Contemporary Mathematics
Volume **238**, 1999
B 0-8218-1196-7-03544-1

Simple Front Tracking

James Glimm, John W. Grove, X. L. Li, and N. Zhao

Abstract. A new and simplified front tracking algorithm has been developed as an aspect of the extension of this algorithm to three dimensions. Here we emphasize two main results: (1) a simplified description of the microtopology of the interface, based on interface crossings with cell block edges, and (2) an improved algorithm for the interaction of a tracked contact discontinuity with an untracked shock wave. For the latter question, we focus on the post interaction jump at the contact, which is a purely 1D issue. Comparisons to other methods, including the level set method, are included.

1. Introduction

Fluid interface instabilities and chaotic multiphase mixing define the microphysics of multiphase flow. Some of the best computations of these instabilities, in the regime of strongly accelerated flows, have been obtained using the front tracking method [**17, 3, 36**]. Front tracking employs a fundamental approach to the numerical modeling of a fluid interface through the use of a numerically defined interface which plays an explicit role in the algorithm. At the opposite extreme, numerical algorithms based on modern finite differences alone and a very fine grid, made feasible by adaptive mesh refinement (AMR), have also produced high quality simulations in these flow regimes [**21**]. The problem of fluid instabilities remains difficult numerically, especially when the practical requirements of multiscale simulations in three dimensions with flow fields not aligned with a rectangular grid are considered.

Section 2 is a review of the essential features that distinguish front tracking from other numerical methods, while section 3 explains simplifications and improvements recently introduced into this method. An illustrative three dimensional simulation shows a dynamic example of interface bifurcation. Section 4 compares the front tracking and the level set method Osher, Sethian and others [**33, 34, 31**]. In

1991 *Mathematics Subject Classification.* Primary: 76N15, 76T05, 76M20.

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 DAAG559810313 and the National Science Foundation, grant DMS-9732876.

Supported by the U.S. Department of Energy.

Supported by the Applied Mathematics Subprogram of the U.S. Department of Energy DE-FG02-90ER25084.

Section 5 we present numerical results for the interaction of a shock wave with a contact discontinuity.

2. The Front Tracking Algorithm

2.1. Modularity and Data Structures. Front tracking, as implemented in the code *FronTier*, makes extensive use of modern programming concepts, including data structures and modular organization. We indicate the use of data structures for description of the interface, as illustration. For more details, see [**12**].

The interface library describes the geometry and topology of piecewise smooth manifolds with piecewise smooth boundaries, embedded in R^3. Boundary and coboundary operators, to map from a manifold to its boundary, and to the manifolds which it bounds, are included in this library. The library forms a base class for the remainder *FronTier*. We begin with a description of the main data structures (whose names are in capital letters) and their interrelationships. At a continuum level, an INTERFACE [**14**] is a collection of non-intersecting geometric objects, NODEs, CURVEs, and SURFACEs, that correspond to zero, one, or two dimensional manifolds respectively. Both CURVEs and SURFACEs are oriented manifolds. NODEs correspond to boundary points of CURVEs, while CURVEs correspond to the boundaries of SURFACEs. We designate as COMPONENT, some labeling scheme, *i.e.* equivalence class, for the connected components in R^3 produced by the SURFACEs. Several connected components may share a common component label and constitute a single COMPONENT.

The discretized version of the INTERFACE has the same structure, with a piecewise linear description built from simplices of the appropriate dimensions. The CURVEs are composed of BONDs. Each BOND is a pair of POINTs, and (conceptually) the straight line segment joining them. SURFACEs are discretized in terms of TRIANGLEs.

In Figure 1, we illustrate the geometric data structures used to represent three dimensional interfaces in *FronTier*.

2.2. The Time Step Algorithm. The solution of systems of conservation laws, of the form

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{G}(\mathbf{U})$$

are supported by the framework discussed here.

2.2.1. *Interior States: Codimension 0.* The propagation in time of interior states uses a one dimensional regular grid stencil, for a sweep along each coordinate direction, and a choice of finite difference operators for this stencil, such as the higher order Godunov method, the Lax-Wendroff scheme, *etc.* Special care is needed only when the stencil is cut by a front; in this case there are missing state values, as the finite difference operator is expected to receive states from a single component only. In this sense, the method takes the idea of weak derivatives seriously, and will never compute a finite difference across a tracked front.

