JOURNAL OF COMPUTATIONAL PHYSICS **123,** 354–368 (1996)
ARTICLE NO. 0029

Two-Dimensional Front Tracking Based on High Resolution Wave Propagation Methods

RANDALL J. LEVEQUE*,[1] AND KEH-MING SHYUE†,[2]

*Departments of Mathematics and Applied Mathematics, University of Washington, Seattle, Washington 98195 and †Department of Mathematics, National Taiwan University, Taipei, China

Received October 27, 1994; revised July 3, 1995

We present a fully conservative, high resolution approach to front tracking for nonlinear systems of conservation laws in two space dimensions. An underlying uniform Cartesian grid is used, with some cells cut by the front into two subcells. The front is moved by solving a Riemann problem normal to each segment of the front and using the motion of the strongest wave to give an approximate location of the front at the end of the time step. A high resolution finite volume method is then applied on the resulting slightly irregular grid to update all cell values. A "large time step" wave propagation algorithm is used that remains stable in the small cut cells with a time step that is chosen with respect to the uniform grid cells. Numerical results on a radially symmetric problem show that pointwise convergence with order between 1 and 2 is obtained in both the cell values and location of the front. Other computations are also presented. © 1996 Academic Press, Inc.

1. INTRODUCTION

We will describe a fairly simple approach to front tracking in two space dimensions, giving a general formulation and then concentrating on shock tracking for the Euler equations of gas dynamics. The method is fully conservative and based on modern high-resolution shock capturing methods. An underlying uniform Cartesian grid is used, with some rectangles subdivided into two or more computational cells where discontinuities in the solution are expected (see Fig. 9 for an example grid). A high resolution finite volume method is applied on the resulting grid, based on the solution of Riemann problems and appropriate slope limiters. This method is implemented in a "wave-propagation" form, as developed in [31, 35], for example. The cell average at the end of a time step is computed from the cell average at the beginning of the step with modifications due to all waves that enter the cell. The waves come from solving one-dimensional Riemann problems at each cell interface in the direction normal to the interface. Transverse propagation of these waves is also introduced to improve numerical stability and incorporate the cross-derivative terms needed for second-order accuracy.

[1] E-mail: rjl@amath.washington.edu.
[2] E-mail: shyue@math.ntu.edu.tw.

An advantage of the wave propagation form is that reasonable time steps can be taken even if some of the subcells created by the interface are orders of magnitude smaller than the uniform Cartesian cells. Uniform time steps are used throughout the computation, with the time step chosen so that the Courant number is near one, relative to the size of the uniform grid cells. The method remains stable despite the small cells near the front. This is discussed further in Section 3.

The grid is modified in each time step based on the solution of Riemann problems across the interfaces representing the front. The motion of strong waves in these Riemann solutions gives an indication of the motion of the front. One major feature of the method is that the subdivision of cells is not assumed to give the definitive location of the true shock. It is viewed rather as an approximate location yielding a "refined grid" that is better able to represent the discontinuous solution than the Cartesian grid alone. Since the high resolution method is fully conservative and capable of capturing shocks even on a uniform grid, this gives a robust method for achieving increased resolution. Numerical results indicate that this can be quite successful and that, in fact, the grid interfaces can do a very good job of tracking a shock.

In previous work [40] (see also [29, 33]) we have developed a one-dimensional shock tracking algorithm based on the same principles and demonstrated that it gives very good results on a variety of problems, including unstable detonation waves.

In one space dimension, collisions of fronts are easily handled simply by adjusting the time step so that any collision occurs at the end of a time step and is then properly resolved in the solution to the Riemann problem at the start of the next time step. This has been used very successfully in the one-dimensional version of this algorithm developed in [40], but unfortunately does not extend to two dimensions. In the current two-dimensional implementation of our method we have assumed a simple structure in the front, namely a single connected curve with no self-intersections. This is clearly a limitation in regard to many

0021-9991/96 $18.00
Copyright © 1996 by Academic Press, Inc.
All rights of reproduction in any form reserved.

interesting problems where fronts collide with one another, join together at triple points, or split up into disconnected pieces. We have not yet tackled many of the topological difficulties, although some preliminary experiments with triple points in Section 5 indicate that this approach can be applied to deal with intersecting discontinuities also.

There are, moreover, many applications where the front topology remains simple and the approach described here is directly applicable to obtain high resolution results. Although for pure gas dynamics the most interesting problems often involve complicated shock structures with interactions and reflections, for these problems there are numerous high resolution ''shock capturing'' methods available that often do a very good job on a fixed grid, avoiding the complication of shock tracking altogether. The real need for tracking seems to be in problems where more complicated phenomena near the front require tracking in order to efficiently obtain high quality results. While ultimately there is a need to handle complicated topologies as well as more complicated phenomena at the front, there are already very interesting problems with simple front topologies. Examples include detonation waves (e.g., [5]), unstable interface problems (such as Rayleigh–Taylor and Kelvin–Helmholz instabilities), and viscous fingering phenomena in porous media flow. Some preliminary results obtained with our approach on unstable interfaces and porous media flow are given in [54, 55] and work is continuing on these problems.

The main purpose of the present paper is to explain in some detail our basic approach to conservative front tracking and to demonstrate that it is viable and promising in two space dimensions. This will be done in the context of the Euler equations of gas dynamics with shock tracking. We hope that this new set of tools can be applied in the development of more powerful algorithms for a variety of interface tracking problems.

Numerous other approaches to front tracking have been proposed over the years and we will mention only a few basic approaches. Many tracking methods are based on advancing the front, using the Rankine–Hugoniot conditions, or other appropriate jump conditions, and then using a nonconservative method on each side of the front. In theory it is necessary to worry about conservation only in the case where discontinuities are being captured, since a lack of conservation may lead to incorrect weak solutions. As long as all discontinuities are properly tracked, the smooth flow can be accurately represented with a nonconservative method. Early methods of this form can be seen in [17, 43, 53].

Glimm and coworkers have developed a very extensive set of tools for shock and interface tracking in two space dimensions with this type of approach that have been successfully applied to a wide variety of problems (e.g., [10, 15, 19–24]). This package includes procedures to deal with complicated interactions of interfaces, Mach triple points, and other such structures despite the lack of conservation at the interface and is currently being extended to three space dimensions.

Our feeling is that it is better to maintain conservation if possible, both to insure that discontinuities are really in the correct locations and to increase the robustness by allowing the possibility that some shocks are captured while others are tracked (see the ramp examples in Section 5, for example). Also, in some calculations, with interfaces between different fluids, for example, it may be important to exactly conserve the mass of each fluid. With our approach it does not seem to be difficult to couple high resolution conservative methods with propagation of the front using the Rankine–Hugoniot condition, which comes out automatically from the solution of the Riemann problem at the front.

In this sense our method is closest to that of Chern and Colella [9]. They use a volume of fluid (VOF) approach to keep track of the front location and then use high resolution conservative finite volume methods on the resulting grid. See also the work of Bell, Colella, and Welcome [1]. The key difference is that they use a more traditional flux differencing form, together with a flux redistribution algorithm to maintain stability in the presence of small cells. Bourlioux [5] has obtained very nice results for unstable two-dimensional detonation waves using Chern and Colella front-tracking, coupled with adaptive mesh refinement.

We believe that our wave-propagation form gives a more physical propagation of information near the front. We also explicitly track the front location, maintaining a data structure similar to what is used by Glimm's group (in the simple case of a single front); a list of points that, when joined by curves, give the location of the front. These points are moved in each time step as the front propagates.

It would be possible to combine our wave-propagation approach to updating the solution with a different representation of the front location. Other techniques may have advantages, especially in dealing with complex topologies. The volume of fluid approach used by Chern and Colella (see also [6, 11, 26, 45, 46]) is one possible approach, in which an additional scalar field is introduced that measures the fraction of each cell that is on one particular side of the front. A differential equation is developed for the advection of this scalar in each time step, and the front is reconstructed in each time step from the information stored in this field. Work by Swartz [56] and Pilliod and Puckett [49] shows that it is possible to obtain second-order accuracy with this type of method and appropriate reconstructions.

Another promising approach for maintaining information on the location of the front is the level set approach (e.g., [44, 48]). Here an additional scalar field $\psi(x, y, t)$ is

introduced in such a way that the front location is given by a particular level set of this function, say $\psi = 0$. Again an additional equation is needed for the evolution of ψ. A major advantage of this approach is that for many applications changes in topology can be handled quite easily.

Many other multi-dimensional front-tracking algorithms have been developed and several groups are actively involved in this area of research. See, for example [7, 8, 16, 25, 27, 38, 42, 46, 57, 58]

Outline of the Paper. Our algorithm has two main steps:

1. Given the grid at time t_n, determine the new grid at time $t_{n+1} = t_n + k$, the end of the time step.

2. Given the old and new grids and the numerical solution U^n on the old grid at time t_n, apply a finite volume method to determine U^{n+1} on the new grid at time t_{n+1}. Each numerical solution consists of cell averages over grid cells on the corresponding grid.

