SIAM J. NUMER. ANAL.
Vol. 41, No. 5, pp. 1926–1947

CONSERVATIVE FRONT TRACKING WITH IMPROVED ACCURACY*

JAMES GLIMM[†][‡], XIAOLIN LI[†], YINGJIE LIU[†], ZHILIANG XU[†], AND NING ZHAO[§]

Abstract. We propose a fully conservative front tracking algorithm for systems of nonlinear conservation laws. The algorithm improves by one order in its convergence rate over most finite difference schemes. Near tracked discontinuities in the solution, the proposed algorithm has $\mathcal{O}(\Delta x)$ errors, improving over $\mathcal{O}(1)$ errors commonly found near a discontinuity. Numerical experiments which confirm these assertions are presented.

Key words. front tracking, conservation, contact discontinuity

AMS subject classifications. 35L65, 74S10

DOI. 10.1137/S0036142901388627

1. Introduction.

We propose and demonstrate a tracking finite difference algorithm for the problem of nonlinear conservation laws which is (a) fully conservative and (b) improves the local error by one power of Δx near tracked discontinuities. The one dimensional (1D) version of these ideas was presented in [9], and a preliminary (but different) two dimensional (2D) algorithm with the same properties was given in [8], while the results were announced in [10].

Discontinuities in the solution of systems of nonlinear hyperbolic conservation laws are a primary difficulty for numerical simulation. These equations have both linear and nonlinear discontinuities, and (perhaps counterintuitively) the former are more difficult. Nonlinear discontinuities are self-focusing, and their numerical solution does not grow in width with time. The linear discontinuities in contrast do grow and may typically occupy 4 to 10 mesh cells in width.

Front tracking was introduced to give special treatment to discontinuities. A robust validated code has been developed and used in production simulation of fluid instabilities [5, 7, 6, 4]. See also the URL http://www.ams.sunysb.edu/~shock/FTdoc/FTmain.html.

In this paper, we address an algorithmic issue—formulation of a conservative tracking algorithm. In its original formulation, conservation was enforced only in regular grid cells, those not cut by the tracked front. The missing points of the computational stencil, in the case of a front cutting through the stencil, are filled in as ghost cells, with the state values obtained by extrapolation from nearby front

*Received by the editors May 1, 2001; accepted for publication (in revised form) April 2, 2003; published electronically November 19, 2003. The research of the first author was supported by the Applied Mathematics Subprogram of the U.S. Department of Energy DE-FG02-90ER25084, the Department of Energy Office of Inertial Fusion, the Army Research Office, grant DAAL-03-91-0027, the National Science Foundation, grant DMS-0102480, and Los Alamos National Lab, sponsor ID 26730001014L. The research of the third author was supported by the U.S. Department of Energy, sponsor ID DEFG0398DP. The research of the fourth author was supported by the U.S. Department of Energy, sponsor ID DEFG0290ER and Los Alamos National Lab, sponsor ID 26730001014L.

http://www.siam.org/journals/sinum/41-5/38862.html

[†]Department of Applied Mathematics and Statistics, SUNY at Stony Brook, Stony Brook, NY 11794-3600 (glimm@ams.sunysb.edu, linli@ams.sunysb.edu, yingjie@ams.sunysb.edu).

[‡]Center for Data Intensive Computing, Brookhaven National Laboratory, Upton, NY 11793-5000.

[§]Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, P.R. China. The research of this author was supported in part by the National Science Foundation of China, grant 10072028.

states of the same component. Thus the state values are double-valued near the front, with the left-component states extending by extrapolation for a small distance into the right component, and vice versa. The use of ghost cell states was introduced into front tracking in 1980 [11]. With the ghost states thus defined, the interior solver follows a conventional finite difference algorithm.

The algorithm proposed in the present study is conservative for all grid cells, including the irregular ones cut by the front. This algorithm presented is related to earlier work of Swartz and Wendroff [18], Harten and Hyman [14], Chern and Colella [2], and Pember et al. [16] but differs from these works in several ways. Chern and Colella and Pember et al. redistribute mass from small cells to nearby large ones to preserve stability and conservation. This issue is addressed here by merging small cells. Swartz and Wendroff discussed only the 1D algorithm. Pember et al. [16] reviews these earlier works in 1995. We emphasize here tracking of a contact rather than the shock tracking of [2].

2. Conservative tracking. Consider the 1D system of conservation laws

$$\frac{\partial u}{\partial t} + \bigtriangledown \cdot f(u) = 0. \tag{1}$$

Weak or discontinuous solutions of this equation are not unique, and the equation must be supplemented by an entropy condition [17]. In the case of discontinuities, the partial derivatives in (1) are not defined, and Rankine–Hugoniot conditions

$$n \cdot ([f] - v[u]) = 0 \tag{2}$$

apply. Here $[A] = A_+ - A_-$ is the jump in the quantity across the interface, v is the velocity of the interface, and n is a unit normal to the interface. In fact, (2) results from (1) if the derivatives in (1) are interpreted in the sense of distributions. Representing (1) in integral form, for a moving discontinuity surface S bounding a time-dependent volume V, we have

$$\frac{\partial}{\partial t}\int_V u dV + \int_S n \cdot (f(u) - vu) dS = 0. \tag{3}$$

Thus $n \cdot (f - vu)$ is the dynamic flux, which replaces the usual flux f for the time-independent surface.

The essence of the new algorithm introduced here is to track the front in space and time, based on the following three steps:

1. Construction of the space-time interface to follow the moving solution discontinuity. This will follow the grid-based construction [7] and extend it to space-time.
2. Construction of space-time finite volume cells, starting as a partition of a regular space-time cell. The cells cut by the space-time interface are defined as irregular. To ensure an adequate Courant–Friedrichs–Lewy (CFL) restriction, portions of such irregular cells with too small a top (at t_{n+1}) or no top at all are merged with neighbor cells.
3. Godunov-type finite volume differencing with limiters to ensure continuity of the dynamic flux (3), so that the algorithm is conservative on a cell by cell basis.

To explain these steps at a more detailed but still simple level, we consider in one dimension an interface whose position at time t is $\sigma_e(t)$, and we assume a linear approximation $\sigma(t)$ to $\sigma_e(t)$ on $[t_n, t_{n+1}]$. The 1D algorithm is divided into two cases. We consider only the first case, in which the cell merger from step 2 above is not required. We assume that the approximate interface does not cross a mesh cell center within the time interval $[t_n, t_{n+1}]$. Thus for some mesh index i, $x_i \le \sigma(t_n), \sigma(t_{n+1}) \le x_{i+1}$. We displace the cell boundary located at $x_{1+1/2}$ to the interface location. This change results in a redefinition of the cell average quantity, to yield

$$U_i^m = (\Delta_{i,e}^m)^{-1} \int_{x_{i-1/2}}^{\sigma_e(t_m)} u(x, t_m) dx, \tag{4}$$

$$U_{i+1}^m = (\Delta_{i+1,e}^m)^{-1} \int_{\sigma_e(t_m)}^{x_{i+3/2}} u(x, t_m) dx \tag{5}$$

for $m = n, n+1$, where $\Delta_{j,e}^m$ is the interval over which U_j^m is averaged.

Denote by $\mathcal{U}_i^m$ on $[x_{i-1/2}, \sigma(t_m)]$ and $\mathcal{U}_{i+1}^m$ on $[\sigma(t_m), x_{i+3/2}]$ the numerical approximations of U_i^m and U_{i+1}^m, respectively, and let Δ_j^m be the interval over which $\mathcal{U}_j^m$ is averaged. Integrating the hyperbolic system over the two trapezoidal regions $[x_{i-1/2}, \sigma(t)] \times [t_n, t_{n+1}]$ and $[\sigma(t), x_{i+3/2}] \times [t_n, t_{n+1}]$, the finite difference equation for irregular cells is replaced by

$$\Delta_i^{n+1} \mathcal{U}_i^{n+1} = \Delta_i^n \mathcal{U}_i^n - \Delta t \{ \mathcal{F}_{\text{int}}^{n+1/2} - \mathcal{F}_{i-1/2}^{n+1/2} \}, \tag{6}$$

$$\Delta_{i+1}^{n+1} \mathcal{U}_{i+1}^{n+1} = \Delta_{i+1}^n \mathcal{U}_{i+1}^n - \Delta t \{ \mathcal{F}_{i+3/2}^{n+1/2} - \mathcal{F}_{\text{int}}^{n+1/2} \}, \tag{7}$$

where $\mathcal{F}_{\text{int}}^{n+1/2}$ is the numerical approximation to the flux

$$F_{\text{int}}^{n+1/2} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \left(f(u(\sigma_e(t), t)) - s_e u(\sigma_e(t), t) \right) dt \tag{8}$$

across the exact interface. Here $\sigma_e(t)$ and $s_e = d\sigma_e/dt$. The definition (8) gives equal values when evaluated on either side of the interface due to the Rankine–Hugoniot condition (2).