The missing points of the 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 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 [**13**]. With the ghost states thus defined, the interior solver follows a conventional finite difference algorithm. The

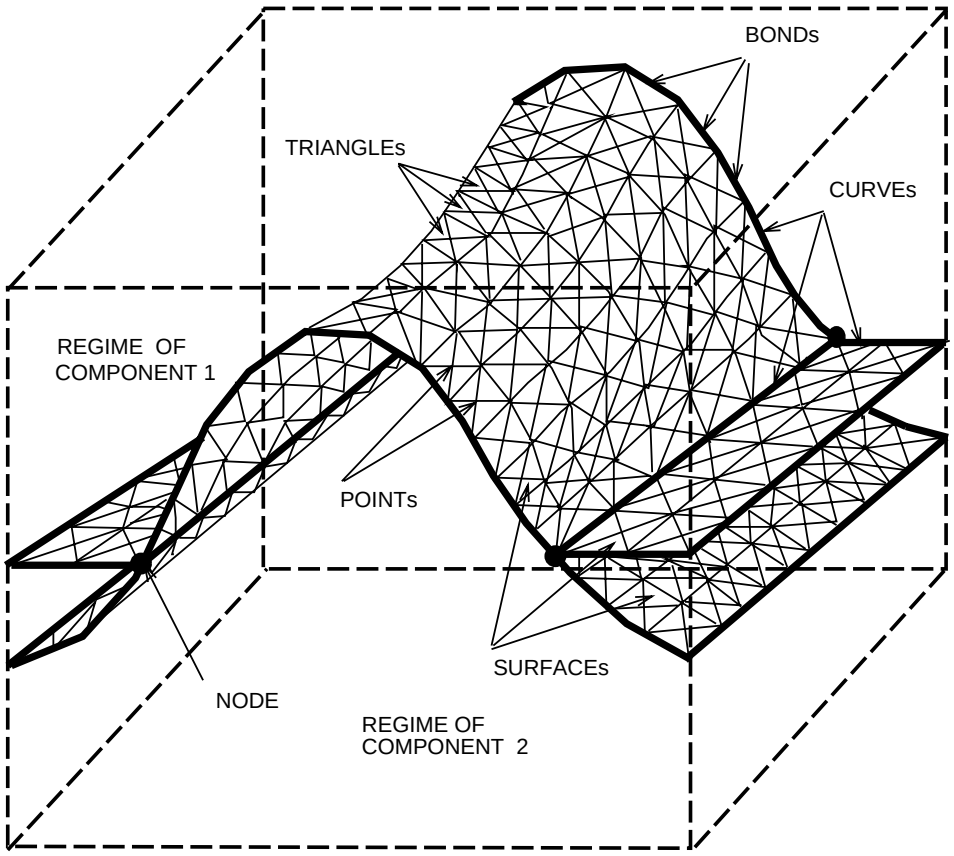

FIGURE 1. An illustration of the three dimensional geometric data structures used in *FronTier*.

stencil for a state on the right side of the interface is composed solely of right component states, and similarly for the stencil for a state on the left side of the interface.

2.2.2. *Regular Front States: Codimension 1.* The propagation of front states and positions is performed in a single step. Operator splitting, in a rotated coordinate system, allows separate propagation steps in directions normal to and tangent to the front. First consider the normal propagation step. The analysis reduces to the integration of a differential equation in one space dimension (the normal direction), and thus is largely independent of spatial dimension.

The leading order term in the propagation of a discontinuity, in the direction normal to the front, is given by the solution of a Riemann problem. This is a one dimensional Cauchy problem, with idealized initial conditions consisting of a single jump discontinuity. The solution will, in general, contain a number of waves. Of these, one is identified with the discontinuity being tracked. The Riemann solution gives the wave speed and states immediately ahead of and behind the advancing front. This speed defines the new interface position, and the states the updated flow states at the propagated points, and thus the lowest order version of the normal propagation algorithm.

