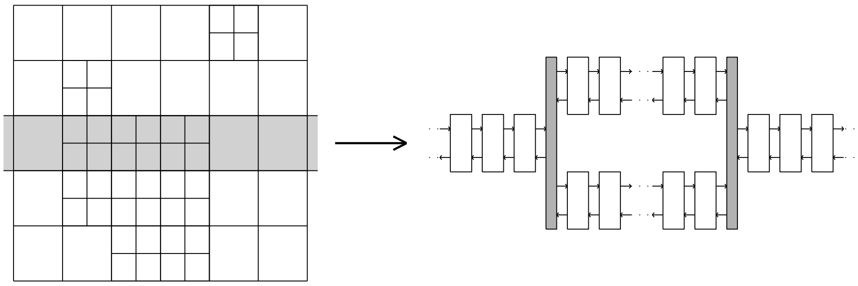

FIGURE 2. Front tracking on one row in the underlying coarse grid.

Boundaries are treated as fronts with a special identifying tag. This allows the code to treat collisions between fronts without taking into account if a front is part of a boundary or not. Moving fronts are thus easily implemented. See [12, 8].

2.2. **Operator Splitting.** Front tracking as we have described it, is a one-dimensional method, and the extension to multidimensions is accomplished by dimensional splitting. In this approach we introduce a rectangular Cartesian grid, and solve the hyperbolic conservation law in "tubes" in one spatial direction. The solution is then projected onto the grid, and the conservation law is subsequently solved in tubes in the next spatial direction followed by a projection onto the grid. The procedure is repeated in all spatial directions.

Consider the solution of $u_t + f(u)_x + g(u)_y = 0$. Let π denote the projection operator and $\mathcal{S}_t^{f,x}$ the front-tracking construction for $u_t + f(u)_x = 0$. Then the approximate solution reads,

$$u(x, y, n\Delta t) \approx \left[\pi\mathcal{S}_{\Delta t/2}^{f,x} \circ \pi\mathcal{S}_{\Delta t}^{g,y} \circ \pi\mathcal{S}_{\Delta t/2}^{f,x}\right]^n \pi u_0(x, y) \tag{2}$$

for the second-order Strang splitting. In the scalar case, this method is well understood, and one can prove that the method converges to the right solution [9] and derive a priori error estimate [20].

A similar strategy is applied for inhomogeneous equations. Consider the equation $u_t + f(u)_x = q(x, t, u)$. Let $\mathcal{R}_t$ denote an approximate solution operator for $u_t = q(x, t, u)$. Then the first-order Godunov splitting reads

$$u(x, y, n\Delta t) \approx \left[\mathcal{R}_{\Delta t} \circ \pi\mathcal{S}_{\Delta t}^{f,x}\right]^n \pi u_0(x). \tag{3}$$

Second order is formally obtained by using Strang splitting.

2.3. **Adaptive Grid Refinement.** To allow for finer resolution near shocks we have implemented adaptive grid refinement as described in [20]. The method uses an underlying coarse grid. Each cell in this regular Cartesian grid may contain a regular subpartition. The subpartitions may appear or disappear during the projection phase (as described below) or remain fixed throughout the computation. We use a fixed subpartition ratio for all coarse cells.

The modified front-tracking algorithm acts on one tube (a row or column) in the underlying coarse grid. This tube may locally contain parallel tubes of fine cells (see Figure 2).

Inside each local tube (consisting of cells with the same cross-section), we can use front tracking as described above, giving a left-to-right list of fronts. A global front list is obtained by connecting local lists at the interfaces between coarse and refined blocks using a special static front (see Figure 2). All collisions are kept in a global collision list that can be updated as explained above. The only exception occurs at interfaces.

A Riemann problem at an interface is resolved by solving one Riemann problem for each refined tube. All fronts going into the refined tube are inserted into the corresponding local front list. Then all fronts going into the coarse tube (possibly from different Riemann problems) are assigned new states according to the cross-sectional average in the coarse tube, collected in a list of increasing wave speeds, and inserted into the local list for the coarse tube. Collisions at the interfaces are handled as follows: Fronts propagating out of a refined region are collected, assigned new states, and inserted into the connected coarse list. Fronts propagating into a refined region are copied and inserted into each connected refined list.

In the projection phase, we traverse the front list (from left to right) and assign values according to the grid-cell average for each cell. Coarse grid cells are refined whenever the variation over all fronts within a grid cell exceeds a predefined value ($\mathrm{TV}_{\mathrm{max}}$). Similarly, in a postprocessing phase (after the projection) we determine the variation over all refined cells within each underlying coarse cell, and if the total variation is less than a given value ($\mathrm{TV}_{\mathrm{min}}$), we return to the original coarse grid in that cell. By adopting this strategy, the projection error in each time step is reduced, and the accuracy is enhanced. It is important to notice that no local time stepping is required. Furthermore, the method is strictly local, and no patching of refined regions is performed. Only one level of refinement is currently implemented. For more details we refer to [20].

2.4. **Shock Tracking.** When front tracking is applied as part of a fractional steps procedure, it is convenient to project solutions onto a fixed grid, i.e., turn the method into a shock-capturing scheme (see (2) and (3)). In doing so, discontinuities in the solution are smeared and resolution is lost. Moreover, for shock waves, the projection introduces waves in the passive families, which may lead to post-shock oscillations that pollute the computed solution (see [8]).

To amend these problems it is natural to introduce shock tracking. The idea is to retain a fixed grid and introduce extra cell boundaries at the location of strong shocks (or contact discontinuities for general systems). These (moving) grid points are created at the end of each hyperbolic step, and destroyed at the outset of the next. No grid points are used during the hyperbolic step, since wave interactions are resolved by the tracking algorithm. Identification of shocks is easy, since the wave type of each discontinuity is a by-product of the Riemann solver. Thus, shock tracking hardly introduces any computational overhead. Figure 3 shows the propagation of a single shock.

No special attention is paid to the projection of rarefaction waves. However, if sharper resolution of these waves is needed locally, both the shock-capturing and the shock-tracking method can easily be modified to incorporate adaptive mesh refinement as described above.

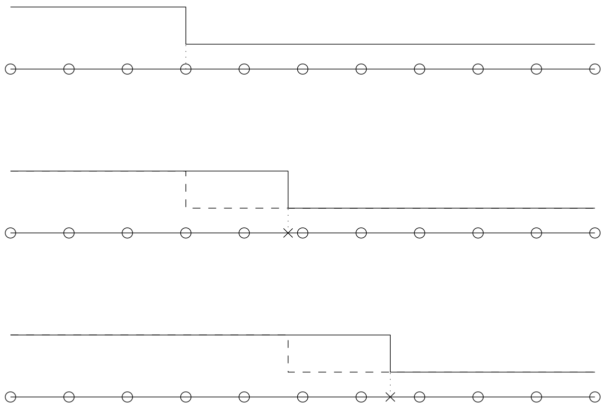

FIGURE 3. Propagation of a single shock by shock tracking; $\circ$ denotes uniform grid cells and $\times$ the cell used to track the shock.

