Journal of Computational Physics **150,** 76–96 (1999)
Article ID jcph.1998.6164, available online at http://www.idealibrary.com on IDEAL®

An Unconditionally Stable Method for the Euler Equations

Helge Holden,[*,‡] Knut-Andreas Lie,[*,§] and Nils Henrik Risebro[†,¶]

[*]*Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway; and* [†]*Department of Mathematics, University of Oslo, P.O. Box 1053, Blindern, N-0316 Oslo, Norway*
E-mail: [‡]holden@math.ntnu.no, [§]andreas@math.ntnu.no, and [¶]nilshr@math.uio.no

Received April 21, 1998; revised September 30, 1998

We discuss how to combine a front tracking method with dimensional splitting to solve systems of conservation laws numerically in two space dimensions. In addition we present an adaptive grid refinement strategy. The method is unconditionally stable and allows for moderately high CFL numbers (typically 1–4), and thus it is highly efficient.

The method is applied to the Euler equations of gas dynamics. In particular, it is tested on an expanding circular gas front, a wind tunnel with a step, a double Mach reflection, and a shock–bubble interaction. The method shows very sharp resolution of shocks.

Key Words: front tracking; dimensional splitting; grid refinement; Euler equations.

1. INTRODUCTION

Front tracking has proved to be an efficient tool for analyzing hyperbolic conservation laws rigorously, both scalar equations and systems, in one space dimension.

We demonstrate how to use front tracking for systems of conservation laws as an unconditionally stable numerical method in two space dimensions, and we test it on the Euler equations of gas dynamics.

Let us first discuss the method of front tracking. Consider the hyperbolic conservation law

$$u_t + f(u)_x = 0, \qquad u|_{t=0} = u_0. \tag{1}$$

If we approximate the initial data by a step function, that is, a piecewise constant function, the problem is locally reduced to solving a series of problems with Riemann initial data, i.e., a single jump separating two constant states.

In the case of general systems, the solution of a Riemann problem is locally characterized by the Lax construction, where one has a set of wave curves forming a local system of coordinates in state space around two nearby constant states. Shocks and contact discontinuities are unchanged in the front tracking technique. However, if two states are connected by a rarefaction wave, we sample points along the rarefaction curve and approximate the (continuous) rarefaction wave by a (discontinuous) function with several small jumps. In this way the approximate solution is a step function for any fixed time. We denote all discontinuities in the solution as fronts. When two fronts collide we will have another local Riemann problem, which can be solved using the same construction. In this way the solution will remain a step function. Notice that there is no associated time step in the construction, and hence the method is unconditionally stable. For strictly hyperbolic systems and for sufficiently small initial data, one can prove that the approximate solution converges in L^1_{loc} to a weak solution of (1) as the sampled points approximate the rarefaction wave better and the approximate initial data approaches u_0; see Risebro [25] and Bressan and LeFloch [5]. A modification of this method, the wave front tracking method of Bressan, has recently been used to show stability, and thereby uniqueness, of solutions of (1); see [4, 5].

It is natural to explore the front tracking method numerically. Risebro and Tveito [26, 27] and Langseth *et al.* [19] have presented one-dimensional numerical implementions of front tracking for systems. In [26] front tracking is applied to the nonstrictly hyperbolic system of polymer flooding. Some examples of gas dynamics in one dimension are studied in [17, 27]. In [19], a modified version of the earlier schemes is introduced and the method is compared with Godunov methods. Implementation issues are discussed by Langseth in [17]. See also the related large time step method of LeVeque [21]. For a recent application of the front tracking method to the shallow water equations we refer to [12, 13].

A fundamental issue with the front tracking method is the potential buildup of infinitely many fronts in finite time. Analytically this is discussed in [2, 5, 25]. Numerically, it is treated as follows. At each Riemann problem we measure the total variation of each elementary wave. Employing preset user-defined cutoff values, small waves are eliminated, thus resulting in finitely many fronts globally in time.

Here we extend the application of front tracking as a numerical method to multidimensional problems by using dimensional splitting. The resulting method is unconditionally stable; that is, the time step is not restricted by the spatial discretization. Front tracking is intrinsically grid-independent. In order to use dimensional splitting, we introduce a rectangular Cartesian grid and solve the conservation laws in each coordinate direction, followed by a projection onto the grid. In the scalar case one can prove that the corresponding method converges to the unique solution; see Holden and Risebro [15]. Here we exclusively discuss the two-dimensional case, but conceptually the approach works in higher dimensions as well. The corresponding computer code is referred to as DIMSPLIT. Lie *et al.* [22] have recently developed an adaptive grid refinement in the context of front tracking for the scalar case. Here we implement grid refinement in the case of systems, and this code is denoted GRIDREF. Both codes include common boundary conditions like absorbing, periodic, Dirichlet, and reflective boundary conditions.

It is important to stress that front tracking, as the term is used here, differs from the front tracking technique used by Glimm and co-workers; see, e.g., [6]. The front tracking in [4, 25] is inherently a method for conservation laws in one space dimension, whereas the front tracking method of Glimm *et al.* in principle is a method that tracks discontinuities, or fronts, in several space dimensions. These fronts are treated as independent computational degrees of freedom. They are defined by the solution and develop dynamically with it. Therefore, an implementation of this method in several space dimensions is *much* more difficult than implementing dimensional splitting using front tracking (in our sense of the term). However, the dimensional splitting approach is more versatile and does not require any special coding for new initial/boundary value problems.

The random choice method [10] has been extended numerically to several space dimensions using dimensional splitting by Chorin [7] and Colella [8, 9]. However, they found that the method will not converge and reported $\mathcal{O}(1)$ errors near discontinuities due to the random sampling. Colella [9] suggested using a conservative, low-order method in the vicinity of discontinuities, and the Glimm scheme elsewhere, to reduce the problem.

Let us now turn to the discussion of the test cases. The first example, suggested by Toro [28], concerns an outward explosion caused by a circular region of high-density gas. The radial symmetry allows us to test the effects coming from the Cartesian grid. The front tracking method gives sharp resolution of shock fronts, but shows some minor grid orientation effects. Moreover, a convergence study indicates convergence to the nonsmooth solution of this problem at a rate of 0.7–0.8.

In the second example we study a wind tunnel with a small step which is hit by a planar shock with speed Mach 3; see Woodward and Colella [29]. Here the corner of the step is a singular point in the flow which is treated as in [29] by adjusting the physical values near the corner. The front tracking method resolves the major flow properties accurately. The adaptive grid refinement code GRIDREF reduces the runtime substantially compared to DIMSPLIT with the same accuracy.

The next example addresses the question of a planar shock with speed Mach 10 hitting a reflecting wall, resulting in the familiar double Mach reflection. Front tracking recovers the finer features of the flow, and again GRIDREF is faster.

Both the second and the third example are run on the same grids as those of Woodward and Colella [29], but with substantially higher CFL numbers.

In the final example we consider a planar shock hitting a circular region of gas with low density. A three-dimensional version of this problem was analyzed by Langseth and LeVeque [18]. The interaction creates complicated wave patterns. Front tracking is here compared with different wave propagation schemes from the CLAWPACK software [20] developed by LeVeque. We observe that front tracking is better than the first-order method and performs similarly to the second-order method with minmod limiter but uses only 1/10 of the CPU time.

Quirk [24] has identified certain fundamental problems with Godunov-type numerical methods, some of which we have encountered here. The front tracking method may generate nonphysical expansion shocks (as can be seen in Figs. 7–9). Such shocks are common to most difference and volume techniques. Moreover, generation of postshock oscillations requires the use of moderate CFL numbers. We have not been able to amend these problems in a completely satisfactory way.

Nevertheless, front tracking has proved to be a highly accurate and efficient numerical technique which compares well with second-order methods.

2. THE ONE-DIMENSIONAL FRONT TRACKING METHOD

We start by giving a brief explanation of the front tracking method; a thorough description can be found in, e.g., [14, 25, 27], and implementation issues are discussed in [17]. Consider the system of conservation laws given by (1). The idea behind front tracking is to construct an approximate solution within the class of piecewise constant functions. First we approximate the initial data u_0 by a step function (piecewise constant function) u_0^Δ, where Δ measures the approximation. This defines a family of local Riemann problems. The solution of each of these Riemann problems consists of constant states separated by elementary waves (rarefaction waves, shock waves, and contact discontinuities). For the Riemann problem (u^L, u^R) we write the solution as $u^L \overset{w_1}{\to} u^1 \overset{w_2}{\to} u^2 \cdots \overset{w_N}{\to} u^R$, where the notation $u^1 \overset{w}{\to} u^2$ means that the constant states u^1 (left) and u^2 (right) are connected by an elementary wave w.