We will first discuss the finite volume methods needed for the second step. The main ideas of the wave-propagation approach used here are reviewed in Section 2 and the extension of this method to irregular grid cells near the front is described in Section 3. The front propagation algorithm is then described in Section 4. The results of some numerical tests are given in Section 5.

2. FINITE VOLUME METHODS BASED ON WAVE PROPAGATION

We will first briefly describe the basic high resolution method in wave-propagation form on a uniform Cartesian grid. See [31, 33, 34] for more details. This method will then be extended to the case of nonuniform grids created when some small cells are subdivided by the tracked front. The method is based on solving Riemann problems at each cell interface and using the resulting waves to update the solution in neighboring grid cells.

Let U_{ij}^n represent the cell average in grid cell (i, j) at time t_n and consider the interface between cells $(i-1, j)$ and (i, j), for example. We solve the one-dimensional Riemann problem normal to this face, which in this case will be

$$u_t + f(u)_x = 0, \tag{1}$$

with initial data given by $U_{i-1,j}^n$ and U_{ij}^n. Roe's approximate Riemann solver [52] is used, in which a linear problem $u_t + Au_x = 0$ is solved with a matrix $A = A(U_{i-1,j}^n, U_{ij}^n)$ that depends on the Riemann data and is a suitable approximation to the Jacobian matrix. The result is that the jump $U_{ij}^n - U_{i-1,j}^n$ is decomposed into eigenvectors of the matrix A,

$$U_{ij}^n - U_{i-1,j}^n = \sum_{p=1}^{m} \alpha_p r_p, \tag{2}$$

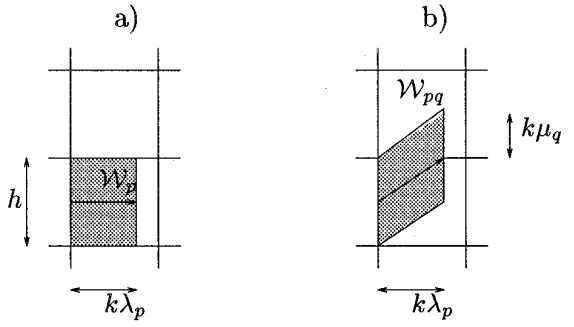

FIG. 1. (a) The wave propagating from the left boundary of the cell moves distance $k\lambda_p$ into the cell. (b) With transverse propagation, the wave is split into pieces traveling also with vertical velocity. The qth piece is shown, moving upwards distance $k\mu_q$.

where $r_p \in \mathbb{R}^m$ is the pth eigenvector of A and $Ar_p = \lambda_p r_p$ with $\lambda_p \in \mathbb{R}$ the corresponding eigenvalue. The scalars α_p give the wave strengths.

If $\lambda_p < 0$ then the pth wave propagates into cell $(i-1, j)$ while if $\lambda_p > 0$ it propagates into cell (i, j). In the simplest version of this method, a two-dimensional extension of Godunov's method, the cell average in the cell affected by this wave is simply updated by $(kh\lambda_p/h^2)\alpha_p r_p$. Note that the quantity in parentheses is simply the fraction of the cell swept out by this wave (see Fig. 1a).

Although this method is not written in "flux-differencing" form, it can easily be shown to be fully conservative provided that

$$f(U_{ij}^n) - f(U_{i-1,j}^n) = \sum_{p=1}^{m} \alpha_p \lambda_p r_p. \tag{3}$$

This is a fundamental property of the Roe solver.

The stability of this first-order algorithm can be improved by introducing "transverse propagation" of these waves, so that they affect the cells above or below row j as well, based on the propagation speeds in the y-direction. For a general system of m equations, each wave should be split into m pieces propagating at different speeds in the y-direction, based on the solution to a Riemann problem for the equation

$$u_t + g(u)_y = 0. \tag{4}$$

The approach proposed in [31] is used, in which each wave is used to define states U^L and U^R just to the left and right of the wave. If we are working with the pth wave, then

$$U^L = U_{i-1,j}^n + \sum_{q=1}^{p-1} \alpha_q r_q, \quad U^R = U^L + \alpha_p r_p.$$

The data U^L and U^R is then used in the Riemann problem

for (4) (actually the Roe approximation to this problem), which then yields a splitting of $U^R - U^L = \alpha_p r_p$ into eigenvectors of the Roe matrix $B = B(U^L, U^R)$:

$$\alpha_p r_p = \sum_{q=1}^{m} \beta_q w_q \quad \text{where } Bw_q = \mu_q w_q. \tag{5}$$

Each piece of the wave then propagates in the y-direction with speed given by the corresponding eigenvalue μ_q. The wave shown in Fig. 1a is thus split into m maves and a typical one is shown in Fig. 1b for the case $\mu_q > 0$. This wave updates the cell averages in cells (i, j) and $(i, j + 1)$, based on the area that the wave overlaps each cell.

The algorithm is implemented by first initializing each cell average to its value at the previous time step,

$$U_{ij}^{n+1} := U_{ij}^{n} \quad \text{for all } i, j$$

and then updating the average based on each wave. The waves $\mathcal{W}_{pq}$ shown in Fig. 1b, for example, would give the updates

$$U_{ij}^{n+1} := U_{ij}^{n+1} - \left(\frac{k\lambda_p}{h}\right)\left(1 - \frac{1}{2}\frac{k\mu_q}{h}\right)\beta_q w_q,$$

$$U_{i,j+1}^{n+1} := U_{i,j+1}^{n+1} - \frac{1}{2}\left(\frac{k^2\lambda_p\mu_q}{h^2}\right)\beta_q w_q.$$

The factors multiplying $\beta_q w_q$ are simply the fractions of each cell that is overlapped by the wave.

Note that with this tangential splitting the method remains conservative because the total contribution of the subwaves satisfies (5) and the area swept out by each subwave is the same as the area of the original wave.

This method is rather expensive for general use since at each interface $m + 1$ Riemann problems must be solved—one normal to the interface to obtain the decomposition (2) followed by a Riemann problem in the orthogonal direction for each of the m waves to obtain the decompositions (5). This results in m^2 waves, each of which must be propagated over the grid. Actually, for the Euler equations in two dimensions, where $m = 4$, we can reduce this to three waves in each direction since two waves always move at the same speed and can be combined.

A more substantial savings can be made by splitting the rightward and leftward going flux differences into upward and downward moving pieces rather than splitting each wave separately. This modified method, described in [35, 36], appears to work just as well, in general. However, for our present purpose of extending the method to irregular cells near the front, the original approach of splitting each wave appears preferable. Since the irregular cells cut by the front are relatively few in number, efficiency is not a great issue. Away from the front either approach can be used, or any other multidimensional finite volume method for that matter.

The method just described is still only first-order accurate, although the inclusion of transverse propagation gives better stability properties than the method in which waves are only propagated normal to interfaces. The method is typically stable as long as the time step k satisfies.

$$\frac{k}{h}\max_{p,q}(|\lambda_p|, |\mu_q|) \le 1. \tag{6}$$

For a scalar problem this same approach has been suggested in many different forms (e.g., the corner transport upwind scheme described by Colella [12].) Although the method is not second order, it turns out that the terms included by this modification give the correct cross-derivative terms needed for second-order accuracy. For example, for the linear system of equations $u_t + Au_x + Bu_y = 0$, a Taylor series expansion of the truncation error shows that for second-order accuracy we need to include an approximation to the term

$$\tfrac{1}{2}k^2 u_{tt} = \tfrac{1}{2}k^2(A^2u_{xx} + ABu_{xy} + BAu_{yx} + B^2u_{yy}).$$

The wave-splitting method described above can be shown to give a consistent approximation to the $\frac{1}{2}k^2(ABu_{xy} + BAu_{yx})$ term [35]. To achieve full second-order accuracy we need only add in the A^2u_{xx} and B^2u_{yy} terms and adjust the one-sided first-order approximations to Au_x and Bu_y to the second-order accurate centered approximations. These corrections are accomplished by the updates

$$U_{ij}^{n+1} := U_{ij}^{n+1} + \left(\frac{k\lambda_p}{h}\right)\left(1 - \frac{1}{2}\frac{k\lambda_p}{h}\right)\alpha_p r_p,$$

$$U_{i-1,j}^{n+1} := U_{i-1,j}^{n+1} - \left(\frac{k\lambda_p}{h}\right)\left(1 - \frac{1}{2}\frac{k\lambda_p}{h}\right)\alpha_p r_p.$$

In practice the strength of each wave is limited using a "slope-limiter," so each α_p in the above corrections is replaced by a limited value $\tilde{\alpha}_p$ obtained by comparing α_p with the corresponding α_p from the neighboring Riemann problem to the left (if $\lambda_p > 0$) or to the right (if $\lambda_p < 0$). More details can be found in [37, 31, 35], but they will not be repeated here since we currently do not implement these correction terms in the irregular cells at the front. There is some discussion in [31] of how such corrections might be applied on irregular cells, and the algorithm would perhaps be improved by implementing this. However, since there is expected to be a large jump at the tracked interface, this is precisely where the limiters are expected to minimize the effect of these "second-order"