The new shock-tracking method is related to several existing methods, including the front-tracking methods of LeVeque and Shyue [18] and Swartz and Wendroff [31]. See, for instance, the discussion in [18, §2]. The unique feature of our method is that the hyperbolic solver is unconditionally stable and resolves *nonlinear* wave interactions explicitly throughout an arbitrary time step. Thus, the size of the splitting steps is only restricted by the dynamics of the problem. Moreover, we avoid stability problems which arise in most shock-tracking methods due to the presence of arbitrary small grid cells. For more examples, see [19]. Unfortunately, it seems difficult to generalize the method to multidimensions.

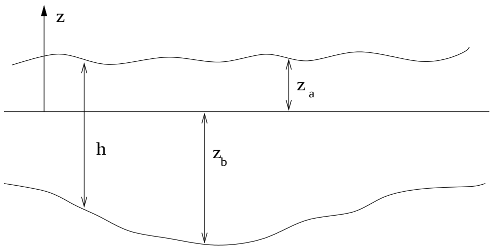

FIGURE 4. The quantities involved in the shallow water equations.

3. GOVERNING EQUATIONS

The shallow water equations describe free surface flow for a fluid under the influence of gravity when the height of the top surface above the bottom is sufficiently small compared with some characteristic length of motion. In particular, vertical motion is ignored, and we have hydrostatic balance in the vertical direction.

We may write the shallow water system as the following inhomogeneous system of conservation laws

$$\begin{bmatrix} h \\ hu \\ hv \end{bmatrix}_t + \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}_x + \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ -gh\frac{\partial z_b}{\partial x} \\ -gh\frac{\partial z_b}{\partial y} \end{bmatrix}. \tag{4}$$

Here h denotes water depth, u and v are depth-averaged velocities in the x and y directions, respectively, and g is the acceleration of gravity. The right-hand side of (4) models the influence of a variable bottom. Figure 4 illustrates the situation and the quantities involved. The function $z_b = z_b(x, y)$ describes the bottom topography while $z_a = z_a(x, y, t)$ models the water surface elevation, and we have that $h = z_a - z_b$.

The Riemann problem for the shallow water equations in the one-dimensional homogeneous case can be solved explicitly. The system is strictly hyperbolic, and genuinely nonlinear in both families. The rarefaction waves read

$$u = \begin{cases} u_l - 2(\sqrt{h} - \sqrt{h_l}) & \text{for the first family with } 0 < h < h_l, \\ u_l + 2(\sqrt{h} - \sqrt{h_l}) & \text{for the second family with } h > h_l, \end{cases} \tag{5}$$

for a fixed left state (u_l, h_l). The Lax shock curves are given by

$$u = \begin{cases} u_l - \frac{1}{\sqrt{2}}(h - h_l)\sqrt{h^{-1} + h_l^{-1}} & \text{for the slow shock with } h > h_l, \\ u_l + \frac{1}{\sqrt{2}}(h - h_l)\sqrt{h^{-1} + h_l^{-1}} & \text{for the fast shock with } 0 < h < h_l. \end{cases} \tag{6}$$

The "vacuum case" $h = 0$ (also called the dry-bed Riemann problem) can be incorporated in the solution, see [25, 10]. In the next section we present some numerical solutions of (4) computed by front tracking.

4. Numerical Results

In this section we solve four different problems in shallow water hydrodynamics by the front-tracking algorithm. The first two problems are homogeneous, while the bottom profile is varying in the last two examples such that the governing equations are inhomogeneous. The numerical solutions are compared with both exact solutions and solutions computed by other numerical methods. These methods are:

- LWLF4 – one-dimensional composite scheme consisting of three Lax–Wendroff steps followed by a Lax-Friedrichs step [22].
- CFLF4 – composite scheme consisting of three steps with a second-order predictor-corrector method followed by a Lax-Friedrichs step [22, 23].
- JT – a second-order, non-oscillatory, central difference scheme [11] based on a predictor-corrector strategy. In the prediction step, point values are constructed using non-oscillatory piecewise linear reconstruction of given cell averages. In the corrector step, the solution is averaged on a staggered grid.
- CLAWPACK $\mathrm{T}^{2,2}$ – wave propagation method from the software package CLAWPACK [15, 16]. In CLAWPACK, Riemann problems are solved at cell interfaces and limiters are

applied to suppress oscillations from second-order terms. To model crossderivatives, Riemann problems are also solved in the transverse directions. The term CLAWPACK $T^{2,2}$ refers to the method with second-order correction terms and transverse propagation of increment waves, i.e., a second-order unsplit method.

CLAWPACK is implemented in FORTRAN, while we coded the other three methods in C.

4.1. **Explosion problem.** To test the accuracy of the front tracking and dimensional splitting algorithms with respect to grid orientation effects we try to solve a rotational symmetric problem on a Cartesian grid. We compare the front-tracking method with the unsplit methods JT, CFLF4 and CLAWPACK $T^{2,2}$. A reference solution is obtained by solving the reduced inhomogeneous system

$$\begin{bmatrix} h \\ hu_r \end{bmatrix}_t + \begin{bmatrix} hu \\ hu_r^2 + \frac{1}{2}gh^2 \end{bmatrix}_r = -\frac{1}{r}\begin{bmatrix} hu_r \\ hu_r^2 \end{bmatrix}.$$

To this end, we use the shock-tracking method introduced above on a very fine grid (5 000 cells and CFL number 2.0).

The explosion problem considered here has initial data that consists of two constant states separated by a circle,

$$\begin{aligned} h(x,y,0) &= \begin{cases} 1.0, & \sqrt{x^2+y^2} \le 0.3, \\ 0.1, & \text{otherwise}, \end{cases} \\ u(x,y,0) &= v(x,y,0) = 0. \end{aligned}$$

We assume absorbing boundary conditions. The solution consists of an expanding circular shock wave. Within the shock there is a rarefaction wave transporting water from the original deep region out to the shock. In the numerical computations the domain is $[-1.0, 1.0] \times [-1.0, 1.0]$ with absorbing boundary conditions. For all methods a uniform Cartesian grid with 160 grid cells in each direction was used. The initial data was approximated by taking the spatial average over each grid cell.