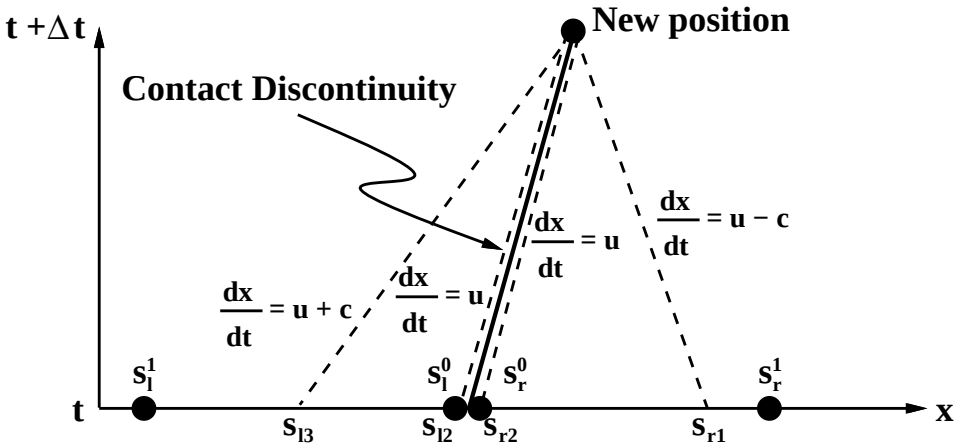

FIGURE 2. A schematic picture of the data used for the normal propagation of a contact discontinuity. The front data at the old time step provides a Riemann solution, that is corrected by interior data, using the method of characteristics.

Corrections are needed, to couple the interior variation of the solution states to the front propagation. For this purpose, a *generalized* Riemann problem is solved. By this we mean a Cauchy problem having a single jump discontinuity in the initial data. However, the initial data on each side of the jump discontinuity, rather than being constant, is now allowed a separate variation, linear in the distance from the front point, on each side of the front. The linear approximation to the nearby interior solution states is constructed by moving in mesh increments Δs away from the point being propagated in the direction of the normal on either side of the interface. The resulting points for solution evaluation, called s_l^i on the left side of the front in Figure 2 and s_r^i on the right, are not, in general, regular grid points, *i.e.*, a cell center corresponding to the rectangular lattice on which the interior states are distributed. The solution at such points must thus be constructed by interpolation using the regular grid points (cell centers) and the front states, but using only states from the same component, *i.e.*, coming from the same side of the interface. The states s_{l3}, s_{l2}, s_{r2}, and s_{r1} in Figure 2 at the feet of the backward characteristics are in turn interpolated from the states s_l^i and s_r^i. The solution of the nonlocal Riemann problem is constructed as a finite difference correction to the previously discussed (local) Riemann problem, using the method of characteristics. Because of the use of finite differences to solve the characteristic equations, the algorithm as specified here (see [**4**]) is suitable for simulations in which the flow variation on either side of the tracked fronts is small relative to the mesh width Δs. One purpose of the present paper is to remove this restriction, so that tracking of (strong) shock waves is not required, and thus so that the complexity of wave interactions solved in three dimensions is reduced. Since a modification of the normal sweep is proposed in §3.2 to allow for strong untracked shock waves interacting with a tracked contact discontinuity, further details are omitted here.

Curvature dependent corrections to the normal propagation are contained implicitly in the tangential sweep. The tangential propagation step modifies the interface states but not the points. The tangential motion of the interface is a reparameterization of the interface, and does not contribute to its dynamics. Thus the tangential motion of the interface is arbitrary, and as a convention, the reparameterization is taken to be the identity.

Separate finite difference steps are carried out for the states on each side of the interface. The splitting into normal and tangential directions is locally orthogonal, and for this reason no explicit source terms are introduced into the difference equations by the splitting. While this seems paradoxical, since *e.g.* radial expanding flow must decrease as the wave front expands, the decay mechanism is found not in an explicit source term, but in the divergence of the velocity field, as seen by the tangential finite difference stencil, after the states are projected onto the tangent plane to the surface.