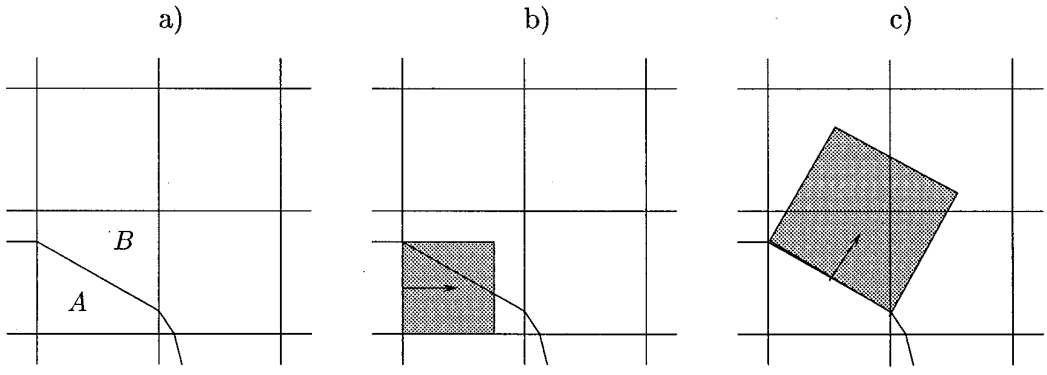

FIG. 2. Wave propagation in irregular cells created by the tracked front: (a) The grid. (b) A wave propagating from the interface between cell A and the cell to its left updates the solution in both cells A and B. (c) A wave propagating from the interface between cells A and B updates four cell averages.

corrections and to reduce the method to Godunov. It seems reasonable to drop these corrections altogether with considerable simplification of the algorithm.

3. FINITE VOLUME METHODS WITH A MOVING FRONT

As in our one-dimensional algorithm [40], our grid consists of two parts. We choose a uniform underlying grid that remains fixed for all time, and we also introduce tracked interfaces which vary from step to step for the discontinuities in the flow field. These tracked interfaces subdivide some regular cells into two or more subcells, creating some irregular cells. We use a piecewise linear representation of the front, so that each irregular cell is polygonal as shown in Fig. 2a. We then view the union of the regular cells and irregular cells as our global grid. In each grid cell, the cell average is defined by integrating the solution over the cell and dividing by the area of the cell. In Fig. 2a the numerical approximation to the cell average over the cell marked A is denoted by

$$U_A^n \approx \frac{1}{\mathcal{M}(A)} \int_A u(x, y, t_n)\, dx\, dy$$

where $\mathcal{M}(A)$ is the measure (area) of this cell. For a *fixed* grid of this form, the wave-propagation method described in the previous section is easily extended to work in these irregular cells. For example, Fig. 2b shows a typical p-wave arising from the Riemann problem in x between cell A and the cell to its left. This wave overlaps part of cell A and part of cell B. These cell averages are updated by

$$U_S^{n+1} := U_S^{n+1} - \frac{\mathcal{M}(\mathcal{W}_p \cap S)}{\mathcal{M}(S)}\, \alpha_p r_p \qquad (7)$$

for $S = A, B$. Here $\mathcal{W}_p$ represents the region swept out by the wave and $\alpha_p r_p$ is the jump across the wave.

At the interface between cells A and B, a Riemann problem is solved in the direction normal to the interface with data U_A^n and U_B^n. Note that the Euler equations are isotropic and easily rotated to this frame. The wave indicated in Fig. 2c overlaps four cells and modifies the cell area in each of these cells by the jump across this wave, weighted by the fraction of the cell area covered by the wave.

Again transverse propagation can be included by using these waves to define Riemann problems in the orthogonal direction. This will give a splitting of the wave $\mathcal{W}_p$ into subwaves $\mathcal{W}_{pq}$ for $q = 1, 2, ..., m$ moving oblique to the interface rather than normal to it. Each of these waves is then propagated in an analogous manner, computing its intersection with the neighboring cells. The method remains fully conservative. These methods are described in more detail in [31, 54].

A crucial step in computing the update (7) is to determine the area of intersection of the wave with each grid cell. This is accomplished by representing both the wave and each grid cell as a polygon with the vertices stored as a linked list. A standard algorithm for computing the intersection of two polygons is then used [47, 50].

Because waves are allowed to propagate through more than one grid cell and are not confined to remain within neighboring cells, this method satisfies the CFL condition and is stable with a timestep chosen relative to the uniform grid, even when there are very small irregular cells near the interface. The method with transverse propagation typically remains stable as long as k is chosen so that (6) is satisfied relative to the uniform grid size h.

Extending this method to a moving grid requires more care. In each time step our front tracking algorithm consists of the following steps:

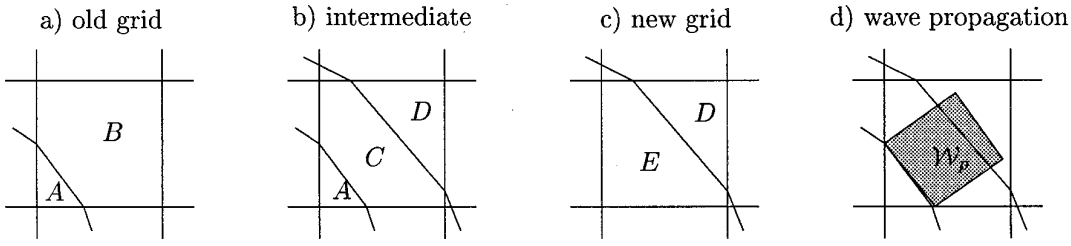

FIG. 3. (a) shows the old grid at time t_n and (c) shows the new grid at time t_{n+1}. The intermediate grid shown in (b) can be used to propagate the waves as shown in (d).

(1) Determine the new location of the tracked interfaces at the next time step.

(2) Insert these new tracked interfaces into the grid. Some cells will be subdivided and the values in each subcell must be initialized.

(3) Take a time step on this nonuniform grid using the finite volume method described above to update the cell averages.

(4) Delete the old tracked interfaces from the previous time step. Some subcells will be combined, and a value in the combined cell must be determined from the subcell values.

In this section we discuss steps 2, 3, and 4. The determination of the new front is discussed in the next section. For now we assume that we know the location of the front both at the beginning of the time step (the "old grid" at time t_n) and at the end (the "new grid" at time t_{n+1}). An example is shown in Fig. 3, where a single grid cell is split into two subcells A and B on the old grid and into two different pieces D and E on the new grid (one of these may be empty if the front moves into or out of the cell in this step).

One way to implement the wave propagation algorithm is to apply it on the "intermediate grid" shown in Fig. 3b, which contains both the old and new subdivisions and so the cell is split inot three pieces, A, C, and D. We will first describe this version since it is easiest to understand (and is a direct generalization of the method used in one dimesion [40]), but it is not used in practice because of difficulties when the old and new interfaces cross, as indicated below in Fig. 4. Instead a different approach is used in practice that is described below.

On the intermediate grid of Fig. 3b, the values U_S^{n+1} for $S = A, B, C$ must first be initialized to their values at time t_n. However, at time t_n we only know the values U_A^n and U_B^n. Since cells C and D are created by subdividing cell B, we initialize

$$U_C^{n+1} := U_B^n, \quad U_D^{n+1} := U_B^n. \tag{8}$$

(One could consider a more accurate initialization, based on higher order interpolation, but there are problems with this which already appear in one dimension, as discussed in [40].)

Figure 3d shows a typical wave $\mathcal{W}_p$ propagating from the interface between cells A and C into cells C and D. If the jump across this wave is $\alpha_p r_p$ then the updated formulas are given as usual by (7) for $S = C, D$ and, more generally, for any cell that the wave overlaps.

In a similar manner we update cell averages due to all waves from all the Riemann problems at each cell interface on the intermediate grid. Note that there are no waves generated from the interface between cells C and D since there is no jump across this interface due to the initialization (8).