Scatter plots of the water surface elevation at time $t = 0.19$ are shown in Figure 5, while Figure 6 shows the corresponding plots of the radial velocity. In both figures the radially symmetric solution is represented by a solid line. CFL numbers are displayed in Table 4.1 together with the CPU time used in each computation. The CPU times cannot be

TABLE 1. CFL numbers and CPU times for the explosion problem.

method	CFL number (fixed)	CPU time (normalized)
front tracking	1.50	1.0
CFLF4	0.50	12.6
JT with min-mod limiter (MM_2 in [11])	0.475	8.2
CLAWPACK $T^{2,2}$ with MC limiter	0.90	27.7

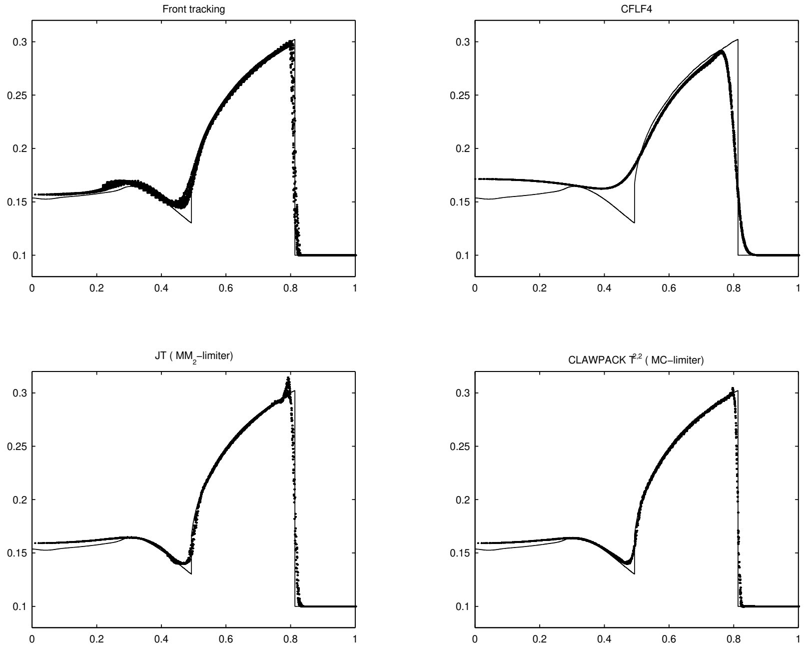

FIGURE 5. Scatter plot (physical quantity plotted against the distance from origin for each grid cell) of the water surface elevation for the explosion problem at time $t = 0.19$. The solid line is the radially symmetric solution.

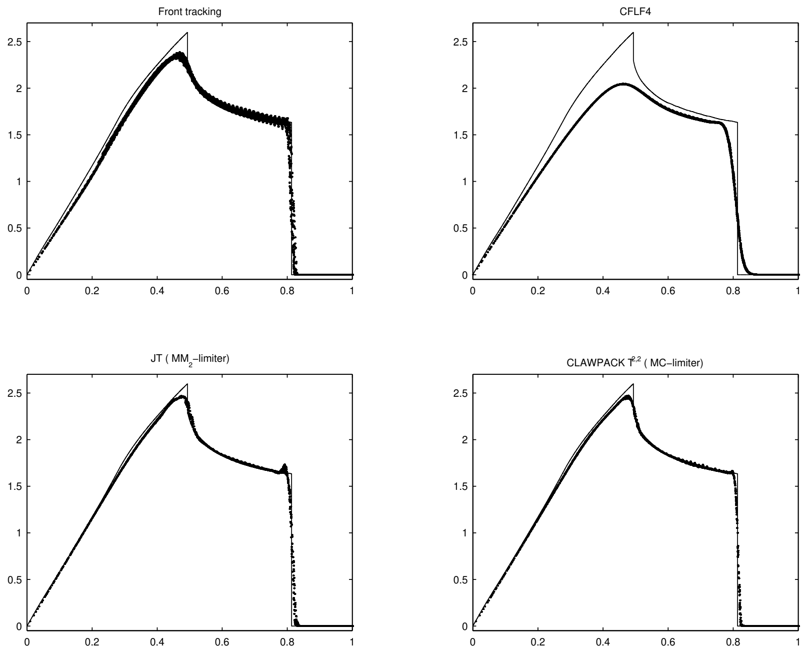

FIGURE 6. Scatter plot of the velocity distribution for the explosion problem at time $t = 0.19$. The solid line is the radially symmetric solution.

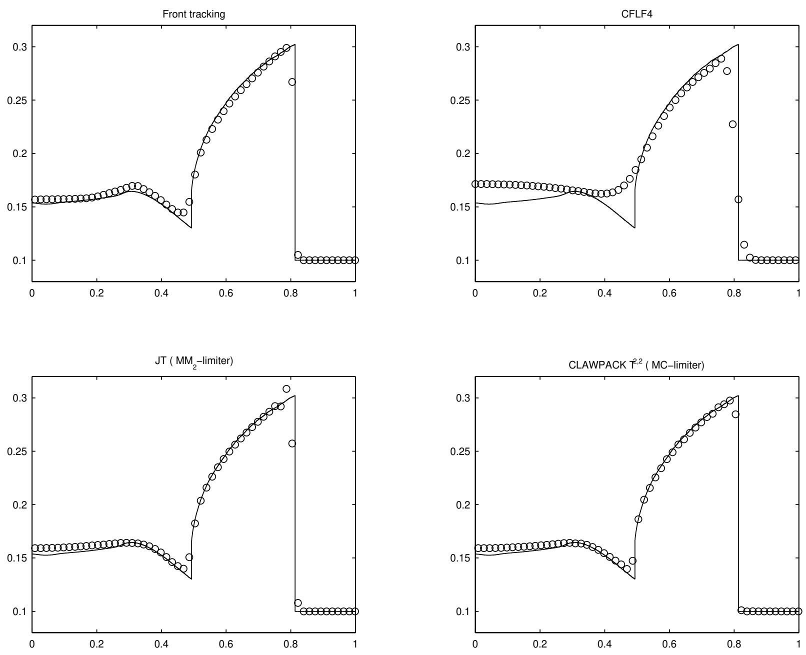

FIGURE 7. The surface elevation along the diagonal $x = y$ for $x > 0$.

compared directly since the numerical schemes have not been optimized, but they should give an indication of the efficiency of the various methods.

As far as computational time is concerned, front tracking is most efficient, and other methods are considerably slower. From the plots of surface elevation and radial velocity we see that the best resolution is achieved by JT and CLAWPACK in all regions except near the origin. Front tracking gives equally good resolution of the leading shock (two grid cells), is less accurate at the minimum/maximum point at $x \approx 0.5$, but gives the best approximation near the sonic point at the origin. CFLF4 resolves all features in the solution rather poorly – the shock is for instance resolved by 4-5 grid cells, see Figure 7.

The scatter plots show that all solutions are influenced by the underlying Cartesian grid. However, CLAWPACK and CFLF4 preserve the symmetry very well. For JT we notice some overshoots behind the leading shock (along the diagonals in the grid, see Figure 7) In the front-tracking solution the grid orientation effects are more noticeable, which is to be expected since this method is the only splitting method of the four.

4.2. **Dambreak problem.** Here we discuss a problem reported in [1, 2, 3, 6]. Consider a 200 m long and 200 m wide reservoir with two different constant levels of water separated by a dam. At $t = 0$ s part of the dam breaks instantaneously. The dam is 10 m thick and the breach is assumed to be 75 m wide. Figure 8 shows a sketch of the reservoir. The initial conditions are given by

$$
\begin{aligned}
h(x,y,0) &= \begin{cases} 10\,\mathrm{m}, & x \le 100\,\mathrm{m}, \\ 5\,\mathrm{m}, & \text{otherwise}, \end{cases} \\
u(x,y,0) &= v(x,y,0) = 0\,\mathrm{m/s}.
\end{aligned}
$$

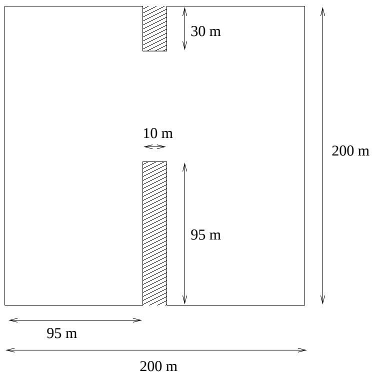

FIGURE 8. The reservoir.