The solution of the Riemann problem is scale invariant and we can express it as a function of $\xi = x/t$. The next step in the front tracking algorithm is to approximate the solution of each Riemann problem by a step function, where δ gives the approximation along rarefaction waves,

$$u_\delta(\xi) = u_i \qquad \text{for}\, \xi_i < \xi < \xi_{i+1}, i = 1, \ldots, N, u_0 = u^L, u_{N+1} = u^R. \tag{2}$$

Assume that we have approximated $u^L \overset{w_1}{\to} u^1 \cdots \overset{w_j}{\to} u^j$; that is, we have defined ξ_i for $i = 1, \ldots, k$ such that $u_k = u^j$; see Fig. 1. If wave $u^j \overset{w_j+1}{\longrightarrow} u^{j+1}$ is a shock (or a contact discontinuity) we define

$$u_{k+1} = u^{j+1}, \qquad \xi_{k+1} = \sigma(u^j, u^{j+1}),$$

where $\sigma(u^j, u^{j+1})$ is the shock speed given by the Rankine–Hugoniot condition. If w_{j+1} is a rarefaction, we define

$$u_{k+l} = \mathcal{R}_{j+1}(u^j; l\tilde{\delta}), \qquad \xi_{k+l} = \frac{1}{2}(\lambda_{j+1}(u_{k+l-1}) + \lambda_{j+1}(u_{k+l})), \quad l = 1, \ldots, M,$$

where $\mathcal{R}_{j+1}(u; \cdot)$ is the $(j+1)$th rarefaction curve emanating from u; $\lambda_{j+1}(u)$ is the $(j+1)$th eigenvalue of the matrix $f'(u)$; and $\lambda_{j+1}(u^{j+1}) - \lambda_{j+1}(u^j) = M\tilde{\delta}$, $\tilde{\delta} \approx \delta$. Here $u_{k+M} = u^{j+1}$. Using this approach, we get a sequence of constant states separated by moving discontinuities for each local Riemann problem. The discontinuities, which we will refer to as fronts, can now be collected globally. This is typically implemented as a linked list of data objects representing the fronts. We track the outgoing fronts up to the time of the first wave interaction. This interaction gives a new Riemann problem which we again can approximate by a step function, and so on. The front tracking algorithm thus consists of solving Riemann problems (rearranging the list of objects) and tracking fronts until they collide (updating a collision list).

Since the approximate solution of the Riemann problem is nonconservative, due to the replacement of the rarefaction wave by a step function, the overall method will not be conservative either. However, the conservation error is of order $\mathcal{O}(\delta)$ [19].

In order to avoid a possible infinite buildup of fronts, we have to do some data reduction. Several approaches have been suggested [2, 17, 19, 25]. Here we will apply a strategy inspired by Glimm's interaction estimate [10]; see [17]. Define the size of the jth wave connecting states u^j and u^{j+1} as $\Delta_j = \sum_{k=1}^{n} |u_k^{j+1} - u_k^j|/K_j$, where K_j is the number of components that are nonconstant over the wave. The wave structure of every Riemann problem is always computed, and wave j is included provided $\Delta_j > c\delta$, for a given positive constant c. In this way we will have only finitely many fronts for all time.

2.1. *Boundary Conditions*

During the tracking phase of the algorithm, a boundary is represented as a front with a special identifying tag. This way, the algorithm for computing possible collisions need not distinguish whether a front collides with a boundary or another front. Moreover, this easily allows for moving boundaries.

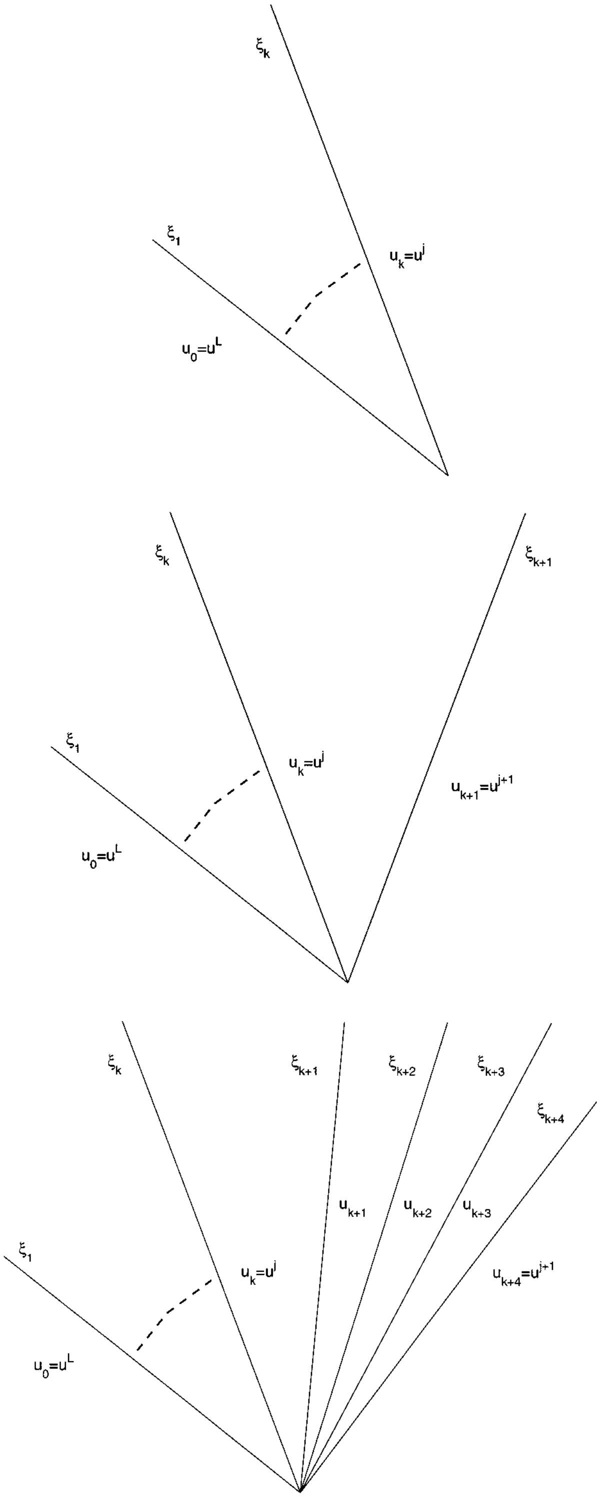

FIG. 1. Adding a new wave w_{j+1} to the approximate Riemann solution given the approximation of waves $w_1, \ldots, w_j$ (top); wave w_{j+1} is a shock or a contact discontinuity (middle) or a rarefaction wave (bottom).

In general there are three kinds of boundary conditions that may be imposed on any system. *Absorbing* boundaries allow the passage of waves without any effect on them and are realized by simply removing fronts that propagate out of the boundary and then updating the collision list. For *periodic* boundaries, the fronts that collide with a boundary are removed from one end of the front list and inserted at the other. Then the collision list is updated. *Dirichlet* boundary conditions mean that the boundary value is prescribed. In this case, Riemann problems are solved as above, with the fixed value as either left or right state. However, only fronts that propagate into the domain are inserted into the front list. For some systems it also makes sense to talk about *reflective* boundary conditions, for instance for the Euler equations or other systems with a velocity component. This case is handled as for Dirichlet boundaries, except that only one state is given in the Riemann problem. The other state is a fictitious state, determined from the known state inside the domain. For the Euler equations this fictitious state is obtained by reversing the sign of the velocity component. The same construction is used for *symmetry* about lines.

2.2. *Front Tracking with Projection*

The front tracking method described above is grid-independent in the sense that a grid is only used to describe the piecewise constant initial data. However, when the method is extended by operator splitting to equations with source terms [17] or to multidimensional problems by dimensional splitting (see Section 3) it is convenient to introduce a fixed Cartesian grid onto which the front tracking solution is projected. Unfortunately, this projection has certain undesired effects that we will discuss next.