Let $s(t) = d\sigma/dt$ be the speed of the numerically tracked interface $\sigma(t)$. The choice of the numerical shock speed is discussed in [3]. Assume a smooth solution in the interior region excluding the tracked waves. Also we assume that the Riemann solution associated with (1) depends Lipschitz-continuously on the left and right states which define the Riemann problem. Using a second order monotonic upstream-centered scheme for conservation law (MUSCL) reconstruction, we first reconstruct a piecewise linear function on each cell out of the cell averages at $t = t_n$ to yield the approximate left and right states $\mathcal{U}_l^n, \mathcal{U}_r^n$ at $\sigma_e(t_n)$ so that $\mathcal{U}_l^n - u(\sigma_e(t_n)-, t_n) = \mathcal{O}(\Delta x^2)$ and $\mathcal{U}_r^n - u(\sigma_e(t_n)+, t_n) = \mathcal{O}(\Delta x^2)$. Solving the Riemann problem with the above two approximate states, we obtain a shock speed s^n which satisfies $s^n - s_e(t_n) = \mathcal{O}(\Delta x^2)$. Therefore, the approximate tracked interface position at $t = t_n + \frac{1}{2}\Delta t$ is

$$\sigma^{n+1/2} = \sigma(t_n) + \frac{1}{2}\Delta t \cdot s^n = \sigma_e(t_{n+1/2}) + \mathcal{O}(\Delta t^2).$$

Using a Taylor expansion, we reconstruct the approximate left and right states $\mathcal{U}_l^{n+1/2}$, $\mathcal{U}_r^{n+1/2}$ at $(\sigma^{n+1/2}, t_{n+1/2})$ from the MUSCL reconstruction so that

$$\mathcal{U}_l^{n+1/2} - u(\sigma_e(t_{n+1/2})-, t_{n+1/2}) = \mathcal{O}(\Delta x^2) \tag{9}$$

and

$$\mathcal{U}_r^{n+1/2} - u(\sigma_e(t_{n+1/2})+, t_{n+1/2}) = \mathcal{O}(\Delta x^2). \tag{10}$$

Finally, solving a Riemann problem with the left and right states $\mathcal{U}_l^{n+1/2}$ and $\mathcal{U}_r^{n+1/2}$, we obtain the half time step shock speed $s^{n+1/2} = s_e(t_{n+1/2}) + \mathcal{O}(\Delta t^2)$, and the new two sides states $\mathcal{U}_{l1}^{n+1/2}$ and $\mathcal{U}_{r1}^{n+1/2}$ across the interface we want to track. Since the exact solution is smooth near the interface, the new states still satisfy (9) and (10). This construction gives a local error $\mathcal{O}(\Delta x^3)$ for the propagated shock position

$$\sigma_e^{n+1} = \sigma^n + s^{n+1/2}\Delta t + \mathcal{O}(\Delta t^3).$$

In fact,

$$\begin{aligned} \sigma_e^{n+1} - \sigma^n &= \int_{t_n}^{t_{n+1}} s_e(t)dt \\ &= \int_{t_n}^{t_{n+1}} [s_e(t_{n+1/2}) + s_e'(t_{n+1/2})(t - t_{n+1/2}) + \mathcal{O}(\Delta t^2)]dt \\ &= s_e(t_{n+1/2})\Delta t + \mathcal{O}(\Delta t^3) \end{aligned} \tag{11}$$

to give the desired accuracy. Let the numerical flux across the tracked front associated with the Riemann problem defined by these two states be

$$\mathcal{F}_i^{n+1/2} = f(\mathcal{U}_{l1}^{n+1/2}, t_{n+1/2}) - s^{n+1/2}\mathcal{U}_{l1}^{n+1/2}.$$

This flux satisfies

$$\mathcal{F}_i^{n+1/2} = F_i^{n+1/2} + \mathcal{O}(\Delta x^2)$$

and is continuous when evaluated from either side of the discontinuity.

The proof that this algorithm is conservative and (for one dimension only) improves its convergence rate near the tracked discontinuity by $\mathcal{O}(\Delta x^2)$ is given in [9].

3. The 2D algorithm. Consider the two space dimensional system of conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = 0, \tag{12}$$

defined in a spatial domain Ω. The discontinuities of u, assumed to lie on curves, are organized to form an *INTERFACE*, which is propagated from one time level to the next.

In the present study, we require at each time level that the *INTERFACE*s are topologically equivalent to a union of nonintersecting line segments or circles [13]. Thus we postulate that triple or multiple *CURVE* intersection points do not occur. Each *CURVE* is assigned an orientation which remains unchanged during the propagation of the *INTERFACE*. The discretized *CURVE* is piecewise linear and connected and composed of *BOND*s. Each *BOND* is a pair of *INTERFACE POINT*s or *POINT*s and (conceptually) the straight line segment joining them. Assume a decomposition of the plane by a rectangular grid with mesh spacing Δx, and assume the boundary $\partial\Omega$ of Ω lies on grid lines. If the *POINT*s are all on the interior of cell edges with at most one *POINT* occurring on the interior of any given grid cell edge, then the *INTERFACE* is called grid-based [7].

The front *POINT*s are propagated through the Riemann solutions in the normal direction followed by a tangential sweep to update the states on the front. Propagation [5, 7, 6, 4] of the *POINT*s of a grid-based *INTERFACE* will yield a general *INTERFACE*, not grid-based, as there is no reason for a propagated *POINT* to lie on a grid cell edge just because it starts on one. The general idea of the grid-based construction is as follows: we consider this propagated *INTERFACE* as a collection of polygonal *CURVE*s in $\Re^2$. Crossing points of the *CURVE* with grid cell edges are inserted as new *POINT*s. The propagated old *POINT*s (named images of propagation in this sense) will be deleted. The *CURVE* is then reconstructed as straight line segments joining these new *POINT*s. In this process, the *CURVE* is displaced by an amount $\mathcal{O}(\Delta x^2)$, assuming that the *CURVE* is smooth, so that all angles between neighboring *BOND*s are $\mathcal{O}(\Delta x)$. See also [15, 7, 6] for detailed discussions of the grid-based INTERFACE construction.

Let B_i^n be a *BOND* on the grid-based *INTERFACE* $\mathcal{I}^n$ at the old time step t^n, and let $\bar{B}_i^{n+1}$ be the image *BOND* after the propagation of the end *POINT*s of B_i^n. A new grid-based *INTERFACE* $\mathcal{I}^{n+1}$ is reconstructed through the new *POINT*s which are produced by the intersection of the image *BOND*s and the gridline segments. Therefore, each new *POINT* P_i^{n+1} corresponds to an old *BOND* B_i^n, but the inverse is not true, because some bonds will not intersect with any gridline segment. On the other hand, since an image *BOND* may have several intersections with different gridline segments, several new *POINT*s may correspond to a single *BOND* B_i^n.

The finite difference method presented here for (12) is an explicit finite volume integration scheme. The spatial domain Ω has two dimensions. The solution of u evolves with respect to time, and we treat the temporal dimension as the third dimension. We join the spatial *INTERFACE*s at two consecutive time steps to form a space-time interface. Assume we have a space-time discretization $\{\mathcal{V}_i\}$ which conforms to the space-time interface as u evolves in one time step from time t_n to t_{n+1}. We solve (12) explicitly in this region. Treating each $\mathcal{V}_i$ as a control volume, we integrate (12) over $\mathcal{V}_i$. By the divergence theorem, we have

$$|\mathcal{V}_i(t_{n+1})|\bar{u}\,|_{t_{n+1}}= |\mathcal{V}_i(t_n)|\bar{u}\,|_{t_n} - \int_{\partial\mathcal{V}_i}(u, f(u), g(u)) \cdot ndS, \tag{13}$$

where $\bar{u}\,|_{t_{n+1}}= \frac{1}{|\mathcal{V}_i(t_{n+1})|}\int_{|\mathcal{V}_i(t_{n+1})|} u(x, y, t_{n+1})dxdy$ is defined as a cell average, $|\mathcal{V}_i(t_{n+1})|$ is the face area of $\mathcal{V}_i(t_{n+1})$ at time t_{n+1}, and n is the outward normal to the space-time surfaces of $\mathcal{V}_i$. We wish to calculate $\bar{u}\,|_{t_{n+1}}$, the solution to (12) at time t_{n+1}.

The major issues in designing the conservative algorithm are (1) to obtain the space-time *INTERFACE*, (2) to determine the discretization $\{\mathcal{V}_i\}$, and (3) to calculate the fluxes defined on the space-time surfaces of $\mathcal{V}_i$.

To construct a finite volume decomposition which respects the space-time interface, we identify the crossings of the approximate space-time interface with the space-time hexahedron. We split the space-time hexahedron whose interior is cut by the space-time interface into parts, each of which belongs to only one side of the space time interface. For the purpose of maintaining numerical stability (the CFL time step restriction), we merge those cells with small top area to form a polyhedron with top area bigger than $0.5\Delta x^2$.

3.1. Construction of the space-time interface. In the current section, we solve the following problem: given two piecewise linear spatial grid-based *INTERFACE*s (*CURVE*s) which are separated in time by a step $\triangle t$, construct (triangulate) a surface joining them. We call this joining surface the space-time interface. The space-time interface thus formed is also grid-based, as a three-dimensional (3D) interface (two spatial and one temporal dimensions). The local configurations within a single grid cell for such a 3D grid-based interface have been discussed in [6]. We introduce two hypotheses regarding the old and new spatial interfaces. These hypotheses limit the local complexity of the interface. More complicated topological structures will not be included in the scope of this paper.

HYPOTHESIS 1. *The INTERFACE is assumed to be grid-based. There is no topological change of the INTERFACE during the time interval of computation. Each CURVE is topologically equivalent to a line segment with its two end points on the boundary, or a circle contained in the interior of* Ω. *No CURVE is totally contained within a square of side* $2\Delta x$ *made up of four cells.*

HYPOTHESIS 2. *The CFL number is less than* $\frac{1}{2}$ *so that each POINT of the INTERFACE is propagated a distance less than* $\frac{1}{2}\Delta x$ *within a single time step.*

Assume Hypothesis 1. For a grid-based *INTERFACE* $\mathcal{I}^n$, each *POINT* on $\mathcal{I}^n$ is a crossing *POINT*; there exists at most one crossing *POINT* on each grid cell edge. No crossing is deleted during the reconstruction of the grid-based *INTERFACE*, as such deletion would indicate a change of topology. Propagation of $\mathcal{I}^n$ *POINT*s at one single time step gives a new *INTERFACE* $\mathcal{I}_0^{n+1}$. The new grid-based *INTERFACE* $\mathcal{I}^{n+1}$ is reconstructed from $\mathcal{I}_0^{n+1}$ through the algorithm discussed above.