2.3. Summary. Front tracking offers sharp resolution of fronts, interfaces, or other solution discontinuities. In contrast to other algorithms with features of this nature:

- Front tracking uses Riemann solvers to enforce jump discontinuities and proper jump relations in solution variables.
- Front tracking applies finite difference operations only to states on the same side of the front.
- The front tracking algorithm is applicable to complex physics, in principle to arbitrary systems of conservation laws.

Differences specific to the comparison to the level set method will be noted in §4.

3. Simple Front Tracking

3.1. A Simplified Geometrical Description of Fronts. Domain decomposition parallel computing requires the decomposition and reassembly of interface fragments during a communication phase of the time step operator. The need for robustness of the reassembly places a premium on operations that are purely local to individual processors. In reference [**11**] we described a new method to resolve the changing topology of the tracked fronts by the reconstruction of the interface based on the micro-topology within each rectangular grid block. In this method, the reconstructed interface is uniquely defined by its intersections with cell block edges.

The algorithm as currently implemented assumes a two fluid model. Suppose that a region is occupied by two immiscible fluids. For simplicity let us label the fluids by colors, say black and white. We wish to reconstruct the material interface separating the two fluids using some subset of the geometrical information associated with the "true" material interface. This subset will consist of the crossings of the original interface with the cell edges associated with a specified three dimensional lattice superimposed over the given region.

The reconstruction is based on the following three hypotheses:

1. At most two fluid components intersect any individual cell in the computational lattice.
2. Each cell edge has at most one interface crossing.

3. The cell corners and edges that lie on the same side of the reconstructed interface form a connected set.

The first hypothesis is clearly true for the two fluid flows considered here. More generally the basic algorithm is still valid for multi-fluid flows provided this condition holds locally in each computational cell. The second hypothesis says that the reconstructed interface will cross each cell edge at most once, while the third implies that if two corners of a cell lie in the same fluid component, then the entire edge connecting those corners also remains in the same fluid. These latter two assumptions rule out oscillations in the reconstructed interface at length scales below that of the lattice grid size.

The reconstruction algorithm is based on two steps. The topological reconstruction of an interface segment internal to a single grid block, and the continuity of the interface between adjacent grid blocks. The latter condition is enforced by the obvious geometric observation that interface crossings along specified lattice edge are the same for all cells that contain that edge.

The topological reconstruction of an interface inside a cell is based on the elementary reconstruction of the interface using the component labels (colors) for the cell corners. In general, given two colors, there are $2^8 = 256$ different possible colorings of the vertices of the graph of a cube. We can subdivide this set of colorings into isomorphy classes based on the subgraphs of the cube generated by vertices of the same color. Since we can always remap the cube by transposing the vertex colors (relabeling white to black and black to white), we need only consider the cases where the black vertices appear four or fewer times.

There are fourteen distinct cases.

1. No black vertices.
2. One black vertex.
3. Two black vertices subdivided into three cases:
   (a) Two connected black vertices.
   (b) Two disconnected black vertices sharing a common face.
   (c) Two disconnected black vertices with no common face.
4. Three black vertices subdivided into three cases:
   (a) Three connected black vertices.
   (b) Two connected and one disconnected black vertices.
   (c) Three disconnected black vertices.
5. Four black vertices subdivided into six cases:
   (a) Four connected black vertices sharing a common face of the cube.
   (b) Four connected black vertices whose subgraph has a single vertex of degree three.
   (c) Four connected black vertices whose subgraph has vertices of degree one or two (*i.e.*, the graph is a broken line).
   (d) Three connected and one disconnected black vertices.
   (e) Two disconnected black edges.
   (f) Four disconnected black vertices.