The creation of postshock oscillations by numerical schemes has been studied by several authors (see, e.g., [1]), especially for slowly moving shocks. For the front tracking method (in one dimension) the mechanism behind this phenomenon is easily revealed; Fig. 2 shows

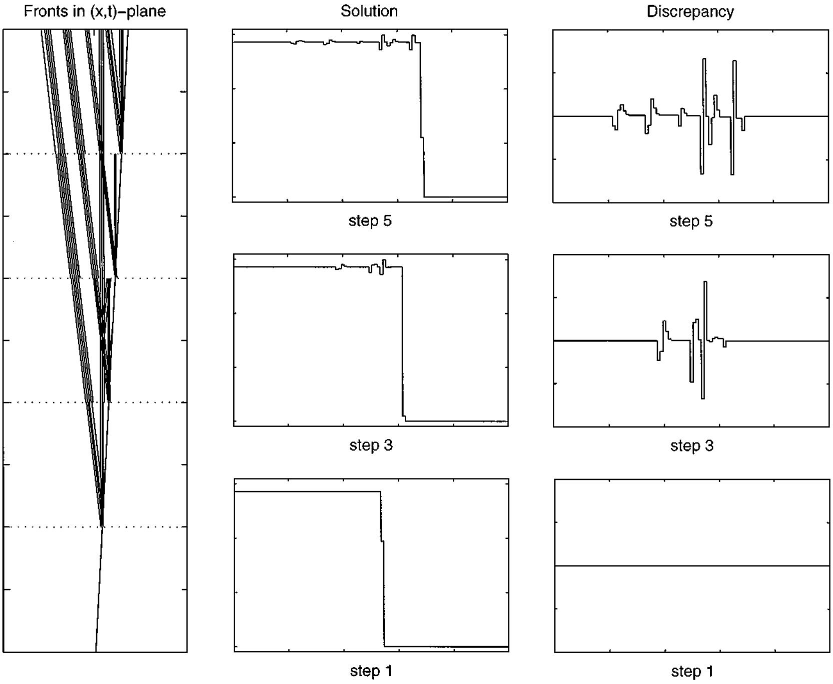

FIG. 2. Illustration of how postshock oscillations are created; fronts in (x, t)-plane (left), projected solution for each step (middle), and difference from the true solution projected onto the same grid (right).

the creation of postshock oscillations for a single propagating shock. When a shock is projected onto the grid, we introduce small waves in the passive families, unless the shock exactly traverses an integer number of grid cells in each time step. These waves produce (small) oscillations and increase the number of interactions that need to be resolved, thus slowing down the algorithm. Note that this phenomenon is observed for all shock speeds, not only slowly moving shocks (see also [1]).

For the results reported for multidimensional problems in Section 4, this effect seems to have little influence on the (visual) quality of the solutions. (In fact, we observed a more pronounced effect for one-dimensional problems.) At moderate CFL numbers, the (multidimensional) numerical diffusion introduced by the projections seems to dampen the oscillations, as does the data reduction.

3. TWO DIMENSIONS: DIMENSIONAL SPLITTING

For the two-dimensional conservation law

$$u_t + f(u)_x + g(u)_y = 0, \qquad u(x, y, 0) = u_0(x, y), \tag{3}$$

with entropy solution $u(t) = \mathcal{S}(t)u_0$, the dimensional splitting approximation is defined as

$$\mathcal{S}(n\Delta t)u_0(x, y) \cong [\mathcal{S}^g(\Delta t)\mathcal{S}^f(\Delta t)]^n u_0(x, y) \qquad \text{(Godunov)}$$

$$\mathcal{S}(n\Delta t)u_0(x, y) \cong [\mathcal{S}^f(\Delta t/2)\mathcal{S}^g(\Delta t)\mathcal{S}^f(\Delta t/2)]^n u_0(x, y) \qquad \text{(Strang)},$$

where $\mathcal{S}^f(t)u_0$ and $\mathcal{S}^g(t)u_0$ are the solutions of (1) with flux functions f and g, respectively. In numerical computations, $\mathcal{S}^f$ and $\mathcal{S}^g$ are replaced by some numerical method.

For scalar equations Holden and Risebro [15] proposed to combine Dafermos' method with dimensional splitting on a Cartesian grid to yield an *unconditionally* stable method. The algorithm is as follows: Solve along each row in the grid. Project the solution back onto the grid. Solve along each column of the grid, and so on. This method has been applied to simulate flow of hydrocarbons in a porous medium; see Bratvedt *et al.* [3]. The only result reported in the literature for a similar front tracking approach to systems is a simple test problem by Langseth [17] for the Euler equations of gas dynamics. (Preliminary results for the double Mach reflection problem were reported in [22].)

Lie *et al.* [22] recently observed that this front tracking method is highly efficient for scalar problems with absorbing boundary conditions, due to its lack of a CFL condition and the relatively simple dynamics of these problems. They also proposed a method for improving the spatial accuracy by using (one-level) adaptive grid refinement on a fixed regular Cartesian grid. The grid may contain a local regular partition at any cell; see Fig. 3. These subpartitions may appear or disappear during the projection step or remain fixed throughout the computation.

The inclusion of local refinement requires a reformulation of the front tracking algorithm. Inside each tube of coarse or refined grid blocks we will now have a local list of moving fronts (discontinuities). At the interface between a coarse and a refined grid block, we insert a special front, hereafter referred to as a static front. This front connects the coarse tube with the refined tubes; see Fig. 3. Altogether, this gives a global list, where each local list can be updated as usual except for the static fronts.

At interfaces initially, we solve Riemann problems as described in Fig. 4. For each refined tube, the Riemann problem is given by the value in the first refined cell and the connected

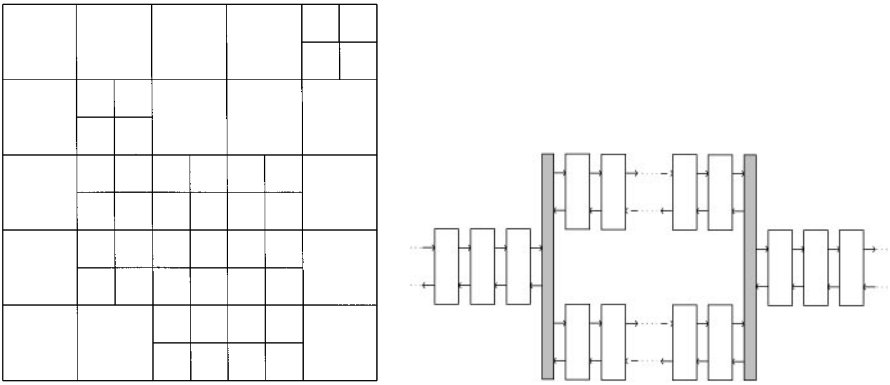

FIG. 3. (Left) Grid refinement by local partitioning of a regular Cartesian grid. (Right) Part of a front list containing two parallel refined tubes. The shaded boxes represent the static fronts.

coarse cell. 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 spatial average across the coarse tube, collected in a list of increasing wave speeds, and inserted into the local list for the coarse tube.

During the tracking, collisions at static fronts are handled as follows: Fronts coming in from a coarse part are copied and inserted into each connected refined list. Fronts coming in from refined tubes are collected, assigned new states according to the spatial average, and inserted into the connected coarse list; see Fig. 4.

The tracking step is followed by a projection step. In this step we measure the *one-dimensional* total variation of the front tracking solution and use this as a monitor function. A coarse grid block is refined if the variation inside the block in the direction we are solving exceeds $\mathrm{TV}_{\max}$. After the projection we postprocess the grid to remove unnecessarily refined blocks. If the *two-dimensional* total variation over all refined cells inside a coarse block is below $\mathrm{TV}_{\min}$, the block is made coarse. This is the adaptive part of the grid refinement, where $(\mathrm{TV}_{\min}, \mathrm{TV}_{\max})$ are two adjustable parameters. The choice of adaptivity criterion is not special and could, for instance, be replaced by a heuristic monitor function based on physical quantities. In addition it pays off to include some kind of extra postprocessing to reduce the number of interfaces between coarse and refined blocks. In this step some blocks are refined to make larger continuous patches of refined blocks. Moreover, one can include a preprocessing step to predict movement of refined structures (based on cheap estimates of wave speeds).

4. NUMERICAL RESULTS

The Euler equations form *the* most frequently used hyperbolic system for testing new numerical methods. In two dimensions they read

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{bmatrix}_x + \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}_y = 0. \tag{4}$$

Here ρ denotes the density, u and v denote the velocity in the x and y directions, p is the