After the reconstruction of $\mathcal{I}_0^{n+1}$, the order of *POINT*s on the reconstructed *INTERFACE* $\mathcal{I}^{n+1}$ agrees with the natural order of the *POINT*s on $\mathcal{I}^n$ in the following sense: Let B_1^n be an $\mathcal{I}^n$ *BOND* connecting adjacent *POINT*s P_1 and P_2. Let B_2^n be an $\mathcal{I}^n$ *BOND* following B_1^n, connecting adjacent *POINT*s P_2 and P_3. After propagation, B_1^n is mapped onto an $\mathcal{I}_0^{n+1}$ linear segment B_1^{n+1} with a left end point M_1 as the image of P_1 and a right end point M_2 as the image of P_2; similarly, B_2^n is mapped onto B_2^{n+1} of $\mathcal{I}_0^{n+1}$ with a left end point M_2 and a right end point M_3 as the image of P_3. The reconstruction first inserts into $\mathcal{I}_0^{n+1}$ the crossing points of $\mathcal{I}_0^{n+1}$ with grid lines as new *POINT*s. The insertion of new *POINT*s does not change the orientation or order of the polygon $B_1^{n+1} \bigcup B_2^{n+1}$, which preserves its order from the polygon $B_1^n \bigcup B_2^n$. Similarly, the removal of *POINT*s, with the deformation of the polygon to connect with the remaining *POINT*s by linear segment, is order-preserving. Thus grid-based $\mathcal{I}^{n+1}$ reconstructed from $\mathcal{I}_0^{n+1}$ by connecting the new *POINT*s as above described preserves the *POINT* order.

For the grid-based method, every *INTERFACE POINT* lies on a cell edge. A *POINT* P is assigned on index (i, j) if it is located within a half grid size $(0.5\Delta x)$ away from the grid node (i, j). The proximity *Prox* P of P includes nine dual grid cells centered at grid node ($i-1$ to $i+1$, $j-1$ to $j+1$).

Assume P_1 and P_2 are the start and end *POINT*s of *BOND* B^{n+1} on $\mathcal{I}^{n+1}$; the (i, j) indices of these two *POINT*s can be identical, adjacent, or diagonally adjacent. The proximity *Prox* B of *BOND* B^{n+1} is defined as *Prox* $P_1 \cap$ *Prox* P_2. Therefore, the following hold.

(1) If (i, j) indices of P_1 and P_2 are identical, say, both are (i, j), *Prox* B is the nine dual grid cells centered at nodes ($i-1$ to $i+1$, $j-1$ to $j+1$).

(2) If (i, j) indices of P_1 and P_2 are adjacent, say, (i, j) for P_1 and $(i+1, j)$ for P_2, then *Prox* B includes the six dual grid cells centered at nodes (i to $i+1$, $j-1$ to $j+1$). *Prox* B for the case in which the second index of P_1 and P_2 differs by 1 is similarly defined.

(3) If (i, j) indices of P_1 and P_2 are diagonally adjacent, say, (i, j) for P_1 and $(i+1,j+1)$ for P_2, *Prox* B includes the four dual grid cells centered at nodes (i to $i+1$, j to $j+1$).

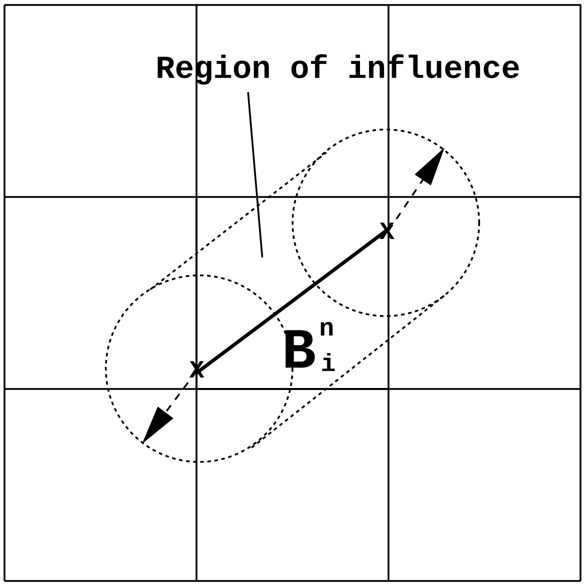

FIG. 1. *Region of influence of a bond B_i^n.*

A *POINT* of $\mathcal{I}^n$ which is located inside the proximity of B^{n+1} is called a *spatially nearest POINT* of B^{n+1} on $\mathcal{I}^n$. If a *BOND* B^{n+1} of $\mathcal{I}^{n+1}$ is the result of propagation followed by the grid-based reconstruction of a single *BOND* B^n on $\mathcal{I}^n$, then B^n is called the *parent BOND* and B^{n+1} is the *child BOND*. In this case, B^{n+1} is formed by the insertion of crossing *POINT*s into the propagated image of B^n. The *region of influence* of any *BOND* in Figure 1 is the region within $0.5\Delta x$ of the points on the *BOND*.

PROPOSITION 1. *Assume Hypotheses* 1, 2. *Let* P_1 *and* P_2 *be two adjacent POINTs connected by a reconstructed grid-based bond* B^{n+1} *of* $\mathcal{I}^{n+1}$. *If* P_1 *is produced (through propagation and intersection of a bond with a gridline segment) by* B_1^n *of* $\mathcal{I}^n$, P_2 *is produced by* B_2^n *of* $\mathcal{I}^n$, *and the curve on* $\mathcal{I}^n$ *is oriented so that* B_2^n *follows* B_1^n, *then there exists at least one POINT between the start of* B_1^n *and the end of* B_2^n *which lies within the proximity of* B^{n+1}.

Proof. Let C with corner nodes (i,j), $(i+1,j)$, $(i,j+1)$, and $(i+1,j+1)$ be the cell containing B^{n+1}. Assume $B_1^n \neq B_2^n$. In this case, to produce B^{n+1}, all the propagated *POINT*s between (including) the end of B_1^n and the start of B_2^n must lie in the cell C at the new time step t^{n+1}. By Hypothesis 2, all the corresponding old points must be in the proximity defined by P_1 and P_2, because the shortest distance from the boundary of the cell C to the boundary of the proximity is at least $0.5\Delta x$.

Next we assume $B_1^n = B_2^n$. In this case, B_1^n is the *parent BOND* of B^{n+1} and the entire B^{n+1} must be within the *region of influence* of B_1^n. Since the proximity of B^{n+1} is the intersection of the proximities of the two *POINT*s P_1 and P_2, it is the smallest when the indices of P_1 and P_2 are diagonally adjacent, a property we now assume. The proximity is the rectangle *ABCD* in Figure 2. We want to prove that at least one *POINT* of B_1^n is located within the rectangle *ABCD*. To prove this, we show that the parent *BOND* B_1^n cannot have both *POINT*s outside the proximity.

We now draw the boundary of the *region of influence* of all the grid-based bonds with both end *POINT*s outside the proximity of B^{n+1}. The inner boundary of this

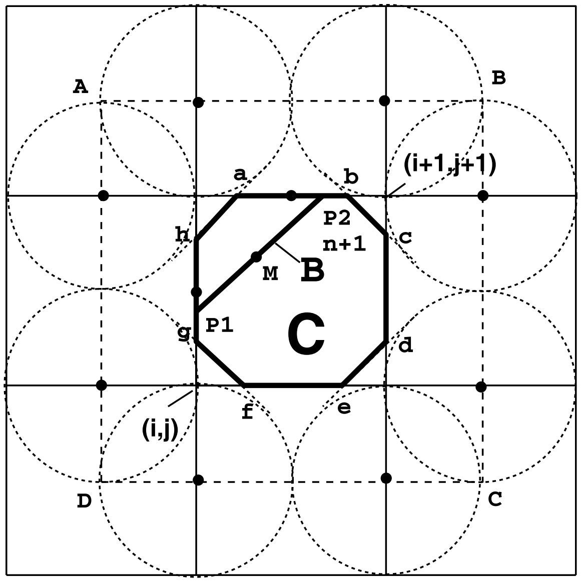

FIG. 2. *Region of influence of all bonds completely outside the proximity of B^{n+1} of which the (i,j) indices of B^{n+1} end points P_1 and P_2 are diagonally adjacent.*

region is the polygon **abcdefgh** as in Figure 2. If the parent *BOND* B_1^n has both end points outside the proximity, then B^{n+1}, lying in its region of influence, should be completely outside the polygon **abcdefgh**. Thus it must lie in one of the four small regions near one corner of the cell C. Since the polygon cuts the edges of the cell C at a distance $0.5(\sqrt{2}-1)\Delta x \approx 0.207\Delta x$ from the four cell corners, the mesh indices of the end *POINT*s of B^{n+1} cannot be diagonally adjacent. Therefore, no bond with both end *POINT*s outside the proximity *ABCD* can be the *parent* bond of B^{n+1}. Therefore, at least one end point of B^n must be located within the proximity *ABCD*.

The other two cases, when the indices of P_1 and P_2 are identical or adjacent, have a larger proximity for B_1^n. Similar but easier arguments prove Proposition 1 in these cases. This completes the proof.

PROPOSITION 2. *Assume Hypotheses* 1, 2. *Let* B_1^{n+1} *with end points* P_1 *and* P_2, *and* B_2^{n+1} *with end point* P_2 *and* P_3 *be two adjacent BONDs on* $\mathcal{I}^{n+1}$ *in their natural order. Let* B_1^n, B_2^n, *and* B_3^n *be the bonds on* $\mathcal{I}^n$ *which produce* P_1, P_2, *and* P_3. *Denote the spatially nearest POINTs to* B_1^{n+1} *on* $\mathcal{I}^n$ *as group* 1 *and the spatially nearest POINTs to* B_2^{n+1} *on* $\mathcal{I}^n$ *as group* 2. *There exist a POINT* M_1 *in group* 1 *and a POINT* M_2 *in group* 2 *such that* (1) M_1 *is a POINT between (including) the start of* B_1^n *and the end of* B_2^n, *and* M_2 *is a POINT between (including) the start of* B_2^n *and the end of* B_3^n; (2) M_1 *precedes or equals* M_2 *in the orientation of* $\mathcal{I}^n$.