To delete the old front, we merge cells A and C into a single cell E on the new grid. The value U_E^{n+1} is determined by the appropriate weighted average of the cell averages in cells A and C:

$$U_E^{n+1} = [\mathcal{M}(A)U_A^{n+1} + \mathcal{M}(C)U_C^{n+1}]/\mathcal{M}(A \cup C).$$

In practice this approach has difficulties if the old and new fronts are not well separated, as illustrated in Fig. 4. Here the old and new fronts cross within the cell. In principle the approach outlined above could still be applied, but now the intermediate grid would have this cell subdivided into four pieces with some complication of the algorithm. To avoid this difficulty we use a more robust algorithm in which the wave propagation is applied on the new grid rather than the intermediate grid. The cell averages U_E^{n+1} and U_D^{n+1} are first initialized by weighted averages of the values U_A^n and U_B^n:

$$U_S^{n+1} := [\mathcal{M}(A \cap S)U_A^n + \mathcal{M}(B \cap S)U_B^n]/\mathcal{M}(S) \quad \text{for } S = D, E$$

and are then updated by all waves propagating across the grid, as before. The waves are still computed, based on the old grid, solving a Riemann problem at each of the old interfaces. But then the wave is applied on the new grid. Figure 4d illustrates a typical wave obtained by solving

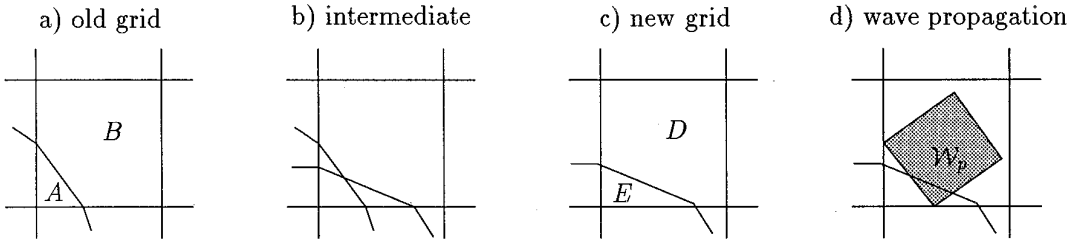

FIG. 4. In this case the intermediate grid in (b) has four subcells since the old and new fronts cross. Instead of using the intermediate grid, it is possible to propagate the waves defined by the old grid directly onto the new grid as shown in (d).

the Riemann problem with data U_A^n and U_B^n. This wave updates the cell averages U_E^{n+1} and U_D^{n+1} by the usual formula (7), now for $S = D, E$.

This method is easy to implement. For the data structures we keep the old grid and new grid separate. The values U^n on the old grid are used to solve Riemann problems and the values U^{n+1} on the new grid are updated by the waves. Step 4 in the algorithm is essentially eliminated since the new grid structure contains no reference to the old grid. The method is still fully conservative and gives exactly the same cell values as the first approach with an intermediate grid.

Note that each wave is propagated independently of all others. In fact for a nonlinear problem there should be interaction of the waves when they collide. The algorithm we used can be viewed as a linearization of these wave interactions, since for a linear problem (if one space dimension, at least) this treatment would be correct. This is discussed in more detail in [31, 32].

4. FRONT MOVING ALGORITHM

In each time step the front is approximated by a piecewise linear *curve* with knots at the points where this curve intersects the grid lines. We call the line segment dividing a given cell into two subcells an *interface,* and with each *interface* we have associated two *points,* the endpoints of the segment. The data structure is illustrated in Fig. 5. The *points* defining the *interface* in the lth subdivided cell are denoted by (x_l^1, y_l^1) and (x_l^2, y_l^2), $l = 1, 2, ..., n$. In general $x_l^1 = x_{l-1}^2$ and $y_l^1 = y_{l-1}^2$ as shown in Fig. 6. (We will focus on explaining the main ideas and ignore the boundary conditions.)

In each time step we first solve a one-dimensional Riemann problem in a direction normal to each tracked interface using the values from the adjacent cells as data and obtain the resulting jumps $\alpha_p r_p$ and speeds λ_p. We expect the solution to this Riemann problem to consist of one strong wave, corresponding to the shock or interface being tracked, and other weaker waves. The strong wave is used to help choose the new interface location. We discuss one simple approach in detail. See Section 6 and [1, 9, 10] for other ways to advance the front.

Let (x_l^*, y_{l-1}^*), $* = 1, 2$, be *points* of the *interface l.* Assume that the strong wave is in the pth wave family, and let λ_p be the speed of the strong wave. Then the new location $(\widehat{x_l^*}, \widehat{y_l^*})$ of the *point* (x_l^*, y_l^*), under the current time step k, can be calculated by simply using the formula

$$\begin{bmatrix} \widehat{x_l^*} \\ \widehat{y_l^*} \end{bmatrix} = \begin{bmatrix} x_l^* \\ y_l^* \end{bmatrix} + k\lambda_p \begin{bmatrix} \cos(\theta_l) \\ \sin(\theta_l) \end{bmatrix}, \tag{9}$$

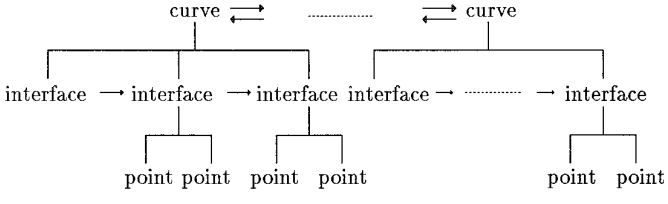

FIG. 5. Data structure of the tracked interfaces in our front tracking code.

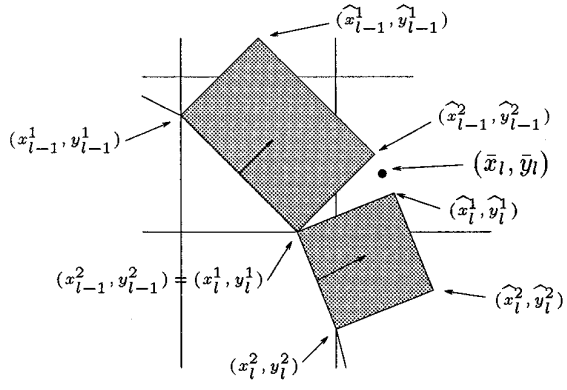

FIG. 6. Strong waves propagating from the old interface are used to determine the new interface location. The point $(\bar{x}_l, \bar{y}_l)$ is halfway between the corners of the waves, and a piecewise linear curve through these points is intersected with the grid lines to determine the new front location.

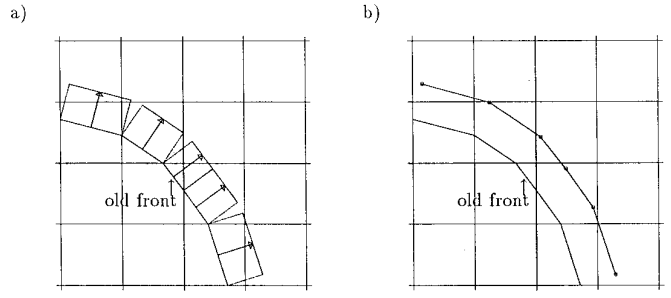

FIG. 7. Front propagation: (a) Tracked interfaces after propagating the original interfaces using the strong wave speeds obtained from the normal Riemann problems during the current time step. (b) Piecewise linear interpolant through the points $(\bar{x}_l, \bar{y}_l)$ denoted by dots. The final *interfaces* and *points* on the new grid are determined by where this curve intersects the grid lines.

where $(\cos(\theta_l), \sin(\theta_l))^T$ is the normal direction to the *interface* l at an angle θ_l to the x-axis, and $* = 1, 2$.

Note that in many problems, e.g., when there is strong shear layer flow along the discontinuities, the tracked interfaces should be advanced not only in the normal direction to the interface as illustrated in Fig. 7a, but also in the tangential direction. This can be done quite easily by introducing a transverse velocity μ, for example, the average transverse velocity from the states on either side of this wave. Then (9) is replaced by

$$\begin{bmatrix} \widehat{x_l^*} \\ \widehat{y_l^*} \end{bmatrix} = \begin{bmatrix} x_l^* \\ y_l^* \end{bmatrix} + k\lambda_p \begin{bmatrix} \cos(\theta_l) \\ \sin(\theta_l) \end{bmatrix} + k\mu \begin{bmatrix} \sin(\theta_l) \\ -\cos(\theta_l) \end{bmatrix}. \quad (10)$$

After moving the points according to (9) or (10), the points $(\widehat{x_l^*}, \widehat{y_l^*})$ will no longer be proper *points* to define a set of *interfaces* (see Fig. 6). The goal is now to use these points to define a new *curve* that passes close to these points. This can be done in many ways. One simple approach is to define average points

$$\bar{x}_l = \tfrac{1}{2}(\hat{x}_{l-1}^2 + \hat{x}_l^1), \quad \bar{y}_l = \tfrac{1}{2}(\hat{y}_{l-1}^2 + \hat{y}_l^1) \quad (11)$$

as indicated in Fig. 6b and then to interpolate a curve through these points and finally determine where this curve intersects the gird lines in order to define the new *interfaces* and *points*. In the simplest case we could do a piecewise linear interpolation through the points $(\bar{x}_l, \bar{y}_l)$, as indicated in Fig. 7b. The averaging procedure (11) is performed (rather than interpolating through the $(\widehat{x_l^*}, \widehat{y_l^*})$ directly) to avoid difficulties when the neighboring waves overlap one another rather than spreading out as in Fig. 7.