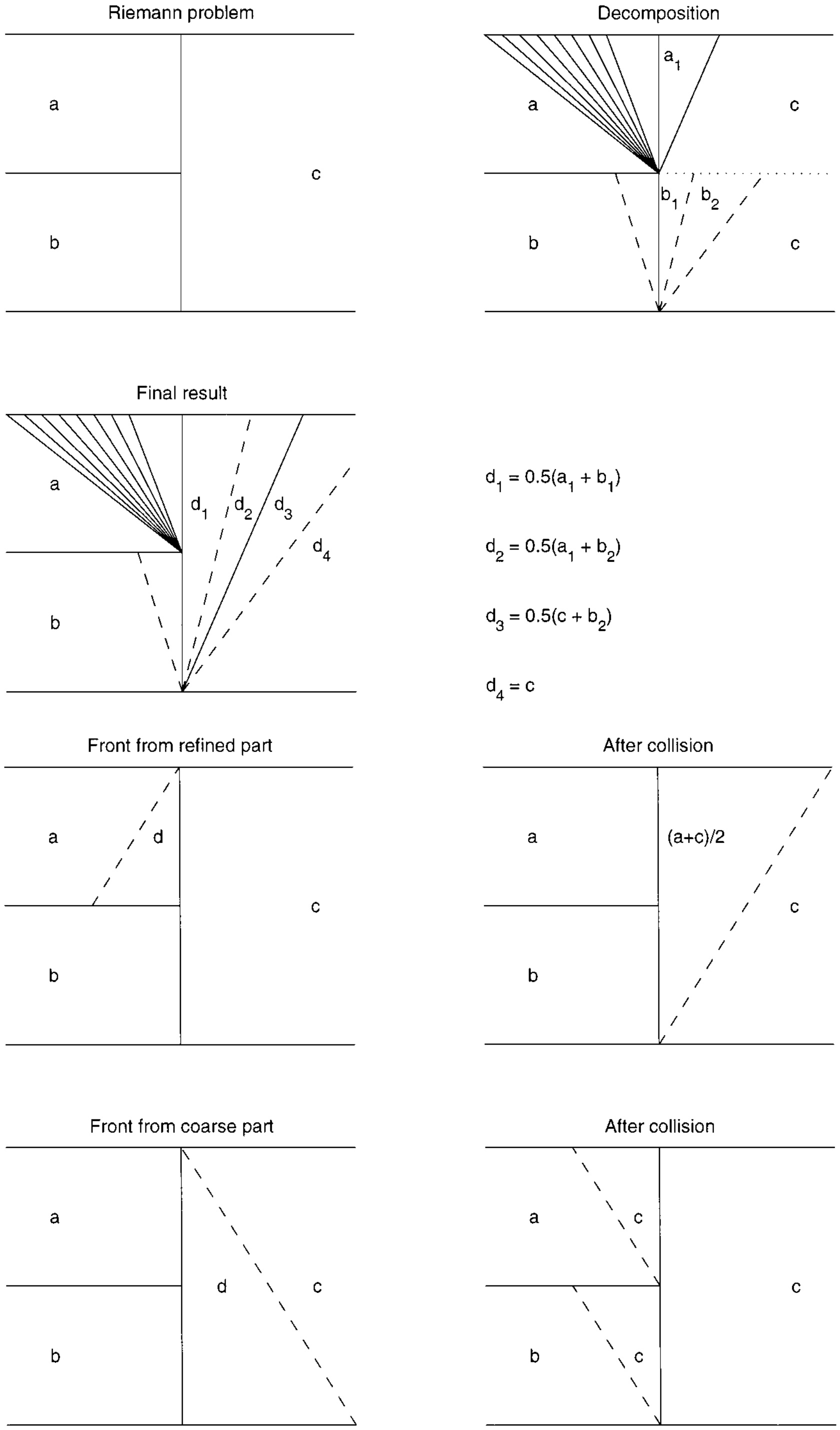

FIG. 4. (Top four) A Riemann problem at an interface between a coarse tube and two refined tubes is decomposed into two simple Riemann problems: solid lines represent (a, c) and dashed lines (b, c). Fronts propagating into coarse part are assigned new states by cross-sectional average. (Bottom four) The static front during tracking: Colliding fronts from the refined part are assigned new states. Colliding fronts from the coarse part are copied.

pressure, and E is the total energy (kinetic plus internal energy). We assume that the gas is ideal and polytropic. Then the energy is given by $E = \rho(u^2 + v^2)/2 + p/(\gamma - 1)$. In all computations we use $\gamma = 1.4$.

For this nonlinear system there are three elementary waves: shocks (S), rarefactions (R), and contact discontinuities (C). The possible wave configurations are SCS, SCR, RCR, and RCS. We will apply a very efficient Riemann solver reported by Gottlieb and Groth [11]. In this solver, all computations are performed in the nonconserved variables (p, u, v, a), where a denotes the sound speed $a^2 = \gamma p/\rho$. However, the projection step in our method proceeds in conserved variables. This induced mapping and remapping of the variables (according to explicit formulas) means a slight decrease in the efficiency of the code.

The eigenvalues of the one-dimensional version of (4) are u and $u \pm a$. These are easily computed and may be used as an efficient tool for predicting the movement of refined structures during a preprocessing step.

We consider four different test cases. First, a problem with cylindrical symmetry is used to evaluate grid alignment effects in the method. The next two problems, flow past a forward facing step and reflections at a wedge, were proposed by Woodward and Colella [29] and are well established as test cases in the literature. In the fourth problem we consider numerical viscosity and the generation of vortices when a planar shock interacts with a region of low density. Comparisons are made with wave propagation methods in CLAWPACK [20], which is coded in Fortran. All other methods are coded in C.

In the following, the discretization parameters will be equal in both spatial directions, unless stated otherwise. If local grid refinement is included, the ratio between the coarse and the refined grid size is 2 in each direction. In all examples we use a *uniform* Cartesian grid.[1] This is not a prerequisite, and the front tracking codes work on any regular Cartesian grid.

4.1. *A Cylindrical Explosion Problem*

Problems with cylindrical symmetry are good test examples for Cartesian grid methods, since reliable numerical solutions can easily be computed for the equivalent one-dimensional inhomogeneous problems. The following test case has been proposed by Toro [28]. Consider a square domain $[-1, 1] \times [-1, 1]$. The initial data are constant in two regions separated by a circle of radius 0.4 centered at the origin. Inside the circle we have $\rho_{\text{in}} = p_{\text{in}} = 1.0$ and zero velocity. Outside the circle $\rho_{\text{out}} = 0.125$, $p_{\text{out}} = 0.1$, and the velocities are zero. We impose absorbing boundary conditions.

The solution consists of a circular shock wave propagating outward from the origin, followed by a circular contact discontinuity propagating in the same direction, and a circular rarefaction wave traveling toward the origin. As time evolves, the shock wave becomes weaker. The contact discontinuity also becomes weaker, and at some time it stops and then travels inward. The rarefaction wave reflects at the center, as a rarefaction wave, and then overexpands and creates an inwardly propagating shock wave. The shock implodes into the origin, reflects, and travels outward, colliding with the contact discontinuity surface, and so on.

For the computations, we used a uniform 101 × 101 grid. The initial data were assigned according to the average over each cell. A reference solution was generated by solving the corresponding one-dimensional inhomogeneous problem on a fine grid, using front

[1] All simulations were carried out on a SUN Ultra 4 workstation with 1024 MB RAM and two 300-MHz CPUs.

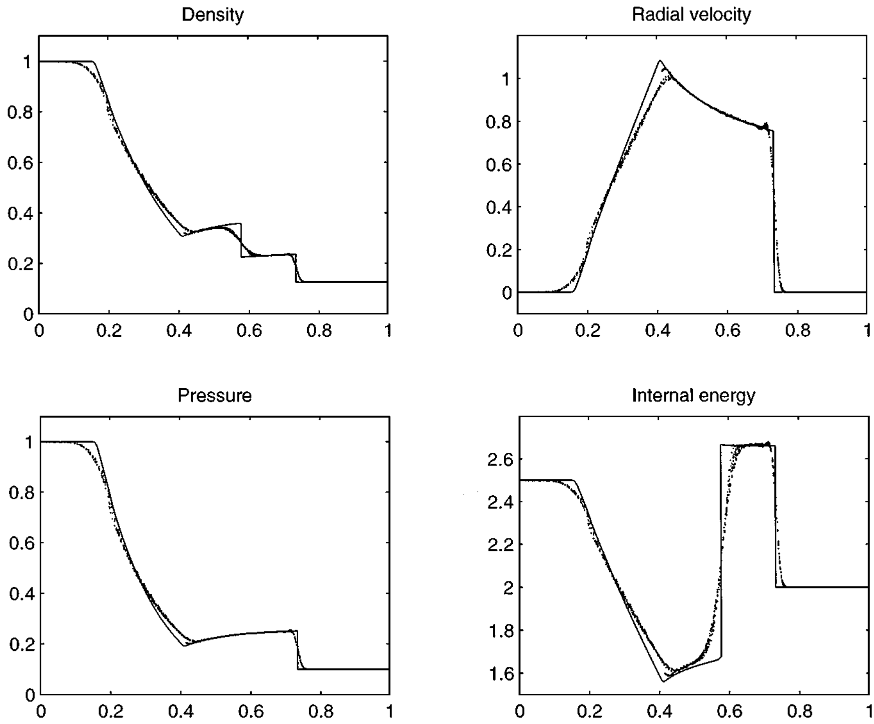

FIG. 5. Comparison between the one-dimensional radial solution (solid line) and a scatter plot (physical quantities are plotted against the distance from origin for each grid cell) of the two-dimensional DIMSPLIT solution at time $t = 0.2$.

tracking combined with operator splitting for the source term. Figure 5 shows a comparison of the one-dimensional radial solution at time $t = 0.2$ and a scatter plot the two-dimensional solution computed by DIMSPLIT with 10 time steps. This gives a CFL number varying between 1.2 and 2.2. The parameters for the Riemann solver are $\delta = 0.1$ and $c = 0.01$. As expected, the symmetry is not preserved perfectly. The shock wave is typically resolved within two grid cells and the contact discontinuity by three or four cells; see Fig. 6. The same problem can be used to investigate the order of convergence for the method. The solution is computed by DIMSPLIT with N time steps on a $10N \times 10N$ grid. In Table 1 the relative L^1 errors are given for $N = 5$, 10, 20, and 40. The errors are computed by comparing with the one-dimensional reference solution, using a standard four-point numerical quadrature for each grid cell. Convergence rates are estimated by the formula

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

Since the solution is nonsmooth, we cannot expect to retain first order convergence. However, the observed rates are well above 1/2; see [22] for a discussion of the scalar case.