It is trivial to verify that any two coloring of the graph of a cube must correspond to one of these distinct cases, see Figure 3. A block interface is defined as an interface segment formed by connecting the grid line crossings that occur along edges where the coloring changes. For zero black vertices the corresponding interface is empty. In each of the remaining cases we form the interface by

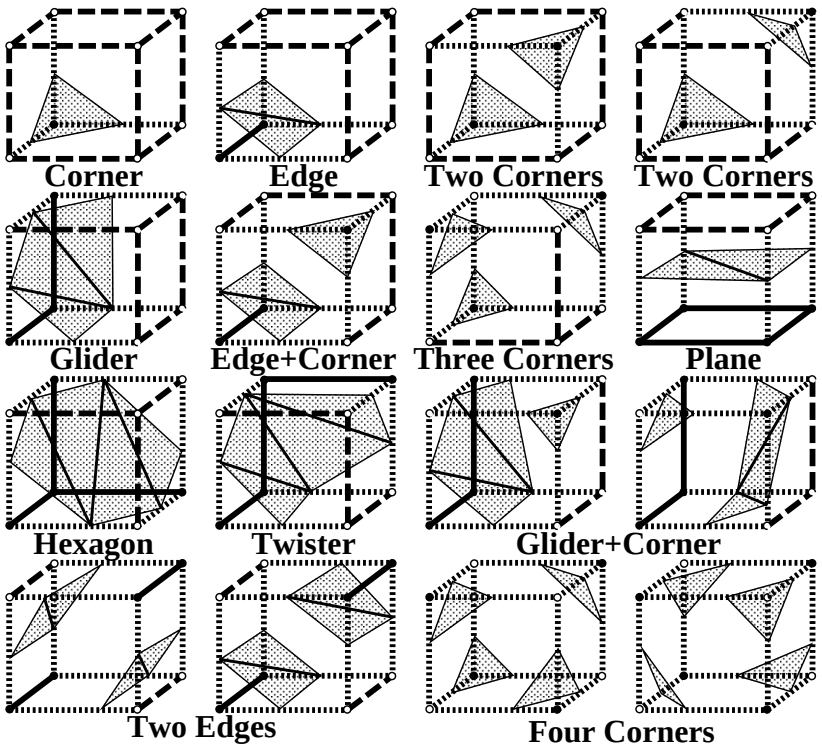

FIGURE 3. Grid based interface reconstruction. For a two-fluid interface within each grid cell, there are $2^8 = 256$ possible configurations for the crossings of the cell edge by the interface. Through elementary operations of rotation, reflection, and separation, these cases can be reduced to the sixteen cases shown above.

constructing a polygon whose sides consist of edge crossings with a common face. These cases are illustrated in Figure 3. We see there are six distinct topological surface elements (corner, edge, glider, plane, hexagon, and twister) and a total of sixteen (not counting the trivial no crossings case) cases in all that are composites of these six elementary surface elements. Note that the cases, glider+corner, two edges, and four corners do not uniquely determine the interface (unlike the previous cases). This is because there are two ways to select the interfaces, those separating out the black or those separating out the white vertices. For example in the glider+corner case we could construct the glider using the three connected black vertices or the three connected white vertices. At the level of this algorithm this choice is arbitrary since either construction will yield a globally consistent interface. In practice one might use additional topological information from the original (*i.e.*, unreconstructed) interface to select the choice that best fits that interface.

The power of the new interface description to untangle interfaces and to automate the dynamic pinchoff or change of interface topology is illustrated in Figure 4. In this figure, we show an instance of interface pinchoff and reconstruction using the grid based interface description in a three dimensional simulation. The numerical