The new *curve* does not exactly match the proper front location, of course, nor does it even match exactly with the waves used to define it. However, it lies in approximately the correct location so that when the finite volume method described in Section 3 is applied, the strong waves arising from each of the tracked *interfaces* will move approximately to the new front boundary, as indicated in Fig. 8. As a result, there is minimal smearing of the sharp discontinuity in the solution. Recall that smearing occurs when a large jump propagates only part way through a grid cell. Averaging it onto the grid gives a smeared intermediate value. With our choice of the new *curve*, the strong waves will overlap some cells nearly completely and others hardly at all as seen in Fig. 8. The ratio $\mathcal{M}(\mathcal{W}_p^l \cap S)/\mathcal{M}(S)$ in (7) will be close to zero or one for each cell S.

5. NUMERICAL RESULTS

We use the Euler equations of gas dynamics in two space dimensions as a model system:

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{pmatrix} = 0, \quad (12)$$

where ρ, u, v, p, E are the density, velocity in the x-direction, velocity in the y-direction, pressure, and total energy of gas per unit mass, respectively. We assume a γ-law gas, in which the internal energy satisfies $e = p/((\gamma - 1)\rho)$, where γ is the ratio of specific heats ($\gamma = 1.4$ is used here). Then the total energy of the gas per unit mass is $E = e + \frac{1}{2}(u^2 + v^2)$. The four components of Eq. (12) express the conservation of mass, momentum in the x-direction, momentum in the y-direction, and energy, respectively [14].

We show some numerical results for problems involving shock waves for this model system. As a first example, we consider a radially symmetric problem in which a circular

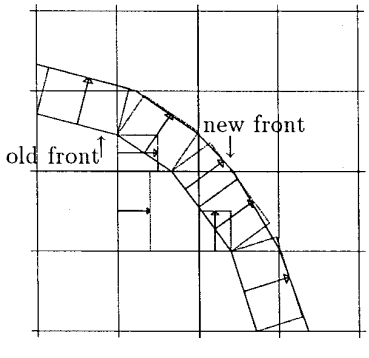

FIG. 8. Wave propagation in Step 3 (only some of the waves are drawn). Each wave is propagated independently as described in Section 3. Note that the tracked waves are propagated close to the tracked interface introduced in Step 2.

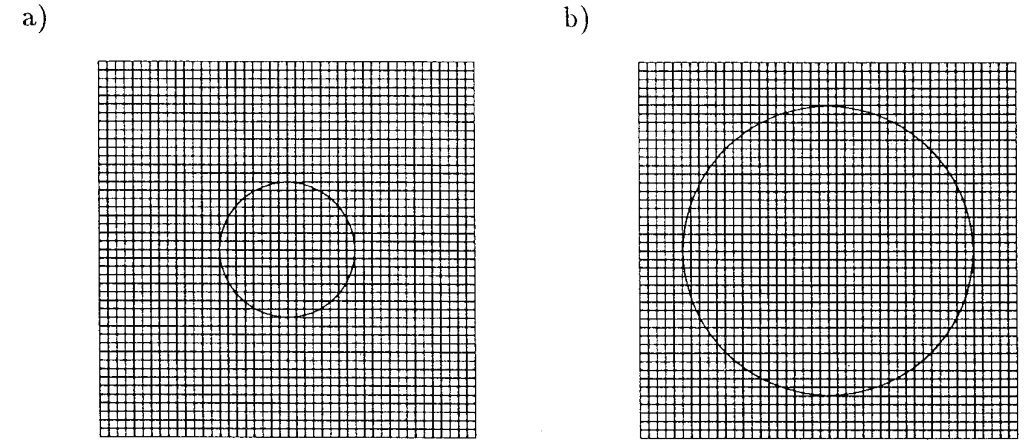

FIG. 9. Grid used for the radially symmetric expanding shock wave. (a) Grid for the initial data. (b) Final grid after 24 time steps (time $t = 0.1$).

expanding shock wave is tracked. Outside of a circle of radius $r_0 = 0.2$, we set

$$\rho = 1.4, \quad u = 0, \quad v = 0, \quad p = 1.$$

Inside the circle, the initial data is

$$\begin{aligned} \rho(x, y, 0) &= 5.143204 \\ u(x, y, 0) &= 2.045108\,(x - x_0)r/r_0^2 \\ v(x, y, 0) &= 2.045108\,(y - y_0)r/r_0^2 \\ p(x, y, 0) &= 9.045462, \end{aligned}$$

where $r^2 = (x - x_0)^2 + (y - y_0)^2$ is the distance from the center $(x_0, y_0) = (0.5, 0.5)$. The initial grid is shown in Fig. 9a, where the initial shock is inserted as an interface that subdivides some cells in the underlying 40×40 grid.

After 24 time steps (time $t = 0.1$ and Courant number $\nu_0 = 0.5$), we obtain the results shown in Fig. 10 on the grid shown in Fig. 9b. Notice that the tracked shock remains smooth and circular and appears to be very well located. The cross section along $y = 0.5$ shown in Fig. 10b shows the sharpness of our result clearly. The solid line in this figure is the ''true'' solution as calculated with our one-dimensional front tracking algorithm [40] on the system $u_t + f(u)_r = \psi(u)$ with appropriate source terms for the radial symmetry, using $h = 0.001$. The two-dimensional results shown above were obtained using the high resolution method of [31] on the regular cells, with second-order corrections and slope limiters. On the irregular cells the Godunov method with tangential splitting is used, as described in Section 3. Second-order slope corrections with limiters were not applied in the irregular cells, where only the first-order propagation (with transverse motion) was used.

For this problem we can compute the error in each grid cell using the one-dimensional results for comparison. We can also investigate the error in the location of the tracked front relative to the true shock location, as is done in Fig. 11a. This was obtained by writing each *point* (x_l^1, y_l^1) along the *curve* in polar coordinates and then plotting r as a function of θ. For this circular front, r should be constant. For small h the graph is fairly flat, showing that the *curve* is nearly circular, and pointwise convergence to the correct radius is clearly observed. If we measure the error in the location of the front by averaging the function $r(\theta)$ over θ and comparing to the correct radius, we compute an order of accuracy of about 1.77 (see Table I).

Figure 11b shows the value of ρ in each irregular cell that is behind the shock (towards the origin) again plotted against θ (measured from the area-weighted center of the cell). Figure 11c shows the same situation for the irregular cells ahead of the shock. In both cases convergence towards the true solution is observed. The large spikes in these figures typically correspond to small irregular cells. As explained in Section 3, there is some unavoidable smearing of the discontinuity with our simple algorithms for moving the front and propagating waves. A tiny cell may be missed entirely by the waves or it may be entirely covered by a small piece of a wave coming from the wrong side of the front. Computing a 1-norm of the error along the front, by weighting the error in cell C_1 by $\mathscr{M}(C_l)$ and summing, shows reasonable convergence on both sides of the front. We define

$$\|E_{\chi^\pm}^n\|_1 = \frac{1}{h} \sum_{l \in \chi^\pm} \mathscr{M}(C_l)\,|\,U_l^n - u_l^n\,|,$$

where the sum is over the set of irregular cells in either the state behind the shock χ^-, or the state ahead the shock

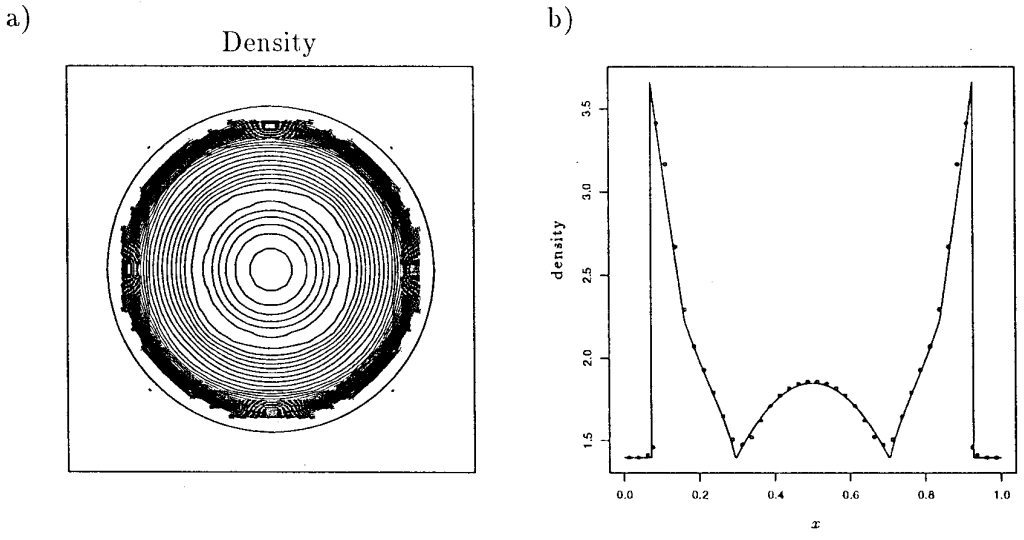

FIG. 10. Results for the radially symmetric expanding shock wave. (a) Density contours at time $t = 0.1$. (b) Cross section of density along line $y = 0.5$. The solid line is the "true" solution obtained from solving the system $u_t + f(u)_r = \psi(u)$ with appropriate source terms for the radial symmetry using the one-dimensional front-tracking algorithm. The dotted points are the two-dimensional result.