Proof. If B_1^n, B_2^n, and B_3^n are three distinct bonds, we take M_1 as any *POINT* between (including) the end of B_1^n and the start of B_2^n and M_2 as any *POINT* between (including) the end of B_2^n and the start of B_3^n. This choice satisfies Proposition 2 in view of Proposition 1.

Next we consider the case $B_1^n = B_2^n \neq B_3^n$. We select M_1 as one of the end *POINT*s of B_1^n which is in group 1. Such a *POINT* exists by Proposition 1. M_2 can be selected between (including) the end of B_2^n to the start of B_3^n. M_1 and M_2 satisfy Proposition 2. The case $B_1^n \neq B_2^n = B_3^n$ is similar.

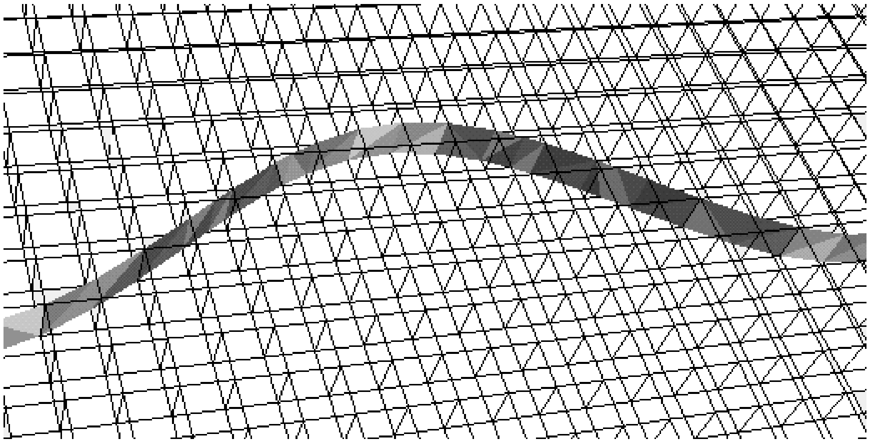

FIG. 3. *The triangulated space-time interface.*

Finally, we consider $B_1^n = B_2^n = B_3^n$. In this case, P_1, P_2, and P_3 lie on a straight line. It is obvious that Proposition 2 holds for the following three cases: (1) the start of B_1^n in both $Prox\ B_1^{n+1}$ and $Prox\ B_2^{n+1}$; (2) the end of B_1^n in both $Prox\ B_1^{n+1}$ and $Prox\ B_2^{n+1}$; and (3) the start of B_1^n in $Prox\ B_1^{n+1}$ and the end of B_1^n in $Prox\ B_2^{n+1}$.

We now prove that it is impossible to have the end of B_1^n in $Prox\ B_1^{n+1} \setminus Prox\ B_2^{n+1}$ and the start of B_1^n in $B_2^{n+1} \setminus Prox\ B_1^{n+1}$. For this to occur, the propagation of both the start and the end points of B_1^n must completely pass through the region of $Prox\ B_1^{n+1} \cap Prox\ B_2^{n+1}$. It is easy to verify that the widths of the intersection in both the x and y directions are at least Δx. However, the maximum propagation distance in one time step is $0.5\Delta x$. The proof is complete.

The surface triangles in space-time are formed by joining the $POINTs$ of $\mathcal{I}^{n+1}$ and $\mathcal{I}^n$. Each triangle has a side taken from a single linear segment ($BOND$) of either $\mathcal{I}^{n+1}$ or $\mathcal{I}^n$ and an opposite $POINT$ from the other. We denote a space-time interface triangle which is composed of a $BOND$ at time t_{n+1} and an opposite $POINT$ from $\mathcal{I}^n$ as an upper triangle, and a triangle which is composed of a $BOND$ at time t_n and an opposite $POINT$ from $\mathcal{I}^{n+1}$ as a lower triangle. The space-time interface triangulation is organized into the following two steps:

1. We first form the upper triangles of the space-time interface. For each $\mathcal{I}^{n+1}$ $BOND\ B_m^{n+1}$ whose start and end $POINT$s are P_m and P_{m+1}, we find by Proposition 1 the spatially nearest $POINT$s on $\mathcal{I}^n$. Denote these $POINT$s as group m. Select one $POINT\ M_m$ from each group m to form the list $[M_1, M_2, M_3, \dots]$ with the same orientation as $\mathcal{I}^n$ (due to Proposition 2); M_i and M_{i+1} are not necessarily distinct. Connect each M_m to P_m and P_{m+1} to form upper triangles.
2. The gap on the space-time interface left by step 1 is filled by lower triangles. Each $BOND\ B_k^n$ on the $\mathcal{I}^n$ is located between a pair of distinct $POINT$s M_k and M_{k+1}. M_k and M_{k+1} are connected to a common $POINT$ on $\mathcal{I}^{n+1}$ during the construction of the upper triangles. Connect this common $POINT$ to the start and end $POINT$s of B_k^n to form a lower triangle. This completes the space-time interface triangulation.

Figure 3 shows the triangulated space-time interface. Each triangle is distinguished from its neighbors by contrasting grey shades.

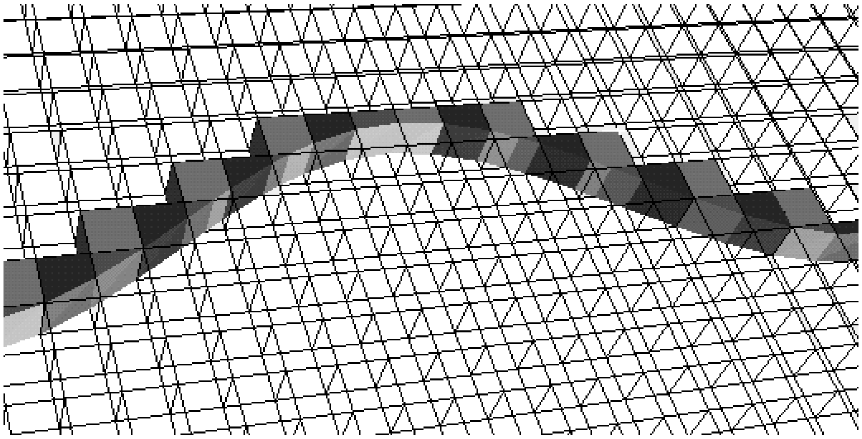

FIG. 4. *Hexahedra and partial polyhedra before volume merging.*

3.2. Construction of the space-time hexahedra. We connect the nodes of a cell D_i^n at time $t = t_n$ to the nodes of the corresponding cell D_i^{n+1} at time $t = t_{n+1}$ to form a space-time hexahedron. We call D_i^{n+1} the top of the hexahedron and D_i^n the bottom. We call a hexahedron *mixed* if the interface passes through its interior, otherwise, it is *pure*. The *mixed* hexahedra are divided into *pure partial* hexahedra, and if necessary, these are combined with neighbors to form a finite volume space-time grid suitable for construction of a conservative difference algorithm in section 3.3. They are adjacent if they share a nontrivial surface which does not meet the space-time interface. Two space-time polyhedra are neighboring if they share a nontrivial vertical line segment which is part of the grid line connecting two corresponding grid nodes at the time levels t_n and t_{n+1} (denoted by a vertical grid line) that does not cross the space-time interface. It is easy to see that two adjacent or neighboring polyhedra must be on the same side of the space-time interface.

The *mixed* hexahedron is separated by the space-time interface into several parts, each of which lies on one side of the space-time interface. These parts are called *pure partial* hexahedra or, in short, *partial* hexahedra. We can similarly define a cell to be *pure*, *mixed*, or *partial*. Any *partial* hexahedron with a trivial or small top will be merged with an adjacent *pure* hexahedron or *partial* hexahedron having a top of minimal size.

Figure 4 shows the control volumes constructed on one side of the space-time interface. Adjacent hexahedra or pure partial polyhedra are represented by contrasting grey shades. Only the volumes near the space-time interface are displayed.

Recalling that two adjacent hexahedra are on the same side of the interface, the following lemma [8] ensures the eventual success of the merging algorithm.

LEMMA 1. *Assume Hypothesis* 1. *If a space-time polyhedron is constructed by merging any number of adjacent partial hexahedra with no top, then the polyhedron will be adjacent to a pure or partial hexahedron on the same side of the space-time interface.*

Proof. At least one nontrivial piece of the side surface of the polyhedron is not on the boundary or the space-time interface; otherwise, the topological structure of the *INTERFACE* changes during this time step and Hypothesis 1 is violated. The proof is complete.

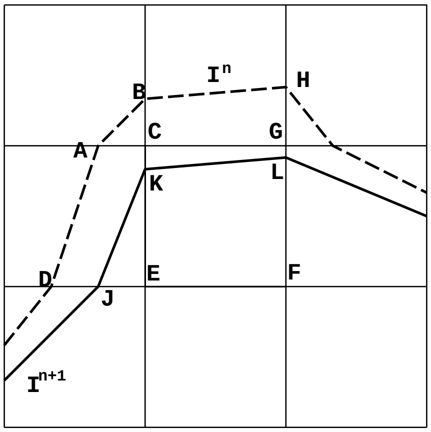

Fig. 5. *A top view of the polyhedra merging process. The solid line represents $\mathcal{I}^{n+1}$, and the dashed line represents $\mathcal{I}^n$. There are four polyhedra in the four upper left mesh blocks: polyhedron* 1 *with bottom ABC and no top, polyhedron* 2 *with bottom face $BHGC$ and no top, polyhedron* 3 *with bottom face $ACED$ and the triangular top KEJ, and polyhedron* 4 *with a square bottom $CGFE$ and the trapezoidal top $KLFE$. They will be merged into one polyhedron with bottom $ACBHGFED$ and top $KLFEJ$.*

Hypotheses 1 and 2 and Lemma 1 ensure that each partial hexahedron with no top and away from the boundary is adjacent to or neighboring one with a nontrivial top. However, for a partial hexahedron with no top and at the boundary, Hypothesis 2 may not be sufficient if the interface intersects the boundary at a small angle. We need to adjust the CFL number so that the intersection point between the *INTERFACE* and the boundary moves a distance less than Δx along the boundary during the time step in order to reach the same property.