4.2. *A Mach 3 Wind Tunnel with a Step*

The test case begins with a Mach 3 flow in a wind tunnel. The tunnel is 1 length unit high and 3 length units long. The step is 0.2 units high and is located 0.6 units from the

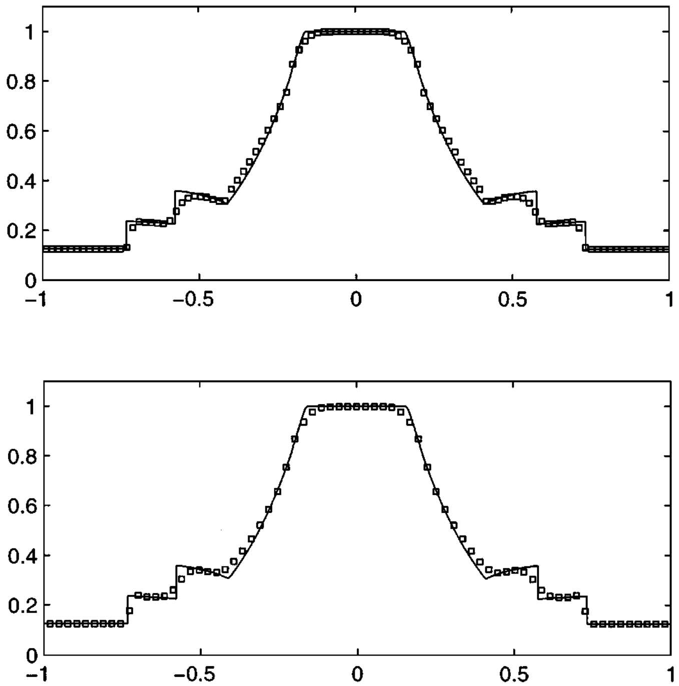

FIG. 6. Computed density along the lines $y = 0$ (upper) and $x = y$ (lower). The solid line represents the reference solution.

left-hand end of the tunnel. Inflow boundary conditions are assumed at the left-hand side, and absorbing conditions at the right-hand side. The walls are assumed to be reflective. The corner of the step is a singular point in the flow. We have adopted the technique proposed by Woodward and Colella [29] to reduce the influence of this point: We reset the values in six grid cells on top of the step, so that the entropy and the sum of enthalpy and kinetic energy per unit mass have the same values as in the grid cells just to the left and below the corner.

Figure 7 shows the density at time $t = 4.0$ computed by DIMSPLIT on three different grids ($\Delta x = 1/20$, $1/40$, $1/80$) corresponding to those used by Woodward and Colella [29]. The number of time steps is 160, 320, and 640, respectively, giving a CFL number approximately equal 2.0 in each step. The parameters for the Riemann solver are $\delta = 0.1$ and $c = 0.01$.

The general shape and position of the shocks are accurately represented by DIMSPLIT. The shocks are thin, and thus some numerical instabilities of strong shocks are evident at

TABLE 1
Runtime and Errors in Density, Radial Momentum, and Energy for the Radially Symmetric Problem

Grid	Steps	ρ	ρu_r	E	Runtime (s)
50	5	1.873×10^{-2}	4.372×10^{-2}	1.590×10^{-2}	0.4
100	10	1.124×10^{-2}	2.525×10^{-2}	9.022×10^{-3}	2.3
200	20	6.751×10^{-3}	1.475×10^{-2}	5.006×10^{-3}	15.3
400	40	4.124×10^{-3}	9.246×10^{-3}	2.931×10^{-3}	43.2
Averaged rate:		0.73	0.75	0.81	2.3

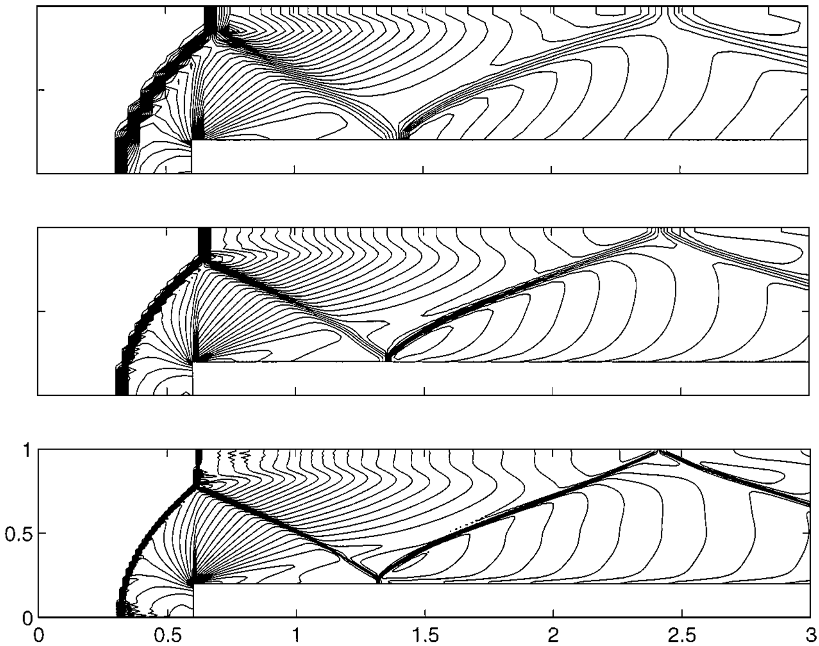

FIG. 7. Flow past a forward facing step computed by DIMSPLIT.

the bottom and behind the Mach stem, where the shocks are nearly aligned with the grid. The contact discontinuity emerging from the Mach stem is present on all grids, but is spread somewhat as it moves away from the three-shock interaction point. On the other hand, the weak shock emerging from the corner of the step and the discontinuity formed when this shock hits the reflected shock is only represented on the finest grid.

The results are slightly marred by an unphysical expansion shock embedded in the rarefaction fan at the step. The same effect is produced by Godunov's method [29]. Although the front tracking method coincides with Godunov's method for low CFL numbers (less than 1/2), they are distinct for the results reported here. For the runs in Fig. 7 the total number of wave interactions is 591,417, 3,942,300, and 28,601,224, respectively.

Increasing the number of time steps for a fixed spatial discretization leads to wider shock fronts and less accurate representation of the Mach stem; see Fig. 8. Decreasing the number of time steps gives sharper resolution of shock fronts and contact discontinuities, but also introduces more numerical instabilities, which gradually will destroy the solution. Choosing the appropriate number of time steps is therefore a subjective and problem-dependent decision, upon which we do not venture to give general advice. However, notice that with many short time steps, most of the computational time will be spent solving initial Riemann problems, whereas with few, but long, time steps, most of the computational time is spent resolving wave interactions. An optimum, with respect to runtime, is therefore obtained somewhere in between; see Table 2.

Similar results obtained by GRIDREF are shown[2] in Fig. 9. The number of time steps are 320, 640, and 1280, respectively, giving an approximate CFL number of 1.0 relative to the

[2] The contours are plotted in Matlab by patching contour plots on the coarse and the refined grid. As a result, some contour lines are discontinuous or entwine at interfaces between coarse and refined cells.