χ^+. The factor $1/h$ is to account for the fact that we are integrating only along a one-dimensional curve in the plane. Table I shows the values of these errors for this grid refinement study. Behind the shock we observe a convergence rate of 1.3. Ahead of the shock it is only 0.82, but note that the magnitude of the error is smaller on this side. The 1-norm over the entire domain is also shown, summing the error in each cell weighted by cell area. These errors are naturally much smaller since the largest errors are concentrated near the front. We see overall convergence in the 1-norm with rate 1.67 for this example. Even in the max-norm (the largest error over any gird cell, including the irregular cells) we observe convergence at rate 1.43. (See [54] for for more accuracy results.)

Note in Table I the number in the parenthesis represents the exponent, for example, 9.1615(−2) means 9.1615×10^{-2}. The order of accuracy of a method is computed by performing a linear least-squares fit to a sequence of mesh refinement data $\{(\log h_l, \log\|E^n\|), l = 1, ..., m\}$ and taking the slope as the order of accuracy of the method.

Our next example of shock tracking concerns a shock reflection problem in which an oblique shock is reflecting off a ramp. This problem has been studied extensively and the solution structure is well documented in many instances; see [18] for an example. See also [4, 13] for numerical results on these ramp-reflection problems using high resolution shock-capturing methods.

As in the computations done by Chern and Colella [9], we initialize the incident shock wave at the ramp corner with an angle normal to the ramp where the preshock state is on the left to the shock. (The shock is moving leftward). For convenience, the ramp is arranged so that it is aligned with the grid in front of the shock and cuts through the underlying Cartesian grid in back of the shock as seen in Figs. 12 and 13.

Figure 12 shows results for a single Mach reflection off a 30° ramp. The Mach number of the incident shock is 1.65. In Fig. 12a, we show the tracked shock at 20 equally spaced times. From the figure, it is easy to observe that the shock–ramp interaction causes a kink to form, which corresponds to the location of a triple point. Our shock-tracking algorithm does quite well in tracking both the incident shock (above the kink) and the Mach stem (below the kink). We have not done anything special to follow the kink. The reflected shock behind the triple point is captured.

In Figs. 12b and c, we show the density contours and the cross section along the ramp, respectively, for the same run at time $t = 0.64$. We employed the same high resolution shock-tracking method as to the previous cylindrical shock problem with a 200×80 grid on a rectangular region ($[0, 2] \times [0, 0.8]$), and Courant number $\nu_0 = 0.9$ (relative to the uniform cells). On the ramp boundary, the proper boundary condition for the Euler equations is zero normal velocity. Here we used a fictitious cell approach in the part of the ramp that aligns with the underlying grid boundary, and a wave reflection approach in the part that cuts through the grid [31]. At the other boundaries, nonreflecting outflow boundary conditions were applied [40]. The solid line in Fig. 12b shows the result obtained using shock-capturing on a uniform grid with twice the resolution.

Figure 13 shows two calculations for a stronger Mach 10 shock, in which case a double Mach reflection is observed. In the calculation on the left, the Roe solver is

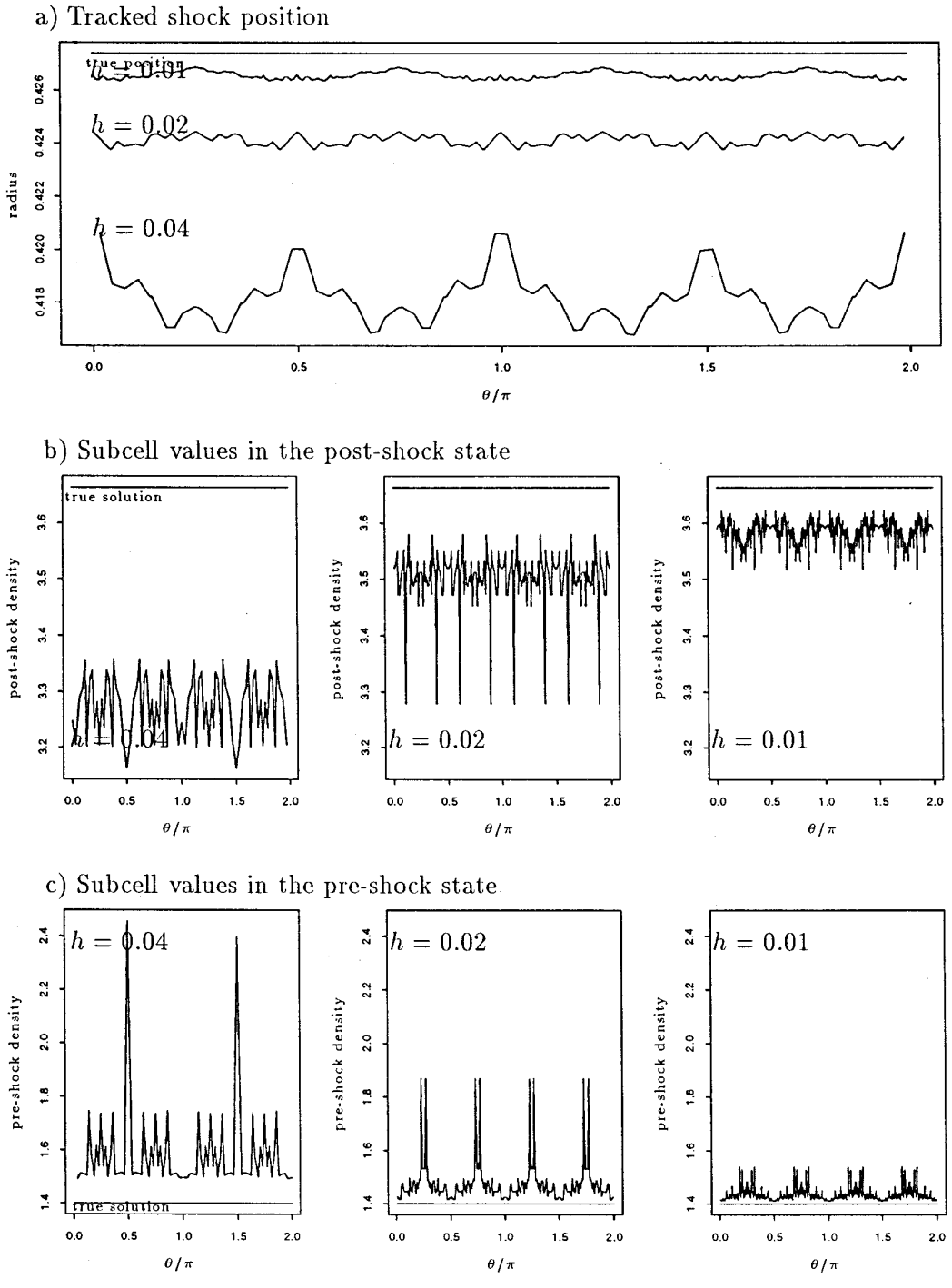

FIG. 11. A convergence study of the tracked shock position and density in the irregular cells for a radially symmetric expanding shock wave. Each figure is plotted as a function of normalized distance around the circular front. The straight line shown in the figure is the "true" solution.

used to solve all Riemann problems, including those at the tracked front. In the calculation on the right, the exact Riemann solver is used. For most calculations the two approaches give essentially the same results and the Roe solver is preferred because it is quicker. For this strong shock reflection problem, however, we do see some slight difference. In particular, the Mach stem hits the ramp more orthogonally with the exact Riemann solver. With the Roe solver there is some pushing out of this shock, an effect that is also seen in other contexts with the Roe solver.

TABLE I

An Accuracy Study for a Radially Symmetric Problem

h	$\|E^n\|_1$	$\|E^n\|_\infty$	$\|E^n_{\chi^-}\|_1$	$\|E^n_{\chi^+}\|_1$	$\|E^n_{\text{front}}\|$
0.04	9.1615(−2)	1.0567(0)	7.9522(−1)	1.5437(−1)	9.2733(−3)
0.02	2.7025(−2)	4.6930(−1)	2.6783(−1)	8.7165(−2)	3.2840(−3)
0.01	9.0258(−3)	1.4570(−1)	1.3099(−1)	4.9681(−2)	8.0282(−4)
Order p	1.67	1.43	1.30	0.82	1.77

Note. Errors in density are shown, with notation defined in the text.

(See, for example, Quirk's description of the "carbuncle phenomenon" in [51].)

Again the solid lines in Fig. 13c show the results obtained using shock-capturing on a uniform grid with twice the resolution. The finer grid is better able to resolve the flow along the ramp behind the shock and does a fine job of capturing the shock, as well, for this problem (and also in the previous single Mach reflection example). Our present purpose, however, is to demonstrate that our shock-tracking approach can track the shock at the correct location, even in these more complicated problems involving triple points.