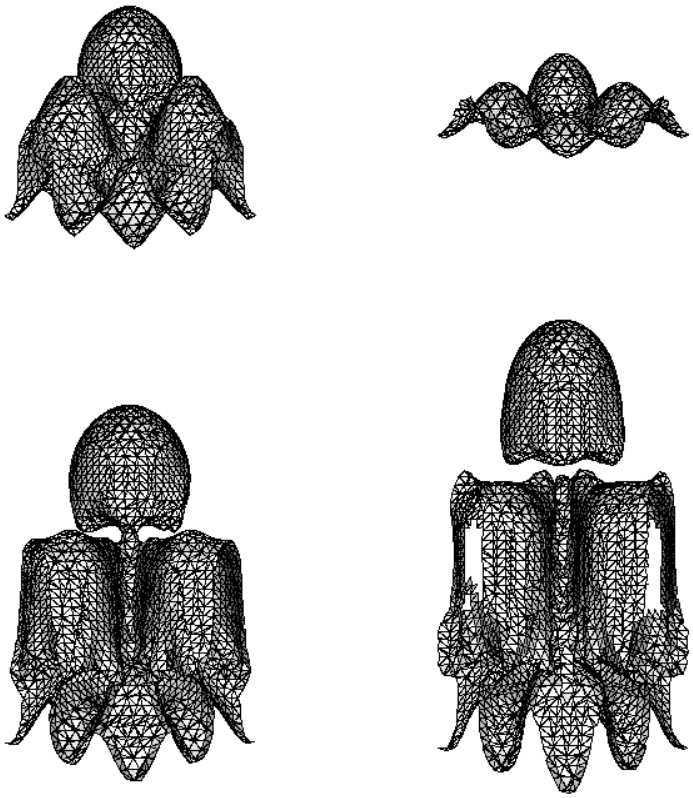

FIGURE 4. A simulation of Rayleigh-Taylor instability using *FronTier*. The initial configuration has four modes of unequal amplitude that evolve at different speeds. In the final frame the lead bubble has pinched off from the main body of light fluid. This bifurcation in the flow is an important part of the dynamics of this instability, and is handled correctly by *FronTier*.

handling of the pinch-off consists of three steps. First the fluid interface is propagated as described in §2. After the propagation, the interface may become tangled. This will be manifested by an inconsistency between the regular grid components and the components at the interface-grid intersections. We walk through the grid cell edges to check the consistency of the components. Unphysical interface-grid intersections, when detected, are removed, leaving only fully consistent interface crossings. Finally, we reconstruct the interface within each grid block as described above using the new crossings. Pinchoff and other topological bifurcations in the interface follow automatically from use of the grid based description with no additional programming complexity.

This grid based interface reconstruction algorithm is comparable in simplicity to the level set method as a geometric interface description.

3.2. Simplified Shock-Contact Interaction. The refraction of a shock wave by a material interface or contact discontinuity is an important physical interaction that has a surprisingly rich and complex structure. An illustration of the variety

of possible refraction behaviors can be seen in the shock tube experiments of Jahn, Henderson, and others, (see for example [**24, 6, 7, 8**]), and numerical computations [**2, 10, 15, 18, 16, 19**].

Basically, the one dimensional interaction of a shock wave with a material interface consists of the acceleration of the material interface due to the differential transfer of momentum from the shock to particles on either side of the interface, and the refraction of the shock front into a set of reflected and transmitted waves. The solution to this interaction is found by solving the Riemann problem with data given by the flow on the upstream side of the material interface and the flow behind the shock wave. It can be shown that the transmitted wave must always be a shock, while the reflected wave may be either a shock or rarefaction wave. If both fronts are tracked, then this Riemann problem data is explicit in the numerical representation of the flow, and the resolution of the interaction is essentially exact.

The situation for two dimensional oblique refractions is considerably more complex. If the angle of incidence between the two waves is not too large, then the local behavior of the interaction can be computed using shock polar analysis [**5**]. The refraction consists of a locally steady state flow with the upstream waves (*i.e.*, the incoming shock and material interface) providing Riemann data for the outgoing downstream waves (the transmitted and reflected waves and the accelerated material interface). In this case the refraction is said to be regular [**15**]. Simulations of locally (in space and time) regular refractions using front tracking have been made with considerable success in a variety of simulations [**15, 17, 23, 22**]. As in the one dimensional case, the application of this algorithm requires the tracking of both the incoming shock and the material interface so that the states about these waves can be used in the shock polar analysis. As the angle of incidence between the two waves is increased the refraction becomes unsteady in space and time and the shock polar equations have no solution. In some cases one can extrapolate the shock polar solutions beyond their strict domain of validity and still obtain reasonable estimates of the flow states [**20, 32, 35**], or else apply a node scattering analysis to interpret the wave behavior into the irregular refraction regime [**18**]. However the virtually infinite set of outcomes for irregular refractions places a practical limitation on the ability of explicit tracking to handle this interaction. In practice, this means that simulations of shock refractions, as occur say in Richtmyer-Meshkov instability, are limited to interactions where the initial perturbations of the material interface are not too large. Fortunately this has not proven to be too onerous a restriction for the simulations conducted previously. The situation in three dimensional flows is even more complex and the need to represent the dynamic local geometry of the interacting waves makes explicit tracking of both the shock front and the material interface virtually impossible. Thus there is a need to formulate versions of the tracking algorithms that do not require explicit tracking of both the incoming shock and material interface.