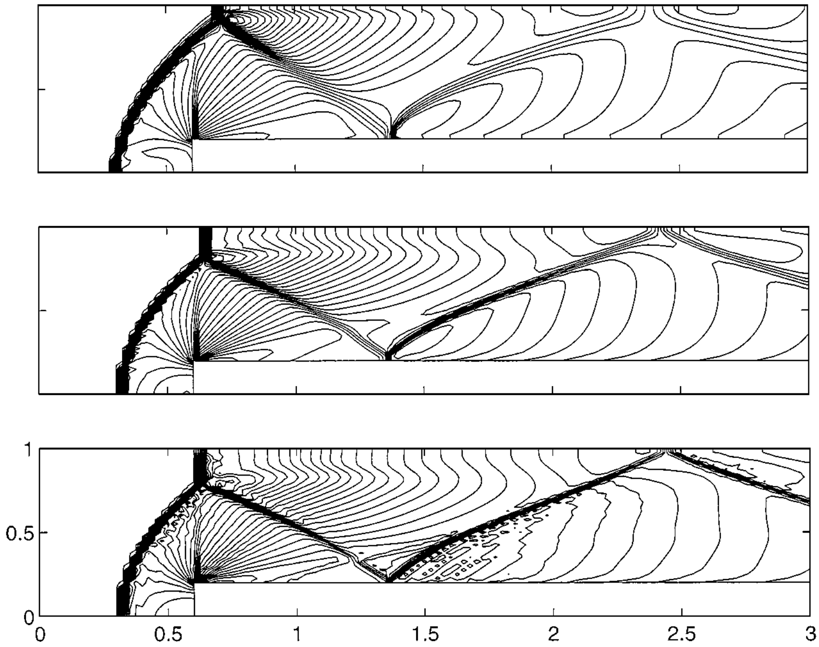

FIG. 8. The effect of different time steps in DIMSPLIT for $\Delta x = 1/40$; from top to bottom, 1280 (Godunov's method), 320, and 160 time steps.

coarse grid and 2.0 relative to the fine grid. The coarsest grid in Fig. 9 corresponds to the middle grid in Fig. 7, and the middle grid in Fig. 9 to the finest grid in Fig. 7. The TV threshold parameters used in the projection are $(\mathrm{TV}_{\min}, \mathrm{TV}_{\max}) = (0.5, 2.0)$. Initially the grid is refined in an L-shaped domain placed upside-down around the corner of the step.

We see that GRIDREF produces equally good results for the two runs where the size of the refined grid cells corresponds to those used by DIMSPLIT. The runtime, however, is reduced by approximately 30% on the coarse and 50% on the middle grid. On the finest grid we see that the rarefaction shock has nearly disappeared, and the contact discontinuity arising from the step is much better resolved. Figure 10 shows the adaptive grid after the projection at time $t = 4.0$. Note how the refinement neatly aligns with the major shocks. For fixed TV parameters, this effect becomes more pronounced as Δx decreases. This explains the larger improvement in efficiency for the middle grid.

4.3. *Double Mach Reflection of a Strong Shock*

This test problem describes the reflection of a planar Mach shock in air hitting a wedge. The setup is of a Mach 10 shock which initially makes a 60° angle with a reflecting wall.

TABLE 2

Runtimes in CPU Seconds for the Forward Facing Step

	Figure 7	Figure 8	Figure 9
Top	7.15×10^0	1.09×10^2	3.34×10^1
Middle	4.95×10^1	7.15×10^0	1.76×10^2
Bottom	3.81×10^2	6.68×10^1	8.83×10^2

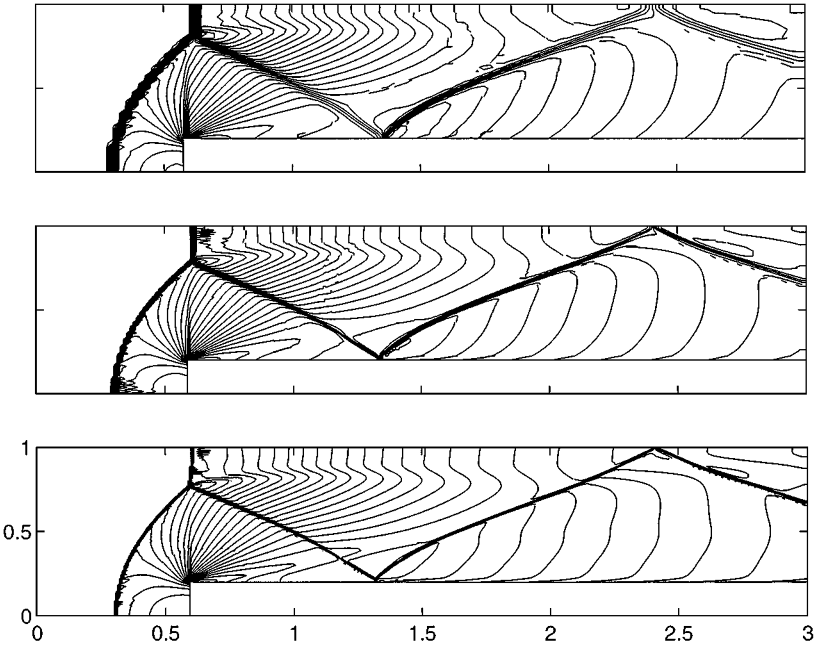

FIG. 9. Flow past a forward facing step computed by GRIDREF.

Ahead of the shock the undisturbed air has density 1.4 and pressure 1.0. The computational domain is [0, 4] × [0, 1] and the reflecting wall lies at the bottom of the domain starting at $x = 1/6$. The short region from $x = 0$ to 1/6 and the left boundary are assigned values for the initial postshock flow. Along the upper boundary, the flow values are set to describe the exact motion of the Mach 10 shock. At the right boundary, absorbing conditions are imposed. See [29] for a more detailed description of the setup.

Figure 11 shows the density at time $t = 0.2$ computed by DIMSPLIT on three different grids ($\Delta x = 1/30, 1/60, 1/120$) corresponding to those used by Woodward and Colella [29]. The number of (equally spaced) time steps is 35, 70, and 140, respectively, giving a CFL number varying between 2.0 and 3.5. The parameters for the Riemann solver are $\delta = 1.0$ and $c = 0.01$. The runtimes are 2.0, 12.8, and 82.3 CPU seconds, respectively.

The double Mach reflection and the jet produced by it are clearly discernible on the coarsest grid, and adequately described on the middle grid. The weak shock generated at the kink in the main reflected shock and the contact discontinuity emerging from the three-shock interaction are fairly broad. This is improved on the finest grid.

Similar results obtained by GRIDREF are shown in Fig. 12. The number of time steps is 70, 140, and 280, respectively, giving CFL numbers in the interval (1.0, 1.75) relative to the

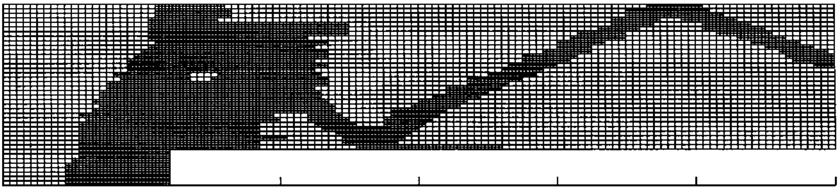

FIG. 10. The adaptive grid after the final projection for $\Delta x = 1/40$.

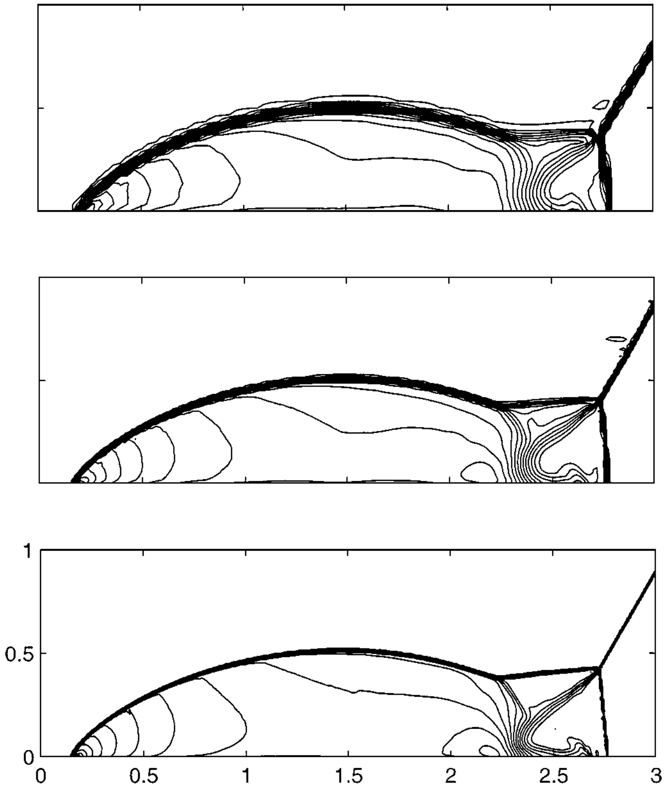

FIG. 11. A double Mach reflection at a wedge computed by DIMSPLIT.