In the references mentioned above the solution is displayed at time $t = 7.2\,\mathrm{s}$, but to illustrate how the solution evolves in time we have shown the solution at times $t = 3.6\,\mathrm{s}$, $7.2\,\mathrm{s}$, and $10.8\,\mathrm{s}$ in Figure 9. In the left column the water surface elevation is plotted, while the right column contains the corresponding contour plots. All computations are made on a uniform Cartesian grid with 40×40 grid cells and a fixed CFL number of 1.5.

Initially the solution consists of a bore propagating into the downstream side of the reservoir while spreading laterally. At the same time a rarefaction wave travels to the left. In the second plot in Figure 9 the bore has been reflected from one of the side walls of the reservoir and in the third plot the bore has just reached the wall in the downstream direction of the dam.

Compared with the numerical solutions computed in the references, the front-tracking solution is less diffusive than both the first-order method in [3] and the second-order methods in [1, 2, 6]. The front part of the bore is resolved by 2 grid points using front tracking, while approximately 4 points are required by the second-order methods. The first-order method is even more diffusive. Furthermore, in regions where the solution is smooth, the front-tracking solution looks very similar to the ones obtained by the second-order methods. In particular, the contour plot is almost the same as the one produced by the second-order method in [1].

The accuracy of our method is determined by two major factors: Errors due to dimensional splitting and numerical diffusion caused by the projections. To resolve the dynamics of any problem in a satisfactory way we have to use a certain minimum number of time steps in the splitting method. For the relatively coarse (40×40) grid used above this minimum corresponds to a low CFL number of 1.5. However, once one is above this minimum of time steps, it is interesting to see how the two error contributions balance. Figure 10 shows the water elevation computed on a finer (200×200) grid with different CFL numbers. We see that increasing the CFL number from 1.0 to 1.5 gives sharper resolution of the leading shock wave. Moreover, increasing the CFL number to 4.0 improves the resolution of the reflected wave near the upper boundary and the region near the walls of the sluice. This

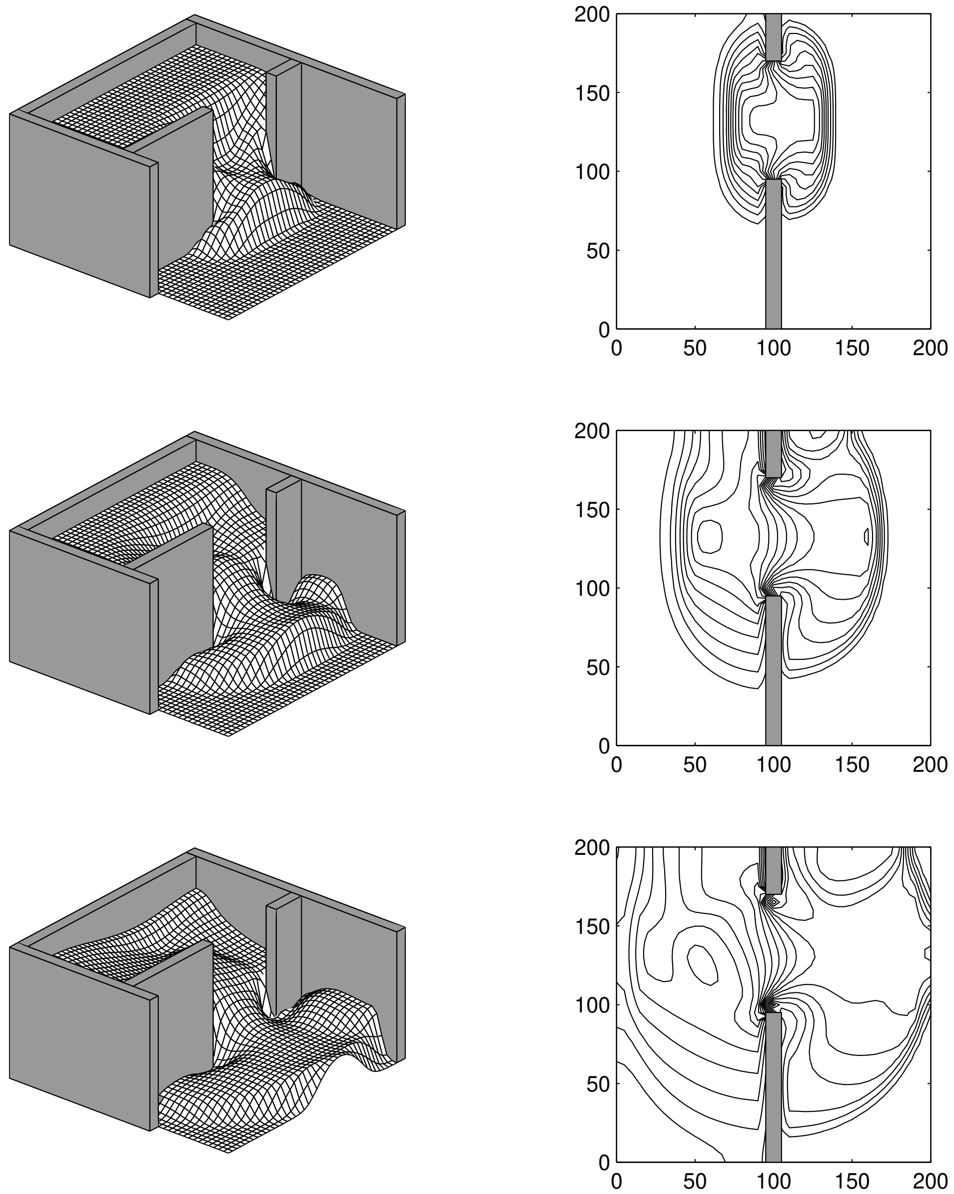

FIGURE 9. Water surface elevation for the dambreak problem at times $t = 3.6\,\mathrm{s}$, $7.2\,\mathrm{s}$, $10.8\,\mathrm{s}$.

means that the projection error is dominant for low CFL numbers, since the accuracy is improved remarkably by increasing the time steps.