We require a hypothesis to limit the local geometric complexity of the *INTERFACE*. To simplify the proof that the merging algorithm converges (rapidly), we state it in a stronger than necessary form. See section 3.4 for a discussion of this issue.

Hypothesis 3. *Each partial hexahedron having top area smaller than $\frac{1}{2}\Delta x^2$ is adjacent to or neighboring one with top area greater than or equal to $\frac{1}{2}\Delta x^2$.*

Because the flux exchange among control volumes is through the shared space-time surfaces, we merge only adjacent *partial* hexahedra on the same side of the space-time interface and not neighboring ones. For this reason, merger is accomplished in stages, i.e., recursively. The merging process then is stated as follows.

Assume Hypothesis 3. *Recursively merge every pure or partial hexahedron having a top area greater than or equal to $\frac{1}{2}\Delta x^2$ with adjacent partial hexahedra having no top or top area smaller than $\frac{1}{2}\Delta x^2$ which have not been merged elsewhere until none of the partial hexahedron having no top or top area smaller than $\frac{1}{2}\Delta x^2$ is left. Denote the resulting space-time polyhedra the big hexahedra. The merging process then is complete. Partial polyhedra generated at each merging stage are called intermediate hexahedra.*

As illustrated in Figure 5, polyhedron 4 with a square bottom face $CGFE$ and top face area $KLFE$ greater than $\frac{1}{2}\Delta x^2$ forms the center of merging. The merged polyhedra include polyhedron 3 with bottom face $ACED$ and a small triangular top KEJ, polyhedron 2 with bottom face $BHGC$ and no top, and polyhedron 1 with bottom face ABC and no top. Polyhedra 2 and 3 are adjacent to 4, while polyhedron

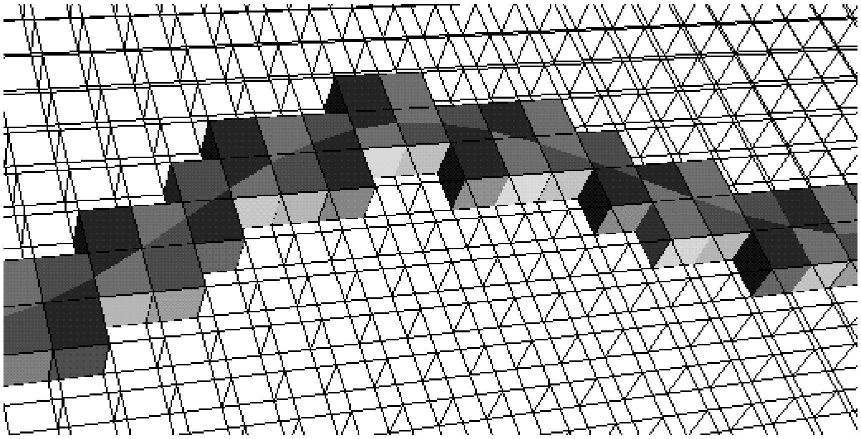

FIG. 6. *Control volumes after merging.*

1 is diagonally adjacent to 4. In the merging process, polyhedra 2 and 3 are absorbed by polyhedron 4 first. A *intermediate* hexahedron with bottom face $ACBHGFED$ is formed. Polyhedron 1 is adjacent to it. Finally, this *intermediate* hexahedron absorbs polyhedron 1, resulting in a *big* hexahedron with bottom face $ABHGFED$ and top face $KLFEJ$.

Determined by Lemma 1 and Hypothesis 3, it is easy to see that a *big* hexahedron contains no more than a fixed number of *pure* or *partial* hexahedra so that the merging process stops rapidly. Actually in most cases the merging process yields *big* hexahedra consisting of two *pure* or *partial* hexahedra. The number of *pure* or *partial* hexahedra in the *big* hexahedron could become larger if the radius of curvature of the moving *CURVE* is small. In fact, we have the following observation.

Assume Hypothesis 3. Let a *pure* or *pure partial* hexahedron $\mathcal{H}$ with top area greater than $\frac{1}{2}\Delta x^2$ be contained inside a space-time cell with cell index (i, j). If $\mathcal{H}$ forms a *big* hexahedron by absorbing *pure partial* hexahedra during the merging process, the bottom faces of these *pure partial* hexahedra which merge with $\mathcal{H}$ are located inside a square, centered at (i, j), with side $3\Delta x$.

Figure 6 shows the control volumes on two sides of the space-time interface after the merging process. Only the volumes near the space-time interface are displayed.

THEOREM 1. *Assume Hypotheses* 1–3. *After the merging algorithm, every partial hexahedron having no top or top area smaller than* $\frac{1}{2}\Delta x^2$ *will be merged into a big hexahedron having top area greater than or equal to* $\frac{1}{2}\Delta x^2$ *on the same side of the interface. The interior of each big hexahedron is connected.*

3.3. The reconstruction, limiter, and numerical scheme. Suppose at the time level $t = t_n$ we know the approximate state averages on each cell, *regular*, *irregular*, or *partial*. We want to reconstruct a piecewise linear state function on these cells with second order accuracy. The reconstruction of the piecewise linear state function on *irregular* cells follows [1], with modifications to the limiter and some simplification. Let D_i^n be a *pure* cell, *regular*, *irregular*, or *partial*, with approximate state average $\mathcal{U}_i$ and cell center (centroid) Y_i, surrounded by any of the types of cells $D_j^n, D_k^n, D_l^n, D_m^n$ with approximate state averages $\mathcal{U}_j^n, \mathcal{U}_k^n, \mathcal{U}_l^n, \mathcal{U}_m^n$ and cell centers Y_j, Y_k, Y_l, Y_m, respectively, on the same side of the $INTERFACE$. Let $\tilde{\mathcal{U}}_i = \mathcal{U}_i + (a, b) \cdot$

$(X - Y_i)$ be the second order accurate linear state function on D_i^n, where a, b are two constants. Choose any two surrounding cells, say, D_j^n, D_k^n so that Y_i, Y_j, Y_k are not colinear. We can determine a, b by solving the following equation:

$$\begin{aligned} \tilde{\mathcal{U}}_i(Y_j) &= \mathcal{U}_j^n, \\ \tilde{\mathcal{U}}_i(Y_k) &= \mathcal{U}_k^n. \end{aligned} \tag{14}$$

Further, for the solution of the above equation to be well conditioned, we require the angle θ formed by line segments $\overline{Y_iY_j}$ and $\overline{Y_iY_k}$ to satisfy $0 < \theta_1 < \theta < \theta_2 < \pi$, where θ_1, θ_2 are two constants. We repeat the above procedure until we find all possible solutions, say, a_i, b_i, for all $0 \leq i \leq I$, where $I \leq 4$. Then we set $a = \text{minmod}\{a_1, \ldots, a_I\}$ and $b = \text{minmod}\{b_1, \ldots, b_I\}$. When there are not enough surrounding cells on the same side of the *INTERFACE*, we choose $a, b = 0$ so that the reconstruction becomes first order.

When D_i^n is a *regular* cell surrounded by *regular* cells, the reconstruction process is simpler. Let the cell center of D_i^n be $(i_1\Delta x, i_2\Delta y)$ with neighboring cell centers $\{((i_1 + k_1)\Delta x, (i_2 + k_2)\Delta y) | k_1, k_2 = -1, 0, 1\}$. Let

$$\begin{aligned} \text{xslope}_\text{i} = \; & \text{minmod}\{[\mathcal{U}((i_1 + k_1)\Delta x, (i_2 + k_2)\Delta y) \\ & -\mathcal{U}((i_1 + k_1 - 1)\Delta x, (i_2 + k_2)\Delta y)]/\Delta x \mid \\ & k_1 = 0, 1; k_2 = -1, 0, 1\}, \end{aligned} \tag{15}$$

and

$$\begin{aligned} \text{yslope}_\text{i} = \; & \text{minmod}\{[\mathcal{U}((i_1 + k_1)\Delta x, (i_2 + k_2)\Delta y) \\ & -\mathcal{U}((i_1 + k_1)\Delta x, (i_2 + k_2 - 1)\Delta y)]/\Delta y \mid \\ & k_1 = -1, 0, 1; k_2 = 0, 1\}, \end{aligned} \tag{16}$$

and define

$$\tilde{\mathcal{U}}_i = \mathcal{U}_i + \text{xslope}_\text{i} \cdot (x - i_1\Delta x) + \text{yslope}_\text{i} \cdot (y - i_2\Delta y).$$

This second order reconstruction is better suited in multiple dimensions than in the operator splitting single line reconstruction (or limiter) for a uniform rectangular grid because, for example, an untracked discontinuity in two dimensions may be in the form of a strip of width between $2\Delta x$ and $3\Delta x$. When the strip is almost parallel to and fully covers the line in which the single line reconstruction occurs, one cannot expect the limiter to choose any smooth solutions nearby.

Next we apply the technique of section 3.2 to generate space-time hexahedra between time levels t^n and t^{n+1}. Let H be a big hexahedron with top D^{n+1}, bottom D^n, and triangle sides $\{S_i\}$ with a unit outer normal n_i and centroid Z_i. Notice that some elements of the $\{S_i\}$ may be on the approximate space-time interface. Integrating (12) over H, we obtain

$$|D^{n+1}| \; U^{n+1} = |D^n| \; U^n - \sum_i \int_{S_i} (u, f, g) \cdot n_i ds. \tag{17}$$

Here $|D^n|$ represents the area of D^n, and similarly $|S_i|$ is the area of S_i.