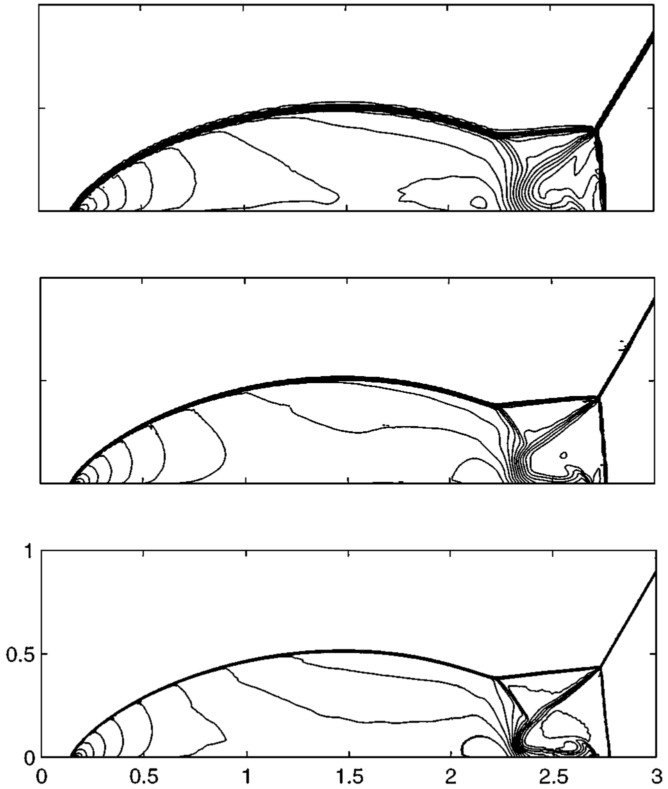

FIG. 12. A double Mach reflection at a wedge computed by GRIDREF.

coarse grid and (2.0, 3.5) relative to the fine grid. The TV threshold parameters used in the projection are $(\mathrm{TV}_{\min}, \mathrm{TV}_{\max}) = (5.0, 20.0)$. Initially the grid is refined around the shock. The runtimes are 11.8, 67.0, and 358 CPU seconds, respectively.

On the coarsest and the middle grid the solution is resolved as accurately as on the middle and the finest grid in Fig. 11. On the coarsest grid 54% of the grid cells are refined and there is only a slight reduction in runtime (8%). However, on the middle grid the fraction of refined cells is lower (35%), giving a 18% reduction in the runtime. On the finest grid all features in the solution are accurately described. Here 21% of the grid cells are refined. Notice that in all runs for both DIMSPLIT and GRIDREF, there is hardly a trace of numerical instabilities. However, the principal Mach stem is slightly kinked.

4.4. *A Shock–Bubble Interaction*

In this example we consider the interaction between a planar shock and a circular region of low density. The example is a two-dimensional version of a three-dimensional problem studied by Langseth and LeVeque [18]. The purpose is to illustrate the induced vorticity and mixing when a shock wave runs through an inhomogeneous medium.

The setup is as follows; cf. Fig. 13. A circle with radius 0.2 is centered at (0.3, 0.0). The gas is initially at rest and has unit density and pressure. Inside the circle the density is 0.1. The incoming shock wave starts at $x = 0$ and propagates in the positive x-direction. The pressure behind the shock is 10, giving a 2.95 Mach shock. The domain is $[-0.1, 1.5] \times [-0.5, 0.5]$ with symmetry about the x axis.

Figure 13 shows five snapshots at times $t = 0.0$ to $t = 0.4$, computed by DIMSPLIT with $\Delta x = 1/400$ and 256 equally spaced time steps. After hitting the bubble, the shock wave separates into a reflected smooth wave and a penetrating shock wave. Due to the higher sound speed inside the low density region, the latter wave will speed up toward the undisturbed bubble wall ahead, where it reflects. At time $t = 0.1$ the incident shock has captured the bubble and deformed it. A complex pattern of discontinuities has formed at the top and bottom of the bubble. Near the front wall we see the reflected wave, and near the back

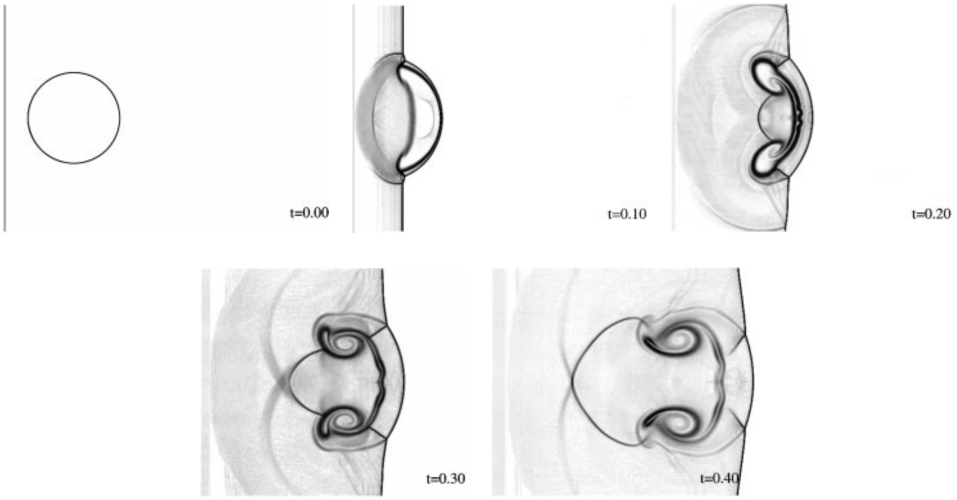

FIG. 13. Emulated Schlieren images of a shock–bubble interaction.

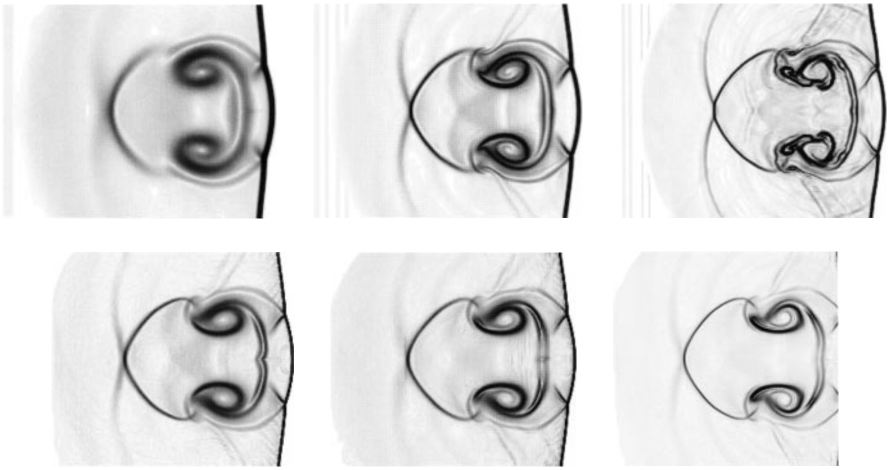

FIG. 14. Emulated Schlieren images of a shock–bubble interaction at time $t = 0.4$ on a $1/200$ grid. (Top) Simulations with CLAWPACK. Left: $(T^{1,1})$: 1058 s. Middle: $(T^{2,2})$, minmod: 1988 s. Right: $(T^{2,2})$, superbee: 2351 s. (Bottom) Simulations with DIMSPLIT and GRIDREF. Left: DIMSPLIT, $n = 128$: 185 s. Middle: GRIDREF, $n = 100$, $\Delta x = 1/100$: 141 s. Right: GRIDREF, $n = 200$, $\Delta x = 1/200$: 780 s.

wall the first traces of vortex formation. At time $t = 0.2$ the remnants of the bubble are contained inside two rotating semicircular vortex regions that are connected by a "duct." At time $t = 0.3$ the "duct" has closed, the vortices have separated, and new secondary vortices have formed.

Resolving the vorticity is a question of resolution and numerical viscosity. We find it futile to discuss the resolution on different grids. Instead we focus on the numerical viscosity in our two schemes. We compare our computations with computations using CLAWPACK [20] on the same grid. CLAWPACK is a collection of wave propagation methods where Riemann problems are solved at cell interfaces and limiters are applied to suppress oscillations from second-order terms. To model cross derivatives, Riemann problems are also solved in the transverse directions. The software allows the user to choose between first- and second-order and different limiters. Figure 14 shows computations by DIMSPLIT and GRIDREF, compared with the unsplit first-order method $(T^{1,1})$ in CLAWPACK and the unsplit second-order method $(T^{2,2})$ with minmod limiter (the most diffusive limiter) and superbee limiter (the most compressive). CLAWPACK $T^{1,1}$ is clearly the most dissipative method. CLAWPACK $T^{2,2}$ minmod and DIMSPLIT produce very similar results, but the latter gives a more narrow representation of the leading shock wave. GRIDREF produces equal results on a $1/100$ grid as DIMSPLIT does on the $1/200$ grid, but with a 25% reduction in runtime. The front tracking method spends most of the computational time where interactions take place and is thus highly effective on problems where interactions are restricted to a smaller part of the computational domain. Although CLAWPACK is not optimized for speed, the large differences in runtime give a good indication of the efficiency of the front tracking method. This efficiency can be further assessed by comparing with a scheme not based on Riemann problems, e.g., the second-order, central difference scheme of Jiang and Tadmor [16]. Even though this scheme has a very low complexity, the runtime was 10 times larger than for DIMSPLIT on the same grid. The results produced (using the UNO limiter) were slightly less dissipative than with CLAWPACK $T^{2,2}$ minmod.