When the CFL number is increased even further, a new factor enters the equation — oscillations that are generated by the projection operator (as discussed above). Unfortunately, these oscillations become more and more dominant for increasing CFL numbers and the quality of the solution deteriorates gradually. Table 2 shows various statistics from the

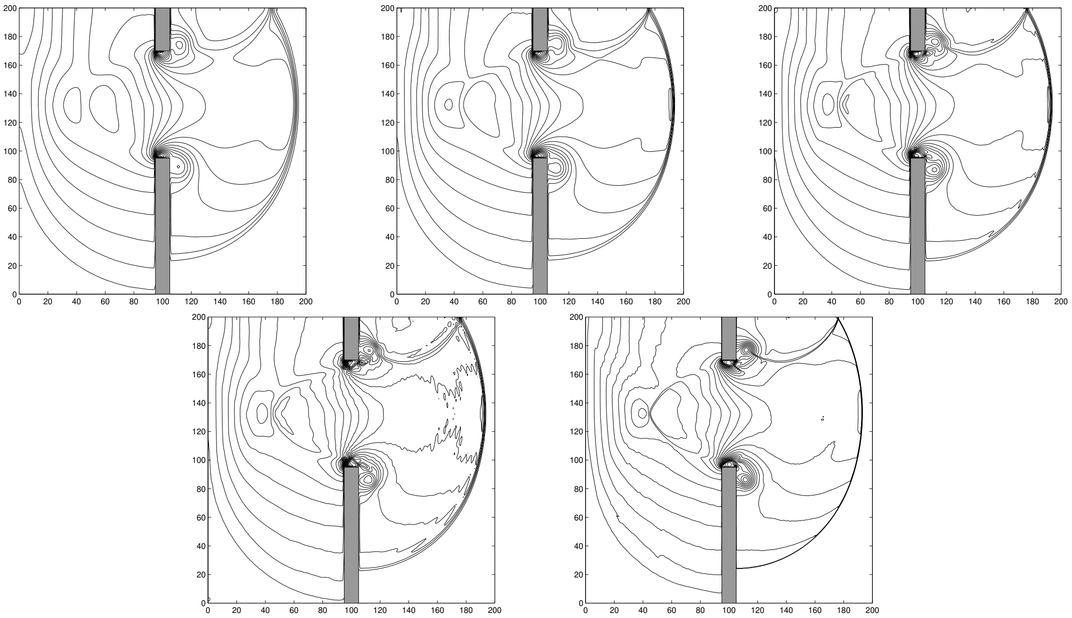

FIGURE 10. Contour plots of the water elevation at time $t = 10.0\,\mathrm{s}$ computed on a 200×200 grid with CFL numbers 1.0 (upper left), 1.5, 4.0, and 6.0 (lower left). The lower right plot is a simulation with CFL number 2.0 on a 800×800 grid.

TABLE 2. Comparison of different performance statistics for the dambreak problem on a 200×200 grid. The third and fourth columns give average values per cell per time step.

CFL	time steps	shock collisions	calls to Riemann solver	CPU time (normalized)
1.0	137	1.59	1.51	1.40
1.5	91	2.06	1.57	1.00
4.0	35	9.80	4.14	1.01
6.0	24	17.31	6.70	1.19

four runs. As the length of each time step increases with the CFL number, the number of wave interactions will also increase as seen from the third column. However, the number of calls to the Riemann solver increases more slowly, indicating an increasing amount of weak interactions (for which no Riemann problem is solved).

4.3. **Dambreak problem with grid refinement.** We have tested the adaptive grid refinement on the dambreak problem of the previous section. In the numerical calculations a local refinement factor 2 was chosen in each direction and one level of refinement was used. The underlying coarse grid had 80×80 cells and we used threshold parameters $(\mathrm{TV}_{\min}, \mathrm{TV}_{\max}) = (2.5, 5.0)$ in the adaption algorithm, see [8]. Figure 11 shows the grid and a contour plot of the surface elevation at time $t = 10\,\mathrm{s}$. For comparison Figure 12

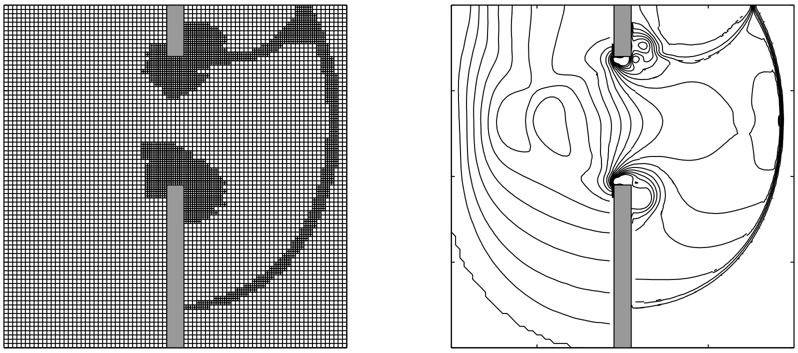

FIGURE 11. Grid in final step and contour plot for the dambreak problem at time $t = 10\,\mathrm{s}$ using grid refinement on a 80×80 grid.

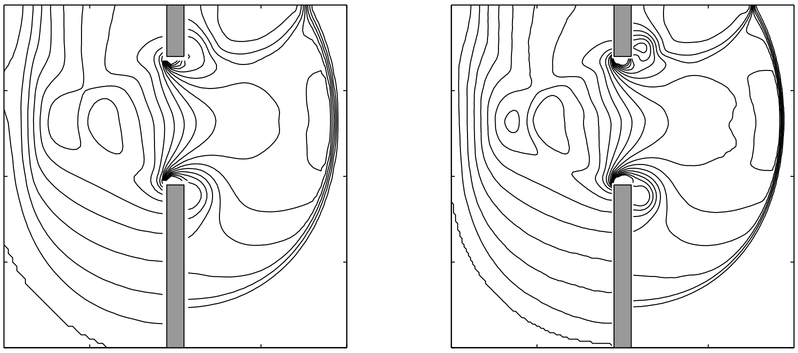

FIGURE 12. Contour plots for the dambreak problem at time $t = 10\,\mathrm{s}$ computed on a 80×80 grid (left) and a 160×160 grid (right).

TABLE 3. CPU times for the dambreak problem with grid refinement.

grid	refinement	CPU time (normalized)
80×80	yes	3.5
80×80	no	1.0
160×160	no	5.7

contains contour plots of the surface elevation computed on a 80×80 and a 160×160 grid without refinement. In all computations the CFL number was 1.5. All details in the solution on the 160×160 grid are captured using the refinement. The weak wave reflected from the upper wall is even slightly sharper in Figure 11.

Table 3 contains the CPU times for the three computations and shows that a gain in efficiency can be obtained by using grid refinement. We have performed several computations which show that the gain increases as the number of cells in the underlying grid increases.

Moreover, while the dimensional splitting code has been optimized, further improvements could probably be obtained by optimizing the grid refinement code.

4.4. **Time dependent motion in a parabolic basin.** There are not many exact solutions of the shallow water equations available, especially not for the case when the boundaries between water and dry land are moving. But in an article by Thacker [32] some exact solutions for shallow water flow in a parabolic basin are presented. The motion corresponds to normal mode oscillations for which the water surface rotates in the basin. For one of the solutions the water surface remains planar at all times. Here we compare this particular solution with the solution computed by front tracking. Since the bottom profile is variable, a source term is present on the right-hand side in the shallow water equations. To handle this term we use operator splitting. The source term only depends on the spatial variables x and y, and hence the resulting ordinary differential equation can be solved exactly.