6. EXTENSIONS

Currently work is under way to apply this algorithm to specific applications, including fluid interface instabilities and porous media flow. In the latter case, multiphase flow problems require the solution of an elliptic equation for the pressure in each time step, with discontinuous coefficients across the moving interface between different fluids. We are now studying the use of *immersed interface methods* as described by LeVeque and Li [39] for this part of the problem, coupled with front-tracking as described in this paper.

The techniques developed here could be applied on arbitrary nonuniform grids, for example, on a quadrilateral body-fitted grid to study shocked flow in an interesting geometry. We prefer the simplicity of Cartesian grids, however, and plan to ultimately combine the front-tracking procedure with a Cartesian grid treatment of irregular geometries, as developed by Berger and LeVeque [2, 3, 30]. This requires some additional work in cells where a front intersects the boundary, since an irregular cell cut off by the boundary is then also subdivided by the front. In principle this should not be difficult to handle.

There are some problems where a more careful job must be done in tracking the front. In particular, the present approach may be insufficient in some problems where there are different fluids or very different equations of state on the two sides of the front. In this case our basic philosophy that the computational front subdividing grid cells is only an approximation to the true front location, and our willingness to accept some smearing of the solution near this front in return for a simple algorithm, would probably be inadequate.

There are two related problems that arise with the current simplified approach in cases where it is crucial to maintain the integrity of the fluid on the two sides of the front:

1. Our algorithm for moving the front makes no explicit attempt to maintain conservation of each fluid separately, in the sense of properly conserving the area on each side of the interface.

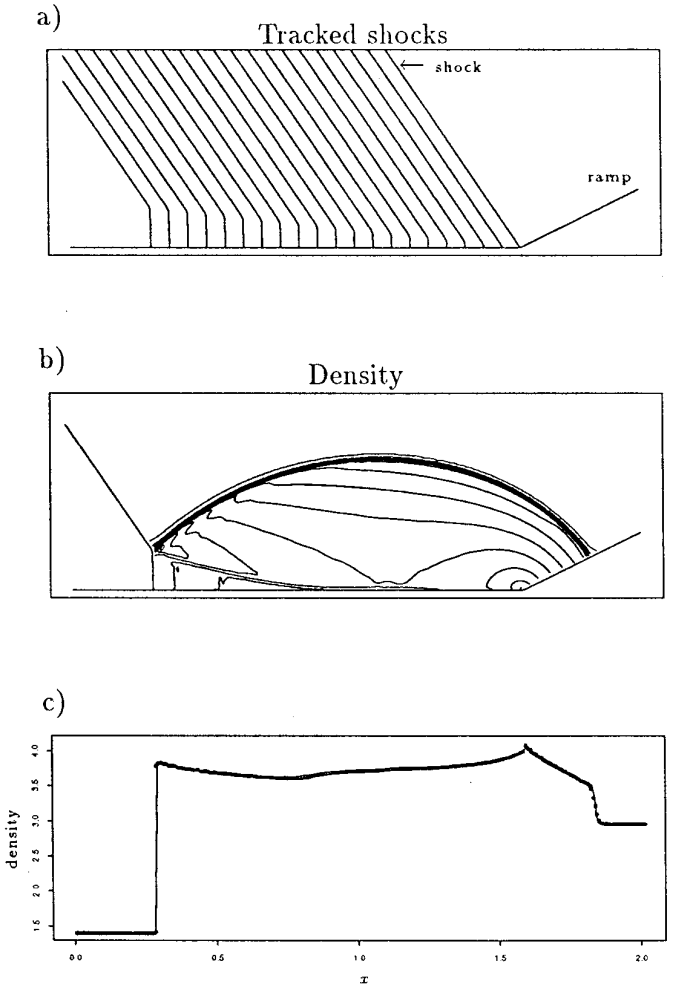

FIG. 12. Results for the single Mach reflection case, a Mach 1.65 incident shock and a 30° ramp: (a) The tracked shocks at 20 equally spaced times. (b) Contours of density at time $t = 0.64$. (c) Cross section of density along the ramp at this time. The solid line is from a shock-capturing calculation on a finer grid.

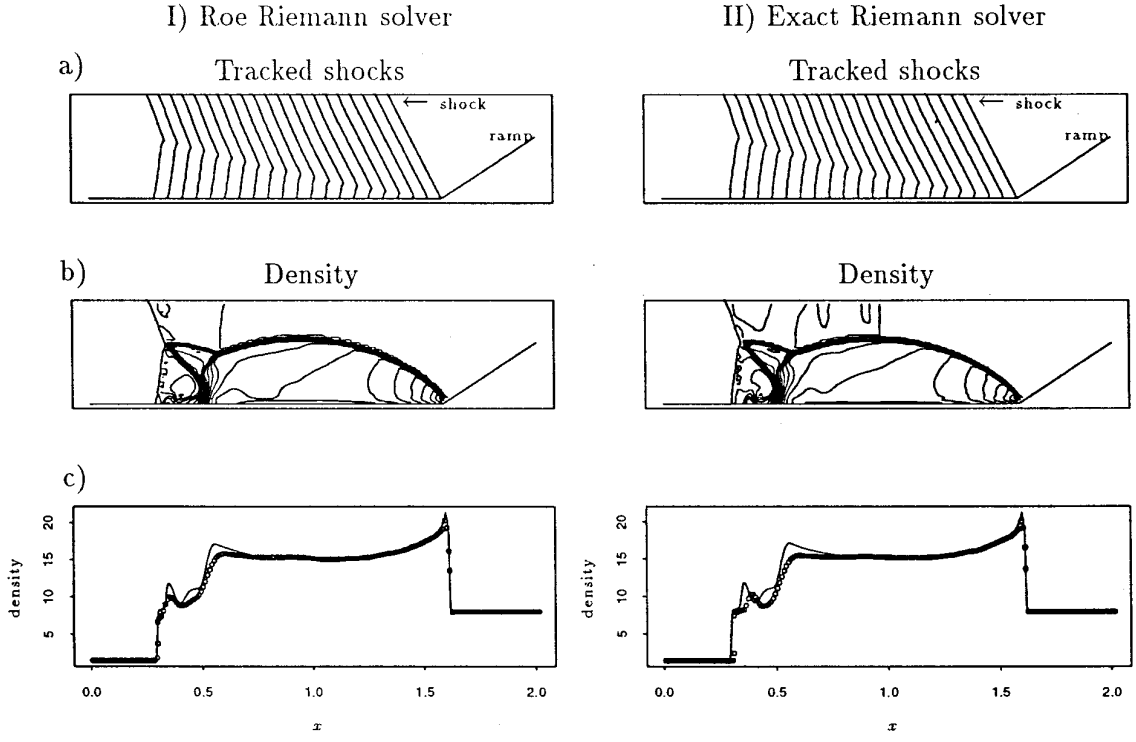

FIG. 13. Results for the double Mach reflection case, a Mach 10 incident shock, and a 30° ramp: (I) Using the Roe Riemann solver. (II) Using the exact Riemann solver, which gives better results along the ramp for this strong shock case. In each case, we show (a) the tracked shocks at 20 equally spaced times, (b) contours of density at time $t = 0.1$, and (c) cross section of density along the ramp at this time. The solid line is from a shock-capturing calculation on a finer grid.

2. In the wave propagation algorithm used to advance the solution on the new grid, waves are allowed to cross the front, which causes smearing near the front and a mixing of the fluids.

These problems could presumably be remedied at the expense of some further complication of the algorithm. Research is underway to develop a more sophisticated version of our algorithm which attempts to track fronts more accurately and avoid this smearing, for use in applications where this is essential.

One possibility for solving Problem 1 is to use a volume-of-fluid (VOF) representation of the front, instead of the spline approach used here. One could use the strong waves defining the front motion to propagate a VOF marker on the "old grid" (as described in Section 3) and then use the updated marker concentrations to choose the new grid according to a conservative reconstruction algorithm such as those in [56] or [49].

We would still be faced with Problem 2, however, that waves arising in the wave propagation step of the algorithm might not respect the front location and may partially cross the front. Of particular concern are the strong waves that define the front. One possibility here would be to compute the intersection of each wave with the cells on the proper side of the new front and only update these cells. To maintain conservation, however, it would be necessary to increase the strength of the wave used in updating these cells. Let $\alpha_p r_p$ be the jump across one such wave, $\mathcal{M}(\mathcal{W}_p)$ the area of the wave that should be propagated, and $\mathcal{M}_{\text{in}}$ the total area of the intersections of this wave will cells on the proper side of the front, so $\mathcal{M}_{\text{in}} \leq \mathcal{M}(\mathcal{W}_p)$. Then replacing $\alpha_p r_p$ by $(\mathcal{M}(\mathcal{W}_p)/\mathcal{M}_{\text{in}})\ \alpha_p r_p$, before using it to update the cells by (7), will ensure that conservation is maintained. Since $\mathcal{M}_{\text{in}} \approx \mathcal{M}(\mathcal{W}_p)$, in general, we expect that this will cause no difficulties with stability.