4.5. *A Word of Caution*

Quirk [23, 24] has catalogued a number of instances where Godunov-type methods give unreliable results. Some of these shortcomings can also be observed when using the front tracking scheme.

The most serious is expansion shocks, as seen in Figs. 7–9. Quirk [24] presents an example of this problem for a strong shock diffracting around a 90° corner, where for instance Roe's method gives an expansion shock and fails to converge to the correct solution. The front tracking method performs similarly; in fact, the same problem is observed for shock diffraction around other geometries, for instance over a half-diamond [24]. For finite difference or volume methods this problem can be circumvented by locally using a more defective approximative Riemann solver such as HLLE [24]. This changes the flavor of the artificial dissipation that is implicit in the scheme and stabilizes the computations. The reason for this may be that the general initial/boundary value problem for the Euler equations is ill-posed in several space dimensions. If this is so, then schemes should rather approximate the more fundamental Navier–Stokes equations. Several authors have discussed possible instabilities arising from solving the exact Euler equations and the importance of introducing correct dissipation mechanisms; see, e.g., Xu [30]. So far, we have not found a proper workaround for the front tracking scheme.

Quirk [24] also discusses what he calls odd–even decoupling, which is seen for strong shocks nearly aligned with the grid. When solving Riemann problems in the transverse direction, where nothing should happen, small perturbations may grow unstably. Numerical experiments indicate that the data reduction in the algorithm counteracts this tendency, but does not eliminate it completely. If the perturbations exceed the cutoff value in the reduction, they grow unstably.

5. DISCUSSION AND CONCLUSIONS

Front tracking has proved to be a very efficient numerical method for one-dimensional problems and scalar problems in multidimensions. Here we have shown an unconditionally stable extension to multidimensional systems and tested it on the Euler equations.

As all shock capturing methods, the method generates postshock oscillations. However, the mechanism behind this phenomenon is easily explained for the one-dimensional method and was discussed in Section 2. In two dimensions, oscillations prevent the use of large time steps in the unconditionally stable method (as opposed to the scalar case [22]). However, for moderate CFL numbers (1–4) the numerical diffusion (and the data reduction) in the scheme seems to dampen the oscillations to an acceptable level.

A natural consequence of using the front tracking method is low-order and possible grid effects due to the dimensional splitting. Moreover, some deficiencies have been pointed out. Despite these limitations, the front tracking method produces surprisingly good results and gives very sharp resolution of shocks even on coarse grids. Similar observations have been made for the shallow water equations [12, 13] and the nonstrictly hyperbolic system describing polymer flooding. Comparisons show that the method performs similarly to some second-order methods with respect to accuracy but with higher efficiency.

The efficiency of the method may be improved by including one level of adaptive grid refinement. Here the total variation inside coarse grid cells has been used as a monitor function, but other choices are possible. Moreover, the method has a natural potential for parallel implementation.

ACKNOWLEDGMENTS

We thank Jan Olav Langseth for many valuable discussions. This research has been supported in part by the Research Council of Norway under Grant 100990/420. The second author has been supported by Grant 100555/410 from the Research Council of Norway.

REFERENCES

1. M. Arora and P. L. Roe, On postshock oscillations due to shock capturing schemes in unsteady flows, *J. Comput. Phys.* **130**, 25 (1997).
2. P. Baiti and H. K. Jenssen, On the front tracking algorithm, *J. Math. Anal. Appl.* **217**, 395 (1998).
3. F. Bratvedt, K. Bratvedt, C. F. Buchholz, T. Gimse, H. Holden, L. Holden, and N. H. Risebro, Frontline and Frontsim, two full scale, two-phase, black oil reservoir simulators based on front tracking, *Surv. Math. Ind.* **3**, 185 (1993).
4. A. Bressan, G. Crasta, and B. Piccoli, Well-posedness of the Cauchy problem for $n \times n$ systems of conservation laws, *Mem. Am. Math. Soc.*, to appear.
5. A. Bressan and P. LeFloch, Uniqueness of weak solutions to systems of conservation laws, *Arch. Rational Mech. Anal.* **140**, 301 (1997).
6. I.-L. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv, Front tracking for gas dynamics, *J. Comput. Phys.* **62**, 83 (1986).
7. A. J. Chorin, Random choice solutions of hyperbolic systems, *J. Comput. Phys.* **22**, 517 (1976).
8. P. Colella, *An Analysis of the Effect of Operator Splitting and of the Sampling Procedure on the Accuracy of Glimm's Method* (Ph.D. thesis, University of California at Berkeley, 1979).
9. P. Colella, Glimm's method for gas dynamics, *SIAM J. Sci. Statist. Comput.* **3**, 76 (1982).
10. J. Glimm, Solutions in the large for nonlinear hyperbolic systems of equations, *Comm. Pure Appl. Math.* **18**, 697 (1965).
11. J. J. Gottlieb and C. P. T. Groth, Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases, *J. Comput. Phys.* **78**, 437 (1988).
12. R. Holdahl, *Front Tracking for the Shallow Water Equations* (Diploma thesis, Department of Mathematical Sciences, Norwegian University of Science and Technology, 1998).
13. R. Holdahl, H. Holden and K.-A. Lie, Unconditionally stable splitting methods for the shallow water equations, *BIT*, to appear.
14. H. Holden and N. H. Risebro, *Front tracking for conservation laws*, lecture notes, Department of Mathematics, Norwegian University of Science and Technology.
15. H. Holden and N. H. Risebro, A method of fractional steps for scalar conservation laws without the CFL condition, *Math. Comp.* **60**, 221 (1993).
16. G.-S. Jiang and E. Tadmor, Non-oscillatory central schemes for multidimensional hyperbolic conservation laws, *SIAM J. Sci. Comput.* **19**, 1892 (1998).
17. J. O. Langseth, On an implementation of a front tracking method for hyperbolic conservation laws, *Adv. Engrg. Software* **26**, 45 (1996).
18. J. O. Langseth and R. J. LeVeque, A wave propagation method for three-dimensional hyperbolic conservation laws; available at www.math.ntnu.no/conservation/.
19. J. O. Langseth, N. H. Risebro, and A. Tveito. A conservative front tracking scheme for 1D hyperbolic conservation laws, in *Nonlinear Hyperbolic Problems: Theoretical, Applied, and Computational Aspects* (A. Donato *et al.*, Eds.), Notes Numer. Fluid Mech., Vol. 43, p. 385 (Vieweg, Braunschweig, 1993).
20. R. J. LeVeque, CLAWPACK software; available from netlib.att.com at netlib/pdes/claw/ or at www.amath.washington.edu/∼rjl/clawpack.html.
21. R. J. LeVeque, A large time step generalization of Godunov's method for systems of conservation laws, *SIAM J. Numer. Anal.* **22**, 1051 (1985).
22. K.-A. Lie, V. Haugse, and K. H. Karlsen, Dimensional splitting with front tracking and adaptive grid refinement, *Numer. Methods Partial Differential Equations* **14**, 627 (1998).

23. J. J. Quirk, Godunov-type schemes applied to detonation flows, in *Combustion in High-Speed Flows* (J. Buckmaster *et al.*, Eds.), p. 575 (Kluwer, Dordrecht, 1994).
24. J. J. Quirk, A contribution to the great Riemann solver debate, *Internat. J. Numer. Methods Fluids* **18**, 555 (1994).
25. N. H. Risebro, A front-tracking alternative to the random choice method, *Proc. Am. Math. Soc.* **117**, 1125 (1993).
26. N. H. Risebro and A. Tveito, Front tracking applied to a nonstrictly hyperbolic system of conservation laws, *SIAM J. Sci. Statist. Comput.* **12**, 1401 (1991).
27. N. H. Risebro and A. Tveito, A front tracking method for conservation laws in one dimension, *J. Comput. Phys*. **101**, 130 (1992).
28. E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics* (Springer-Verlag, Berlin, 1997).
29. P. Woodward and P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys*. **54**, 115 (1984).
30. K. Xu, Gas evolution dynamics in Godunov-type schemes; available at www.math.ntnu.no/conservation/.