In this example it is most convenient to work in nonconservative variables z_a, u and v. If we take $g = 1$, the shallow water equations can be written as

$$\begin{aligned}
\frac{\partial z_a}{\partial t} + \frac{\partial}{\partial x}[(z_a - z_b)u] + \frac{\partial}{\partial y}[(z_a - z_b)v] &= 0, \\
\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial z_a}{\partial x} &= 0, \\
\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial z_a}{\partial x} &= 0.
\end{aligned}$$

Furthermore, we assume that the bottom profile is parabolic, i.e.,

$$z_b(x, y) = D_0\left(\frac{x^2 + y^2}{L^2} - 1\right).$$

Under these assumptions Thacker found two basic solutions where the water surface remains planar and the velocities vary sinusoidally with time

$$\begin{aligned}
z_a &= 2A\frac{D_0}{L}\left(\frac{x}{L}\cos\omega t \pm \frac{y}{L}\sin\omega t + B\right), \\
u &= -A\omega\sin\omega t, \\
v &= \pm A\omega\cos\omega t, \\
\omega &= \sqrt{\frac{2D_0}{L^2}}.
\end{aligned}$$

A is here a constant that determines the amplitude of the motion, B determines the reference level and ω is the frequency. Since the shallow water equations are based on the assumption of a hydrostatic pressure distribution, the vertical acceleration must be negligible compared with g. The solutions are therefore only valid when $A \ll L^2/2D_0$.

For the numerical computations we have chosen the positive sign and $A = L/2$, $B = -A/2L$, $D_0 = 1$, $L = 2500$. Computations were made on a uniform Cartesian grid with 101 grid cells in each direction. Numerical and exact solutions along the plane $y = 0$ at different times during the first half period of the oscillation are shown in Figure 13. The

TABLE 4. L^1 errors and convergence rates for conserved variables.

grid	steps	L^1-error h	hu	hv	rate h	hu	hv
32×32	15	5.500e-02	8.945e-02	5.394e-02	—	—	—
64×64	30	2.620e-02	4.045e-02	2.738e-02	1.07	1.14	0.98
128×128	60	1.337e-02	2.045e-02	1.395e-02	0.97	0.98	0.97
256×256	120	6.815e-03	1.040e-02	7.109e-03	0.97	0.98	0.97
512×512	240	3.482e-03	5.399e-03	3.602e-03	0.97	0.95	0.98

TABLE 5. L^∞ errors and convergence rates for primitive variables.

grid	steps	L^∞-error h	u	v	rate h	u	v
32×32	15	1.611e-01	3.037e+00	1.199e+00	—	—	—
64×64	30	7.127e-02	3.227e+00	8.642e-01	1.18	-0.09	0.47
128×128	60	4.060e-02	1.209e+00	6.033e-01	0.81	1.42	0.52
256×256	120	2.270e-02	1.094e+00	3.870e-01	0.84	0.14	0.64
512×512	240	1.217e-02	1.013e+00	2.665e-01	0.90	0.11	0.54

velocities have been reset to zero in the dry-land region. Figure 14 displays the water level $h = z_a - z_b$ at three different locations along the plane $y = 0$. For the computations in Figure 13 we have used a fixed CFL number of 1.25, while in Figure 14 we have chosen a fixed time step $\Delta t = 55.5$. From Figure 13 we see that the numerical solutions approximate the exact water elevation quite well. Some inaccuracies are noticeable along the shore line. The second figure also shows excellent agreement between exact and numerical solution. However, the velocity is poorly represented near the shore line due to the use of dry-bed Riemann problems [25, 10].

This test case can be used to verify first-order convergence of the method with respect to the L^1 norm. To this end, we simulate the solution at time $t = T/4$ on a $2^N \times 2^N$ grid using $15 \cdot 2^{N-5}$ time steps for $N = 5, \ldots, 9$. Table 4 gives the L^1 errors for each conserved quantity together with the corresponding convergence rates. The L^1 errors are computed according to a four point numerical quadrature and normalized using the area of the computational domain. Velocities are reset to zero in the dry-land region. The convergence rates are given by the formulae

$$\text{rate} = \log_2\Big(\frac{\text{error}(\Delta x)}{\text{error}(\Delta x/2)}\Big).$$

Although the transition from water to dry land is not smooth, the convergence rates are of order one in the L^1 norm. Convergence rate of order one was also observed in the primitive variables.

It is also instructive to measure the convergence in the L^∞ norm. Table 5 gives errors and convergence rates for the primitive variables. We see that the water elevation converges

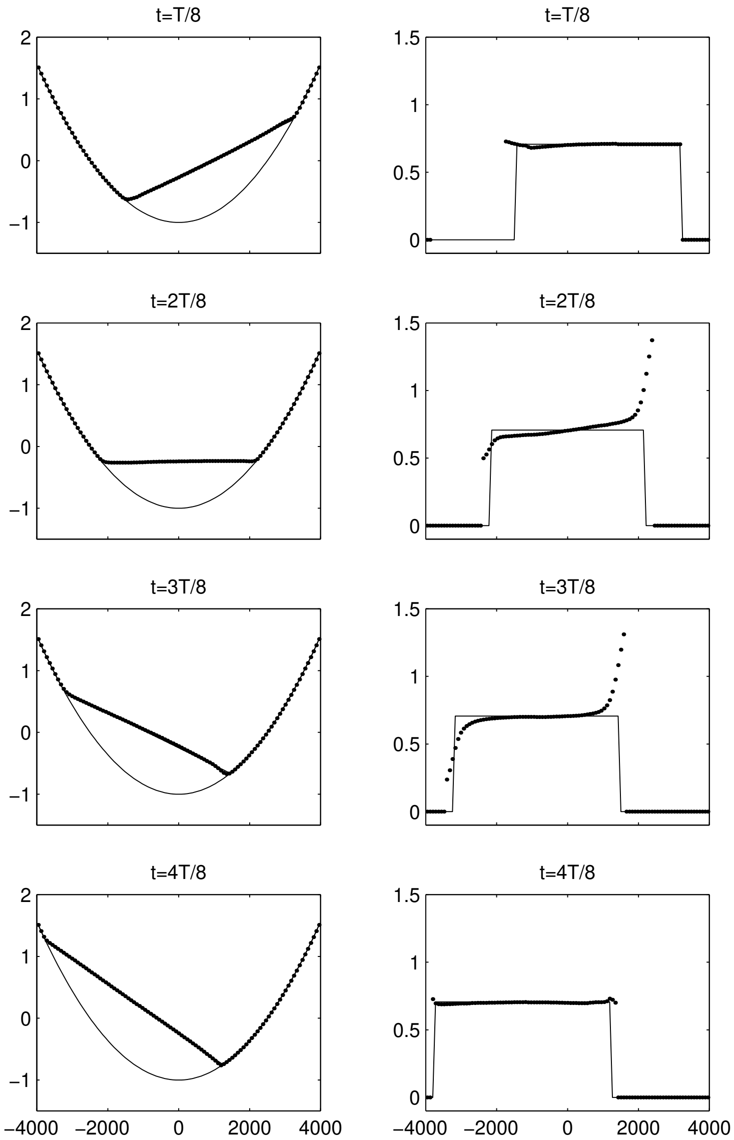

FIGURE 13. Front-tracking solution along the plane $y = 0$ for oscillations in a parabolic basin. T denotes the period. The solid line represents the exact solution. The left column shows water elevation and the right column the magnitude of the velocity.

nicely with order one (which was also observed for the other conserved variables). The zero velocity component (v) converges with order one half, while the nonzero velocity component (u) converges very slowly. This is in good correspondence with the observations made above for Figure 13.