The numerical scheme can be written as

$$|D^{n+1}|\mathcal{U}^{n+1} = |D^n|\mathcal{U}^n - \sum_i |S_i|(\tilde{\mathcal{U}}_{i,m}, f(\tilde{\mathcal{U}}_{i,m}), g(\tilde{\mathcal{U}}_{i,m})) \cdot n_i. \tag{18}$$

The fluxes through triangle sides $\{S_i\}$ can be calculated by a higher order Godunov-type algorithm.

We first calculate $\tilde{\mathcal{U}}_{i,m}$ as follows: First use a Cauchy–Kowalewski procedure on the reconstructed state function on each side of S_i to get second order approximate states at Z_i on the respective sides of S_i, say, $\mathcal{U}_{i,l}$ and $\mathcal{U}_{i,r}$. If S_i is not on the tracked space-time interface, we can simply use a Riemann solver, say, R, to get the middle state on S_i, i.e.,

$$\tilde{\mathcal{U}}_{i,m} = R(\mathcal{U}_{i,l},\ \mathcal{U}_{i,r}).$$

If S_i is on the tracked space-time interface, we use the Riemann solver to get the left and the right side states $\tilde{\mathcal{U}}_{i,l}$ and $\tilde{\mathcal{U}}_{i,r}$ on the wave we are supposed to track and the wave speed ν_i. Then $\tilde{\mathcal{U}}_{i,m}$ in (18) can be replaced by either $\tilde{\mathcal{U}}_{i,l}$ or $\tilde{\mathcal{U}}_{i,r}$, depending on whether l or r is located within H or not. Also, the n_i in (18) should be replaced by $\tilde{n}_i/|\tilde{n}_i|$, where $\tilde{n}_i = (-\nu_i\sqrt{n_{ix}^2 + n_{iy}^2}, n_{ix}, n_{iy})$, $n_i = (n_{it}, n_{ix}, n_{iy})$. Note that $\tilde{n}_i$ is normal direction of the tracked space-time wave from the Riemann solver; therefore, this modification ensures that the Rankine–Hugoniot condition is satisfied.

The finite volume difference algorithm constitutes a flux through each boundary of the full, *partial*, and *big* hexahedron. Since the flux through a boundary face of the hexahedron is identical when viewed from either side of the face, we have the following theorem.

Theorem 2. $\sum_{cells} |D^n|\mathcal{U}^n$ *in the finite volume difference scheme is conserved so that its increment over any time interval is equal to the net influx at the boundary.*

Away from the *INTERFACE* the scheme is clearly a second order scheme. For the cells along the *INTERFACE*, its local error is one order lower than in the 1D case since we use a piecewise linear approximation to the smooth *INTERFACE* and the local displacement error of this approximation is $\mathcal{O}(\Delta x^2)$. The scheme is one order better than untracked schemes, which typically have $\mathcal{O}(1)$ local error at the untracked fronts.

Theorem 3. *Suppose the exact space-time interface and the solution on either side of it are smooth. Then the L_∞ error is $\mathcal{O}(\Delta x)$ for cells adjacent to the INTERFACE.*

Proof. Let the *INTERFACE* at t_n be exact, and let H be a big hexahedron adjacent to the approximate space-time interface. We apply the finite volume scheme to obtain the approximate state average $\mathcal{U}_i^{n+1}$ at the time level t_{n+1}, with top T and bottom B and side boundaries $\{S_i\}$, where each S_i is a triangle. The *INTERFACE* at time t_{n+1} has an $\mathcal{O}(\Delta x^2)$ displacement from the exact interface. The exact space-time interface will cut H into two pieces. Let H_1 be the piece on the same side of the interface as H. Let T_1, B_1, and S^1 be the top, bottom, and side boundaries of H_1, respectively. Let $U_{T_1}^{n+1}, U_{B_1}^n$ be the exact state averages over T_1 and B_1, respectively. Choosing $\mathcal{U}_B^n = U_{B_1}^n$, we want to show that $U_{T_1}^{n+1} - \mathcal{U}_T^{n+1} = \mathcal{O}(\Delta x)$. In fact, from (18),

$$|T|\mathcal{U}_T^{n+1} = |B|\mathcal{U}_B^n - \sum_i |S_i|(\tilde{\mathcal{U}}_{i,m}, f(\tilde{\mathcal{U}}_{i,m}), g(\tilde{\mathcal{U}}_{i,m})) \cdot n_i. \tag{19}$$

The exact solution satisfies

$$|T_1|U_{T_1}^{n+1} = |B_1|U_{B_1}^n - \int_{S^1} (u, f(u), g(u)) \cdot n ds. \tag{20}$$

Note that $|B|\mathcal{U}_B^n = |B_1|U_{B_1}^n$. Also, the numerical flux in (19) approximates the exact flux in (20) to at least $\mathcal{O}(\Delta x^3)$. In fact, when S_i is not on the approximate space-time interface, this is easily seen since $\int_{S_i}(u,f,g)\cdot n_i ds = |S_i|(u,f,g)(Z_i)\cdot n_i + \mathcal{O}(\Delta x^4)$.

Next suppose that S_i is on the approximate space-time interface. Because of the smoothness of the exact space-time interface, it has an $\mathcal{O}(\Delta x^2)$ displacement error to the exact one. The difference between their respective areas is of $\mathcal{O}(\Delta x^3)$. The area of $\bigcup S_i$ is $\mathcal{O}(\Delta x^2)$. Also, the choices of $\tilde{\mathcal{U}}_{i,m}$ and n_i in (19) ensure that $(\tilde{\mathcal{U}}_{i,m}, f(\tilde{\mathcal{U}}_{i,m}), g(\tilde{\mathcal{U}}_{i,m}))\cdot n_i$ in (19) is a first order approximation to the integrand in (20) at any point within an $\mathcal{O}(\Delta x)$ distance from the centroid Z_i of S_i. Thus $\int_{S_i}(u,f,g)\cdot n_i ds = |S_i|(u,f,g)(Z_i)\cdot n_i + \mathcal{O}(\Delta x^3)$ in the case that S_i is on the approximate space-time interface. Therefore, we have

$$\begin{aligned} U_{T_1}^{n+1} - \mathcal{U}_T^{n+1} &= (|T_1|U_{T_1}^{n+1} - |T|\mathcal{U}_T^{n+1})/|T| + U_{T_1}^{n+1}((|T|-|T_1|)/|T|) \\ &= (\mathcal{O}(\Delta x^4) + \mathcal{O}(\Delta x^3))/\mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta x^3)/\mathcal{O}(\Delta x^2) \\ &= \mathcal{O}(\Delta x), \end{aligned} \tag{21}$$

where $\mathcal{O}(\Delta x^4)$ and $\mathcal{O}(\Delta x^3)$ in the first bracket follow from the local error of the numerical approximation of the flux defined on the non space-time interface and space-time interface, respectively. The proof is complete.

3.4. Cell level complexity and interface topological change. Because the dynamic evolution of the *INTERFACE* often leads to geometrically complex situations, Hypothesis 3 might fail. For example, the Richtmyer–Meshkov (RM) instability develops very long and thin structures at the tips of bubbles and spikes at late time; see Figure 7 for an illustration.

The narrow structures and approximate or actual bifurcations will degrade the algorithm. Excessive cell merging to ensure CFL stability will degrade accuracy, and in any case actual bifurcations are (presently) excluded. We require a robust algorithm to solve problems for which any of the above occurs. We propose that these situations

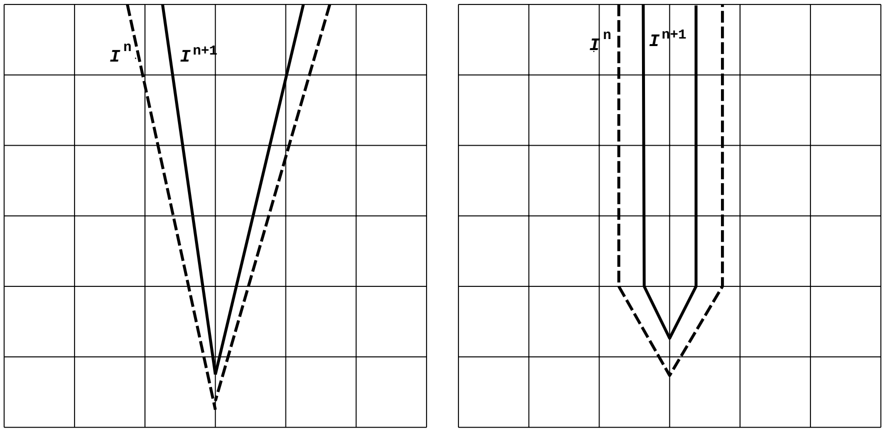

FIG. 7. *Limits on the merging process. $\mathcal{I}^n$ is represented by the dashed line and $\mathcal{I}^{n+1}$ by the solid line. At the time t^{n+1} level, in the first frame, the triangular cell at the tip is adjacent to a triangular cell and quadrilateral only; all of these cells form partial hexahedra having top area smaller than $\frac{1}{2}\Delta x^2$ and thus require further merger. In the second frame, the two branches of the curve near the tip of $\mathcal{I}^n$ and $\mathcal{I}^{n+1}$ are close and parallel to each other (forming a thin wall), thus forming a set of neighboring polyhedra with top area smaller than $\frac{1}{2}\Delta x^2$.*

Table 1
Comparison of error analysis for the test problem (22) *for Burgers' equation.*

Method	N	L_1 error	L_1 order	$\Sigma U_i \Delta x_i$
Untracked	30	6.83e-2	-	1.732
	60	3.49e-2	0.969	1.733
	120	1.63e-2	1.10	1.733
	240	8.24e-3	0.984	1.733
Nonconservatively tracked	30	2.80e-2	-	1.721
	60	6.89e-3	2.02	1.716
	120	4.23e-3	0.704	1.742
	240	2.01e-3	1.07	1.741
Conservatively tracked	30	2.17e-2	-	1.732
	60	7.07e-3	1.62	1.733
	120	2.11e-3	1.74	1.733
	240	6.04e-4	1.80	1.733

should be resolved by locally nonconservative tracking using the ghost cell algorithm of the authors [12]. Since these events will often occur on a lower dimensional space-time manifold, they will not impact the formal order of accuracy of the algorithm.