7. CONCLUSIONS

We have described a simple approach to multidimensional front tracking that is based on the following ideas:

1. Use a uniform Cartesian grid with some cells divided into two or more subcells.

2. Choose the subdivisions based on information about propagation of strong waves from the solution of Riemann problems.

3. Apply a high-resolution finite volume method on the resulting grid.

Because high-resolution finite volume methods are used, it is not necessary that the grid capture the exact location of the discontinuity. The grid can be viewed as a refinement of a uniform grid chosen to conform reasonably well with the expected location of discontinuities. The method is then expected to work at least as well as a uniform grid method (hopefully much better) with relatively little additional work. Numerical experiments have confirmed that this method does in fact do a very good job of maintaining sharp discontinuities in the correct locations, at least for a certian class of problems. Ongoing work on various applications will help to delineate the set of problems for which this is adequate, while we also develop improvements more suitable for other classes of problems.

ACKNOWLEDGMENTS

This work was supported in part by NSF Grants DMS-8657319 and DMS-9204329 and was performed while the second author was a graduate student in the Applied Mathematics Department at the University of Washington and a postdoctoral research associate in the Department of Applied Mathematics and Statistics at the State University of New York, Stony Brook. Work at Stony Brook was supported (in part) by the Oak Ridge National Laboratory, Subcontract 19X-SJ067V and by NSF Grant DMS-9201581.

REFERENCES

1. J. B. Bell, P. Colella, and M. L. Welcome, UCRL-JC-105251, preprint, 1991; *AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 24–26.*
2. M. Berger and R. J. LeVeque, *Comput. Systems Eng.* **1,** 305 (1990).
3. M. Berger and R. J. LeVeque, in *AIAA Conference on Computational Fluid Dynamics, 1991,* CP-91-1602 (unpublished).
4. M. J. Berger and P. Colella, *J. Comput. Phys.* **82,** 64 (1989).
5. A. Bourlioux, Ph.D. thesis, Princeton University, June 1991 (unpublished).
6. J. U. Brackbill, D. B. Kothe, and C. Zemach, *J. Comput. Phys.* **100,** 335 (1992).
7. F. Bratvedt, K. Bratvedt, C. F. Buchholz, *et al.,* ''Three-Dimensional Reservoir Simulation Based on Front Tracking,'' in *North Sea Oil and Gas Reservoirs-III* (Kluwer Academic, New York Amsterdam, 1994), p. 247.
8. F. Bratvedt, K. Bratvedt, C. F. Buchholz, T. Gimse, H. Holden, L. Holden, and N. H. Risebro, *Survey Math. Industry* **3,** 185 (1993).
9. I.-L. Chern and P. Colella, UCRL-97200, Lawrence Livermore Laboratory, 1987 (unpublished).
10. I.-L. Chern, J. Glimm, O. A. McBryan, B. Plohr, and S. Yaniv, *J. Comput. Phys.* **63,** 83 (1986).
11. A. J. Chorin, *J. Comput. Phys.* **35,** 1 (1980).
12. P. Colella, *J. Comput. Phys.* **87,** 171 (1990).
13. P. Colella, *J. Comput. Phys.* **87,** 171 (1990).
14. R. Courant and K. O. Friedrichs, *Supersonic Flow and Shock waves* (McGraw–Hill, New York, 1954).
15. Y. Deng, J. Glimm, and D. H. Sharp, *Los Alamos Sci.* **21,** 124 (1993).
16. B. A. Finlayson, *Numerical Methods for Problems with Moving Fronts* (Ravenna Park, Seattle, 1992).
17. M. D. Giacinto and M. Valorani, *Comput. Fluids* **17,** 61 (1989).
18. H. M. Glaz, P. Colella, I. I. Glass, and R. L. Deschambault, *Proc. R. Soc. London A* **398,** 117 (1985).
19. J. Glimm, A Review of Interface Methods for Fluid Computations,'' in *Computational Methods for Subsurface Hydrology* (Springer-Verlag, New York, 1990), p. 421.
20. J. Glimm, J. Grove, B. Lindquist, O. A. McBryan, and G. Tryggvason, *SIAM J. Sci. Stat. Comput.* **9,** 61 (1988).
21. J. Glimm and O. A. McBryan, *Adv. in Appl. Math.* **6,** 422 (1985).
22. J. Grove, *J. Appl. Numer. Math.* **14,** 213 (1994).
23. J. Grove, R. Holmes, D. H. Sharp, Y. Yang, and Q. Zhang, *Phys. Rev. Lett.* **7121,** 3473 (1993).
24. J. Grove and R. Menikoff, *J. Fluid. Mech.* **219,** 313 (1990).
25. F. H. Harlow and J. E. Welch, *Phys. Fluids* **8,** 2182 (1965).
26. C. W. Hirt and B. D. Nichols, *J. Comput. Phys.* **39,** 201 (1981).
27. J. M. Hyman, *Physica D* **12,** 396 (1984).
28. B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, *J. Comput. Phys.* **113,** 134 (1994).
29. R. J. LeVeque, ''Shock-Tracking with the Large Time Step Method,'' in *Proccedings 7th Int. Conf. Comput. Methods in Appl. Sci. Eng., Versailles, 1985* edited by R. Glowinski and J.-L. Lions.
30. R. J. LeVeque, ''Cartesian Grid Methods for Flow in Irregular Regions, in *Num. Meth. Fl. Dyn. III,* edited by K. W. Morton and M. J. Baines (Clarendon Press, Oxford, 1988), p. 375.
31. R. J. LeVeque *J. Comput. Phys.* **78,** 36 (1988).
32. R. J. LeVeque, *SIAM J. Numer. Anal.* **25,** 1 (1988).
33. R. J. LeVeque, *Hyperbolic Conservation Laws and Numerical Methods,* Von Karman Institute for Fluid Dynamics Lecture Series, 90-03, 1990.
34. R. J. LeVeque, *Numerical Methods for Conservation Laws* (Birkhäuser-Verlag, Basel, 1990).
35. R. J. LeVeque, ''Simplified Multi-dimensional Flux Limiter Methods,'' in *Numerical Methods for Fluid Dynamics* 4, edited by M. J. Baines and K. W. Morton (Oxford Univ. Press, Oxford, 1993), p. 175.
36. R. J. LeVeque, CLAWPACK—A Software Package for Solving Multi-Dimensional Conservation Laws,'' in *Proceedings 5th Int. Conf. Hyperbolic Problems, 1994.*
37. R. J. LeVeque SIAM J. Numer. Anal. (1996); UW Applied Math Tech Report 93-03.
38. R. J. LeVeque and Z. Li, *SIAM J. Sci. Comput.* (1997), to appear.
39. R. J. LeVeque and Z. Li, *SIAM J. Numer. Anal.* **31,** 1019 (1994).
40. R. J. LeVeque and K.-M. Shyue, *SIAM J. Sci. Comput.* **16,** 348 (1995).
41. D.-K. Mao, *J. Comput. Phys.* **92,** 422 (1991).
42. D.-K. Mao *J. Comput. Phys.* **104,** 377 (1993).
43. G. Moretti, *Comput. Fluids* **15,** 59 (1987).
44. W. Mulder, S. Osher, and J. A. Sethian, *J. Comput. Phys.* **100,** 209 (1992).
45. W. F. Noh and P. Woodward, *SLIC* (*Simple Line Interface Calculation*), Lecture Notes in Physics (Springer Verlag, New York/Berlin, 1976).
46. E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flow* (Elsevier, New York, 1987).
47. J. O'Rourke, *Computational Geometry in C* (Cambridge Univ. Press, New York, 1994).
48. S. Osher and J. A. Sethian, *J. Comput. Phys.* **79,** 12 (1988).
49. J. E. Pilliod Jr. and E. G. Puckett, preprint, 1993.

50. F. P. Preparata and M. I. Shamos, *Computational Geometry: an Introduction* (Springer-Verlag, New York, 1985; corrected and expanded second printing, 1988).
51. J. J. Quirk, ICASE Report No. 92-64, NASA Langley Research Center, 1992 (unpublished).
52. P. L. Roe, *J. Comput. Phys.* **43,** 357 (1981).
53. M. D. Salas, *AIAA J.* **14,** 583 (1976).
54. K.-M. Shyue, Ph.D. thesis, University of Washington, 1993.
55. K.-M. Shyue, *Proc. Symp. Appl. Math.* **48,** 383–387 (1994).
56. B. Swartz, *Math. Comput.* **52,** 675 (1989).
57. C. Tu and C. S. Peskin, *SIAM J. Sci. Stat. Comput.* **13,** 1361 (1992).
58. S. O. Unverdi and G. Tryggvason, *J. Comput. Phys.* **100,** 25 (1992).