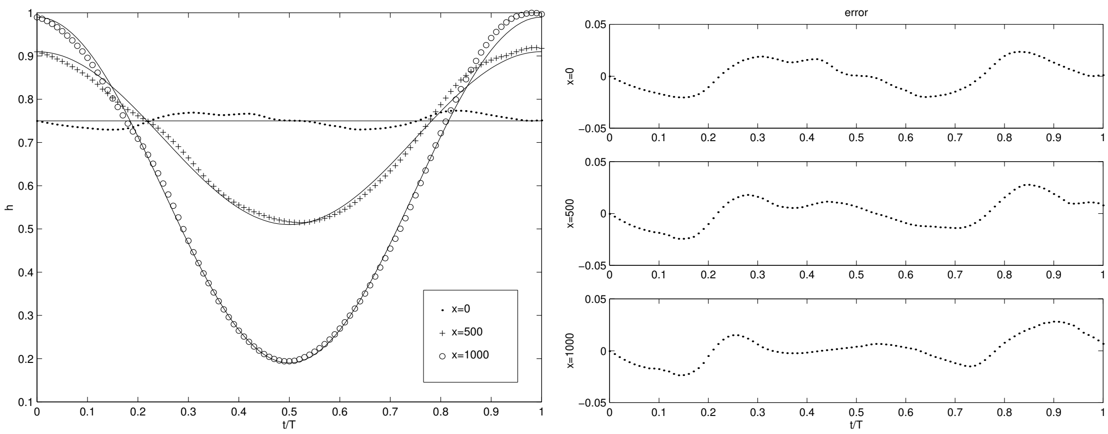

FIGURE 14. (Left) Water level at three different locations along the plane $y = 0$. (Right) Errors in the numerical solutions.

4.5. **Flow over an obstacle in one dimension.** In this example we study shallow water flow over a topography in one dimension. We use scaled equations, so that $g = 1$. The initial boundary value problem is described in [22], where it is solved by the composite scheme LWLF4. The spatial interval is $x \in [-10, 10]$ with absorbing boundaries, the initial conditions are

$$h(x,0) + z_b(x) = 1.0, \qquad u(x,0) = 1.0,$$

and the bottom profile reads

$$z_b(x) = \begin{cases} 0.2\left(1 - \frac{x^2}{4}\right), & -2 \le x \le 2, \\ 0, & \text{otherwise.} \end{cases}$$

First-order Godunov operator splitting (3) is used to handle the source term on the right hand side of the equations. To solve the homogeneous conservation law, we utilize front tracking in two ways. In the shock-capturing method, front-tracking solutions are projected onto a uniform grid at the end of each hyperbolic step. In the shock-tracking method (introduced above), we add extra grid points at the location of strong shocks.

Figure 15 shows solutions at times $t = 5$ and 10 computed with 100 grid cells by shock capturing, shock tracking, and by the composite scheme LWLF4. We use a fixed CFL number 1.75 for the two front-tracking methods and 0.8 for LWLF4. For reference, the solid lines give the solutions obtained on a 2000 grid for each method. In the shock-tracking method, only two extra grid cells were added. Although the shock-capturing method gives better resolution than LWLF4 on the same grid, it can by no means compete with the shock-tracking method. This method resolves both shocks within *one* grid cell.

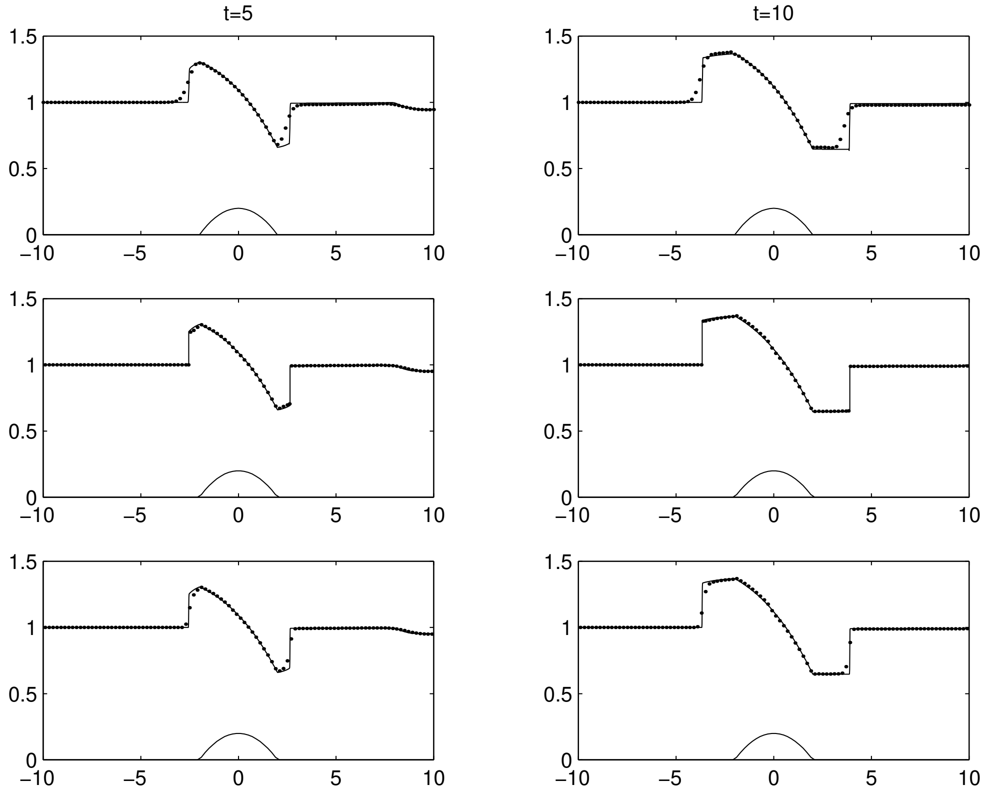

FIGURE 15. Shallow water flow over a topography in one-dimension computed by LWLF4 (top), the shock-tracking method (middle), and by the shock-capturing method (bottom).

5. Discussion and Conclusions

We have developed front tracking together with operator splitting as a numerical technique for the shallow water equations and tested it on four test cases.