4. Numerical examples. In this section we present numerical examples showing the convergence and conservation properties of the conservative front tracking scheme.

4.1. Burgers' equation. Consider Burgers' equation $\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(\frac{1}{2}u^2) = 0$ on $[0,6] \times [0,T]$, with initial conditions

$$u(x,0) = \begin{cases} 0.2 * (x-1)^2 + 0.2, & x \in [1,3], \\ 0.2, & \text{elsewhere}. \end{cases} \tag{22}$$

In Table 1 we present numerical results at ($T = 3.2$) using three different methods: the untracked MUSCL scheme, the nonconservatively (shock) tracked scheme with an MUSCL interior solver, and the conservatively (shock) tracked scheme with an MUSCL interior solver. Here the column labeled L_1 error indicates the L_1 norm of $u - \tilde{U}$, where u is the exact solution and $\tilde{U}$ is the second order approximate solution reconstructed from the piecewise constant numerical solution U at time T. Figure 8 displays the comparison of the numerical results obtained with $N = 30$ cells. For all of section 4, the CFL number is equal to 0.4.

4.2. 1D Euler equations. Next we conduct a convergence test for the 1D Euler equations for a gamma law gas, $\gamma = 1.4$. We consider a tracked shock wave interacting with C^∞ data (a rarefaction wave with smooth edges). The computational domain is $[0,4]$ with flow-through boundary conditions. At time $T = 0$ there is a right facing rarefaction wave in $(1,2)$ and a left moving shock at $x = 3$. The left facing shock interacts with the rarefaction wave by the final time $T = 1$. We first define the initial states V_0 as follows: on $[0,1]$, the density, pressure, and velocity are $2.0, 0.5$, and -1.0, respectively. On $[1,2]$, V_0 has a centered rarefaction wave, ending at a pressure 1.5. On $[2,3]$, the state is constant. On $[3,4]$, the velocity is -1.5. Since the first

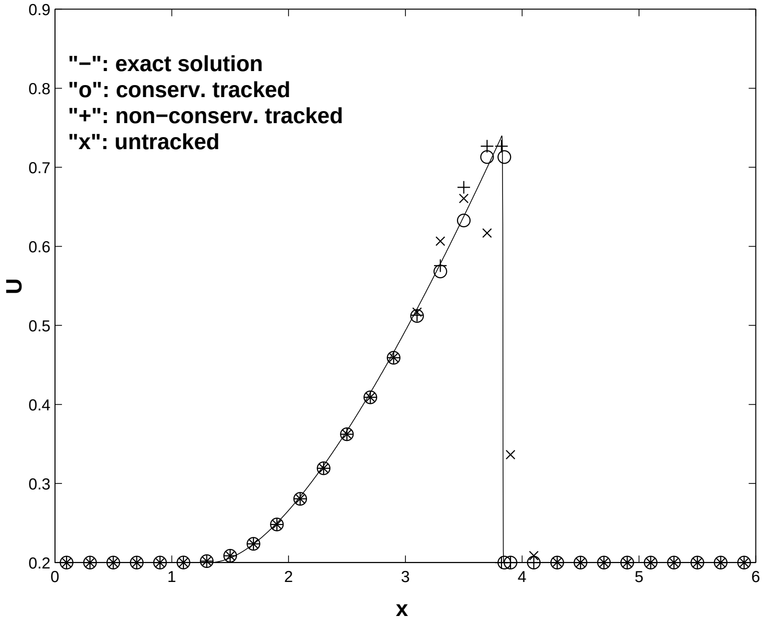

FIG. 8. *Comparison of numerical results for Burgers' equation.*

TABLE 2
Comparison of L_1 errors.

Nonconserv. tracked			Conserv. tracked	
N	L_1 error	L_1 order	L_1 error	L_1 order
100	0.0373	-	0.0395	-
200	0.0135	1.47	0.0106	1.90
400	0.00649	1.06	0.00361	1.55
800	0.00290	1.16	0.000891	2.02
1600	0.00148	0.970	0.000245	1.86
3200	0.000761	0.960	0.0000615	1.99

derivatives of V_0 have jumps at the rarefaction wave edges, we smooth the initial data V_0 so that

$$U_0(x) := \begin{cases} V_0(1)(2-\beta(x)) + V_0(2)(\beta(x)-1), & x \in (1,2), \\ V_0(x), & \text{elsewhere,} \end{cases}$$

where $\beta(x) = \frac{1}{2}(\sin\pi(x-\frac{3}{2})+3)$. We conduct the convergence test with the smoothed initial states U_0. The interior scheme is the second order MUSCL scheme with the shock wave tracked conservatively in one case and nonconservatively in the other. It is compared with a very fine ($N = 12800$, conservatively tracked) numerical solution to calculate the error in the L_1 norm. The comparison of the L_1 errors is shown in Table 2; the shock position errors $\sigma_e - \sigma_n$ are compared in Table 3, where σ_e

Table 3
Comparison of shock position errors.

Nonconserv. tracked			Conserv. tracked	
N	$\sigma_e - \sigma$	order	$\sigma_e - \sigma$	order
100	-4.20e-6	-	2.90e-4	-
200	-4.58e-6	-	9.15e-5	1.66
400	-2.54e-5	-	1.71e-5	2.42
800	-2.29e-5	-	2.85e-6	2.58
1600	-1.12e-5	1.03	9.70e-6	1.55
3200	-5.38e-6	1.06	2.00e-7	2.28

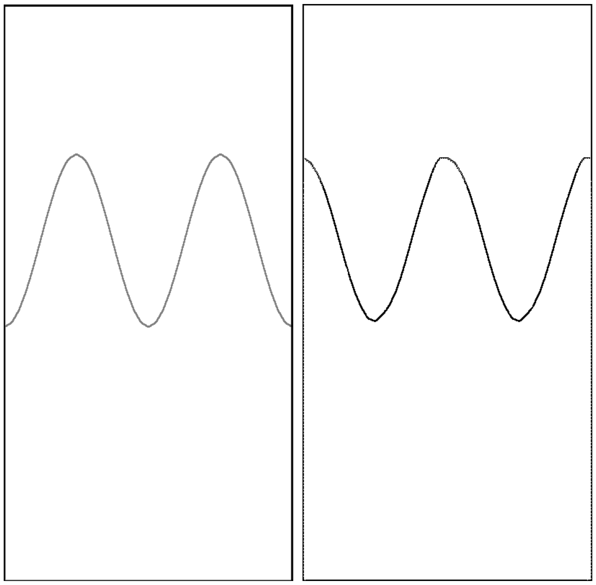

Fig. 9. *Front plot for the simulation of a horizontally moving contact discontinuity. The first frame displays the initial position of the contact; the second displays it after moving horizontally one quarter domain width in* 169 *time steps.*

denotes the exact shock position and σ_n denotes the numerical shock position. The conservatively tracked scheme is second order accurate.

4.3. 2D advection. We conduct a horizontal advection conservation test for the Euler equations to compare the fully conservative tracking scheme to the nonconservative tracking scheme. The numerical experiments were performed on a rectangular 1×2 domain with a 40×80 grid, displacing the interface horizontally half the domain width in 337 time steps. For the lower and upper boundaries of the domain, we use flow-through boundary conditions on which the states are normally extrapolated from the interior, and periodic conditions on the side boundaries. We use a polytropic gas, with polytropic exponent $\gamma = 1.4$. The contact discontinuity separating distinct gas states is tracked. The interface is sinusoidally perturbed with frequency 2.0 and amplitude 0.3. The initial configuration and the one-quarter width displaced configuration are shown in Figure 9. Excellent preservation of the sine wave is evident.

Table 4
Conservation error.

Conservation Error		
	Conservative tracking	Nonconservative tracking
Mass	0.0	0.21%
x-mom	0.0	0.21%
Energy	0.0	0.21%

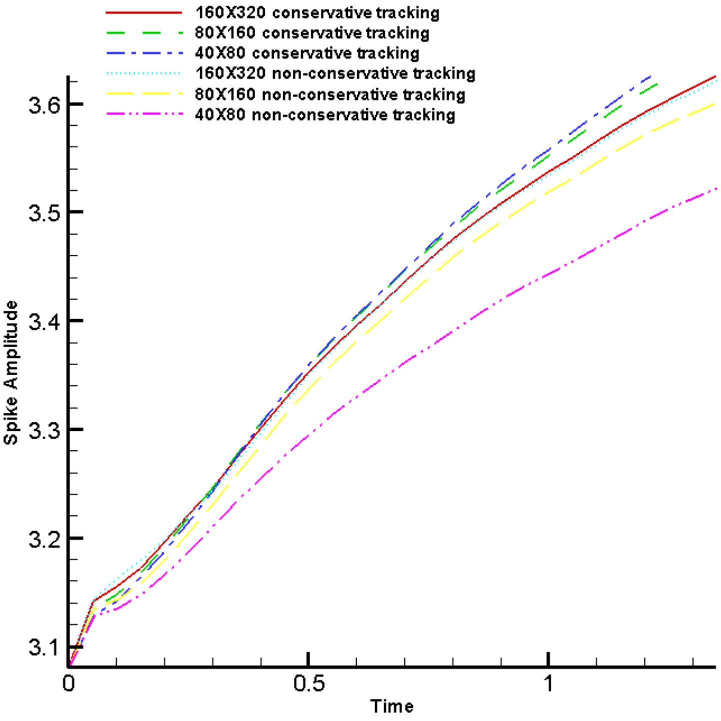

Fig. 10. *Spike amplitude in the RM instability simulations, as functions of time. The conservative tracked amplitude for a coarse grid is in approximate agreement with the nonconservative tracked amplitude for a fine grid.*

In Table 4, we compare the total conservation for the two methods, which is defined for the mass as

$$(\text{final mass} - \text{initial mass} + \text{boundary mass flux})/(\text{initial mass}), \tag{23}$$

with similar definitions for other conserved quantities. The conservative quantities refer to the lower gas in Figure 9. The total mass, momentum, and energy in the computational domain for the conservative tracking scheme show essentially perfect conservation, while the nonconserved tracking shows conservation errors of 0.21%.

4.4. Richtmyer–Meshkov instability. A Richtmyer–Meshkov (RM) instability is generated when a shock wave refracts through a perturbed interface which

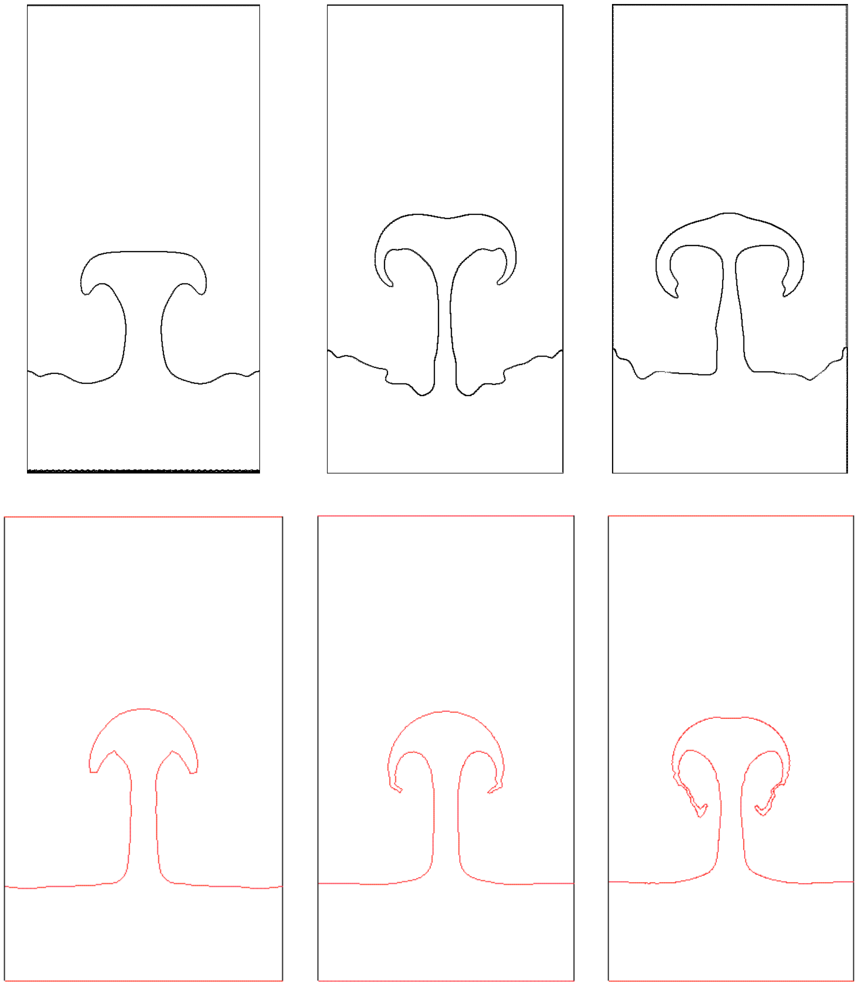

FIG. 11. *Front plot for the RM instability simulations. The upper row shows the plots of nonconservatively tracked interface at time* $= 1.38$. *The lower row shows the plots of conservatively tracked interface at the same time. For both rows, from left to right, are* 40×80, 80×160 *and* 160×320 *grids, respectively.*

separates fluids of differing densities. We compare simulations produced by the conservative and the nonconservative tracking schemes.

The numerical experiments were performed on a rectangular 1×2 domain, with a 40×80 grid, the lower and upper boundaries with flow-through boundary conditions, and periodic conditions for the side boundaries.

The initial configuration consists of a Mach 5.0 shock in a polytropic gas (with unshocked density 1.0) striking an interface separating two polytropic gases (both have polytropic exponent $\gamma = 1.40$). The preshock contact density ratio is $1 : 5$. The interface is sinusoidally perturbed with wavelength 1.0 and amplitude 0.1. Figure 11 shows the interface evolution of the RM instability; the initial configuration is shown as the left column. We also performed refined nonconservatively tracked simulations with 80×160 and 160×320 grids. The results indicate the convergence of the growth rate with nonconservative simulation to that of the conservative simulation when the computational mesh of the nonconservative simulation is refined. The 40×80 conservatively tracked solution appears to be comparable to both the finest (160×320) nonconservatively and conservatively tracked solutions, while the nonconservatively coarse grid run (40×80) tends to have a smaller growth rate. See Figures 10 and 11.

5. Conclusions.

We have proposed a new fully conservative front tracking algorithm. The algorithm is derived from an integral formulation of the PDEs. The 1D version of the algorithm is fully second order accurate away from the intersection of tracked waves. This has been determined by both the formal derivation and numerical experiments. In two dimensions, we provided the formal derivation that the scheme should be second order in the interior region and first order near the front. The convergence of bubble growth rate in the simulation of the RM instability seems to support this claim. Numerical tests in one and two dimensions demonstrate the improved conservation and convergence properties of the algorithm. The stability of the algorithm is verified by numerical experiments.

Conservative tracking is fundamentally an exercise in computational geometry to define the space-time interface. The finite volume differencing defined by the geometry follows standard algorithms. Further study of the space-time interface construction is called for. A robust algorithm may include nonconservative tracking for regions of greater local complexity than the conservative space-time interface construction will support.

REFERENCES

[1] R. Abgrall, *On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation*, J. Comput. Phys., 114 (1994), pp. 45–58.

[2] I.-L. Chern and P. Colella, *A Conservative Front Tracking Method for Hyperbolic Conservation Laws*, LLNL report UCRL-97200, Lawrence Livermore National Laboratory, Livermore, CA, 1987.

[3] I.-L. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv, *Front tracking for gas dynamics*, J. Comput. Phys., 62 (1986), pp. 83–110.

[4] J. Glimm, J. W. Grove, X.-L. Li, W. Oh, and D. H. Sharp, *A critical analysis of Rayleigh-Taylor growth rates*, J. Comput. Phys., 169 (2001), pp. 652–677.

[5] J. Glimm, J. W. Grove, X.-L. Li, K.-M. Shyue, Y. Zeng, and Q. Zhang, *Three-dimensional front tracking*, SIAM J. Sci. Comput., 19 (1998), pp. 703–727.

[6] J. Glimm, J. W. Grove, X.-L. Li, and D. C. Tan, *Robust computational algorithms for dynamic interface tracking in three dimensions*, SIAM J. Sci. Comput., 21 (2000), pp. 2240–2256.

[7] J. Glimm, J. W. Grove, X.-L. Li, and N. Zhao, *Simple front tracking*, in Nonlinear Partial Differential Equations, Contemp. Math. 238, G.-Q. Chen and E. DiBenedetto, eds., AMS, Providence, RI, 1999, pp. 133–149.

[8] J. Glimm, X.-L. Li, and Y.-J. Liu, *Conservative front tracking in higher space dimensions*, in Proceedings of International Workshop on Computational Methods for Continuum Physics and Their Applications (IWCCPA), Nanjing, China, Transactions of Nanjing University of Aeronautics and Astronautics, 18 (2001), suppl. 1–15.

[9] J. Glimm, X.-L. Li, and Y.-J. Liu, *Conservative front tracking in one space dimension*, in Fluid Flow and Transport in Porous Media: Mathematical and Numerical Treatment, Contemp. Math. 295, AMS, Providence, RI, 2002, pp. 253–264.

[10] J. Glimm, X.-L. Li, Y.-J. Liu, and N Zhao, *Conservative front tracking and level set algorithms*, Proc. Natl. Acad. Sci. USA, 98 (2001), pp. 14198–14201.

[11] J. Glimm, D. Marchesin, and O. McBryan, *Subgrid resolution of fluid discontinuities* II, J. Comput. Phys., 37 (1980), pp. 336–354.

[12] J. Glimm, D. Marchesin, and O. McBryan, *A numerical method for two phase flow with an unstable interface*, J. Comput. Phys., 39 (1981), pp. 179–200.

[13] J. Glimm and O. McBryan, *A computational model for interfaces*, Adv. in Appl. Math., 6 (1985), pp. 422–435.

[14] A. Harten and J. Hyman, *Self-adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws*, Report LA-9105, Los Alamos National Laboratory, Los Alamos, NM, 1981.

[15] W. E. Lorensen and H. E. Cline, *Marching cubes: A high resolution* 3*D surface construction algorithm*, Computer Graphics, 21 (1987), pp. 163–169.

[16] R. Pember, J. Bell, P. Colella, W. Cruchfield, and M. Welcome, *An adaptive Cartesian grid method for unsteady compressible flow in irregular regions*, J. Comput. Phys., 120 (1995), pp. 278–304.

[17] J. Smoller, *Shock Waves and Reaction-Diffusion Equations*, 2nd ed., Springer-Verlag, New York, 1994.

[18] B. Swartz and B. Wendroff, *Aztec: A front tracking code based on Godunov's method*, Appl. Numer. Math., 2 (1986), pp. 385–397.