For hyperbolic equations time stepping should be governed by the dynamics, and, in particular, should be fairly independent of the spatial discretization. Front tracking is unconditionally stable in this sense and may increase the computational speed. However, when used in conjunction with operator splitting one introduces errors due to the splitting as well as post-shock oscillations that effectively reduces the feasible CFL numbers to moderately above unity (i.e., typically in the interval 1–5). Similar observations have been made for the Euler equations [8].

By construction the method resolves shocks very sharply. Grid refinement enhances the efficiency of the method, and gives better resolution of sharp waves. For one-dimensional inhomogeneous problems (only one shown here) the resolution is greatly improved by introducing explicit shock tracking.

Our experience shows that the first-order front-tracking method compares well with second-order methods, see also [8, 20].

References

[1] F. Alcrudo and P. Garcia-Navarro. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *Internat. J. Numer. Methods Fluids*, 16:489–505, 1993.

[2] D. Ambrosi. Approximation of shallow water equations by Roe's Riemann solver. *Internat. J. Numer. Methods Fluids*, 20:157–168, 1995.

[3] A. Bermùdez, A. Dervieux, J. A. Désidéri, and M. E. Vázquez. Upwind schemes for the two-dimensional shallow water equations with variable depth using unstructured meshes. Rapport de recherche N^0 2738, INRIA, 1995.

[4] A. Bressan, G. Crasta, and B. Piccoli. Well-posedness of the Cauchy problem for $n \times n$ systems of conservation laws. *Mem. Amer. Math. Soc.* To appear.

[5] A. Bressan and P. LeFloch. Uniqueness of weak solutions to systems of conservation laws. *Arch. Rational Mech. Anal.*, 140(4):301–317, 1997.

[6] R. J. Fennema and M. H. Chaudhry. Explicit methods for 2-D transient free-surface flows. *J. Hydraul. Eng.*, 116:1013–1034, 1990.

[7] A. E. Gill. *Atmosphere–Ocean Dynamics.* Academic Press, New York, 1982.

[8] H. Holden, K.-A. Lie, and N. H. Risebro. A front tracking method for conservation laws in two dimensions. *J. Comput. Phys.* To appear.

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

[10] H. Holden and N. H. Risebro. *Front Tracking for Conservation Laws.* Department of Mathematics, Norwegian University of Science and Technology. Lecture Notes.

[11] G.-S. Jiang and E. Tadmor. Nonoscillatory central schemes for multidimensional hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 19(6):1892–1917, 1998.

[12] J. O. Langseth. On an implementation of a front tracking method for hyperbolic conservation laws. *Advances in Engineering Software*, 26(1):45–63, 1996.

[13] J. O. Langseth, N. H. Risebro, and A. Tveito. A conservative front tracking scheme for 1D hyperbolic conservation laws. In A. Donato and F. Oliveri, editors, *Nonlinear Hyperbolic Problems: Theoretical, Applied, and Computational Aspects (Taormina, 1992)*, volume 43 of *Notes Numer. Fluid Mech.*, pages 385–392. Vieweg, Braunschweig, 1993.

[14] R. J. LeVeque. Large time step shock-capturing techniques for scalar conservation laws. *SIAM J. Numer. Anal.*, 22:1051–1073, 1985.

[15] R. J. LeVeque. CLAWPACK software. Available from `netlib.att.com` in `netlib/pdes/claw/` or at the URL `http://www.amath.washington.edu/~rjl/clawpack.html`.

[16] R. J. LeVeque. CLAWPACK User Notes. Applied Mathematics, Box 352420, University of Washington, Seattle, WA 98195–2420, Nov. 1995. Available from `netlib.att.com` in `netlib/pdes/claw/doc/` or at the URL `http://www.amath.washington.edu/~rjl/clawpack.html`.

[17] R. J. LeVeque. Wave propagation algorithms for multidimensional hyperbolic systems. *J. Comput. Phys.* 131:327–353, 1997.

[18] R. J. LeVeque and K. M. Shyue. One-dimensional front tracking based on high resolution wave propagation methods. *SIAM J. Sci. Comput.*, 16(2):348–377, 1995.

[19] K.-A. Lie. *Front tracking and operator splitting for convection dominated problems.* Dr. ing. thesis, NTNU Trondheim, 1998.

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

[21] R. Liska, L. Margolin, and B. Wendroff. On nonhydrostatic two-layer models of incompressible flow. Preprint, 1996.

[22] R. Liska and B. Wendroff. Composite schemes for conservation laws. To appear in SIAM J. Numer. Anal.

[23] R. Liska and B. Wendroff. 2D shallow water equations by composite schemes. Technical Report LAUR-97-4879, LANL, Los Alamos, 1997.
[24] R. Liska and B. Wendroff. Analysis and computation with stratified fluid models. *J. Comput. Phys.*, 137(1):212–244, 1997.
[25] T.-P. Liu and J. Smoller. On the vacuum state for isentropic gas dynamics. *Adv. Pure Appl. Math.*, 1:345–359, 1980.
[26] A. T. Morel. *A Genuinely Multidimensional High-Resolution Scheme for the Shallow Water Equations.* PhD Thesis, Swiss Federal Institute of Technology, Zürich, 1997.
[27] A. N. Nazarov. Mathematical modeling of a snow-powder avalanche in the framework of the equations of two-layer shallow water. *Fluid Dynamics*, 26:70–75, 1991.
[28] N. H. Risebro. A front–tracking alternative to the random choice method. *Proc. Amer. Math. Soc*, 117(4):1125–1129, 1993.
[29] 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.
[30] N. H. Risebro and A. Tveito. A front tracking method for conservation laws in one dimension. *J. Comput. Phys.*, 101(1):130–139, 1992.
[31] B. K. Swartz and B. Wendroff. AZTEC: a front tracking code based on Godunov's method. *Appl. Numer. Math.*, 2(3-5):385–397, 1986.
[32] W. C. Thacker. Some exact solutions to the nonlinear shallow-water wave equations. *J. Fluid Mech.*, 107:499–508, 1981.
[33] E. F. Toro. Riemann problems and the WAF method for solving the two-dimensional shallow water equations. *Philos. Trans. Roy. Soc. London Ser. A.*, 338:43–68, 1992.
[34] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Mechanics.* Springer, Berlin, 1997.
[35] E. F. Toro and S. J. Billett. A unified Riemann-problem-based extension of the Warming-Beam and Lax-Wendroff schemes. *IMA J. Numer. Anal.*, 17:61–102, 1997.
[36] C. B. Vreugdenhil. Two-layer shallow-water flow in two dimensions, a numerical study. *J. Comput. Phys.*, 33:169–184, 1979.
[37] B. Wendroff. An analysis of front tracking for chromatography. *Acta Appl. Math.*, 30:265–285, 1993.

(Runar Holdahl)
Department of Mathematical Sciences,
Norwegian University of Science and Technology,
N–7034 Trondheim, Norway
E-mail address: `holdahl@math.ntnu.no`

(Helge Holden)
Department of Mathematical Sciences,
Norwegian University of Science and Technology,
N–7034 Trondheim, Norway
E-mail address: `holden@math.ntnu.no`
URL: `http://www.math.ntnu.no/~holden/`

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