In general the material interface needs to be tracked to eliminate numerical diffusion across that wave front, so we are primarily interested in formulating a modification of the point propagate algorithm that allows for the interaction of a captured (*i.e.*, untracked) shock with a tracked material interface. The basic problem is this, the point propagate algorithm as described in §2 uses the method of characteristics to integrate the incoming waves into the material interface. In the event that a captured shock front is nearby the interface, the flow gradients near

that wave are not small, and the integration of the characteristic differential equations on either side of the interface produce substantial errors. Furthermore these errors produce spurious wave modes that move with the local fluid velocity. Since this is the same velocity as the material interface, these error waves tend to lock into interface and do not dissipate. In reality, errors of this type occur in all finite difference methods, and these method rely on the numerical viscosity inherent in their schemes to diffuse these error modes away from the fluid fronts, which in most cases are captured waves themselves. Paradoxically, it is precisely the attempt to control numerical diffusion at the material interface that reduces the numerical viscosity near the front and prevents these spurious wave modes from dissipating away from the interface. This problem is compounded since subsequent wave refractions through the material interface are affected by the previously established errors.

The modified point propagation algorithm is based on the analogy between one dimensional front tracking and Lagrangian hydrodynamics. Indeed in one space dimension front tracking can be interpreted as a locally Lagrangian update to the states at the fronts. In multiple space dimensions this same analogy applies to the normal propagation sweep to the interface points. As described in §2, the data for the normal sweep is obtained by sampling the flow state at points p_l^i, p_r^i distributed from the interface in equal spatial increments Δs along the normal direction to the point being propagated. We denote the states at these points by s_l^i and s_r^i for $0 \le i < N$ where N is the number of stencil points on either side of the interface. Thus s_l^0 and s_r^0 are the states at the unpropagated front, and the s_l^i and s_r^i are the states at locations a distance $i\Delta s$ on the left or right hand side of the interface respectively. Next, to each state and position in the propagation stencil we associate a corresponding slope to be used in interpolating the flow in a one dimensional interval centered at that point. Let us denote these slopes by δs_l^i and δs_r^i respectively. For $i = 0$ and $i = N - 1$ the interpolation intervals are of width $\frac{\Delta s}{2}$ while for $1 \le i < N-1$ the intervals are of length Δs. The original algorithm is included in this formulation if we choose $N = 2$ and compute the slopes so that the total interpolation profile between the two points on the same side of the interface is linear with endpoints given by the states at those points. The modification of the point propagate algorithm is to use the van Leer limiter to compute the slopes. For example on the right hand side we choose

$$\delta s_r^i = sgn \min(|\frac{s_r^{i+1} - s_r^{i-1}}{2}|, |s_r^{i+1} - s_r^i|, |s_r^i - s_r^{i-1}|) \tag{1}$$

where sgn is the common algebraic sign of the three differences if all agree in sign, zero otherwise. At the endpoints (in particular at the interface) we simply copy the slopes from the adjacent interior stencil point (*i.e.*, $\delta s_r^0 = \delta s_r^1$). In practice we choose $N = 3$ so that we only compute one slope for all three states, and this is the value centered about a point a distance of one Δs from the interface.

The basic algorithm then proceeds as before. We use the method of characteristics to trace back along the incoming sound waves to the interface. States at the feet of these characteristics are computed using the data states and computed slopes. If the flow gradient is not large and the flow profile is monotone this is a second order interpolation and agrees to this order with the original algorithm. In the case of a strong wave near the interface, the slopes are limited to zero, which provides the additional numerical dissipation needed to allow the entropy waves

to escape from the material interface where they can diffuse into the surrounding region.

4. Comparison of Front Tracking and Level Sets

4.1. Level Sets. We constructed a level set code [**26, 27, 28, 29**] and obtained the first level set simulations of accelerated fluid instabilities in three dimensions. The code was validated by comparison to previous simulations and to perturbative analytic solutions [**37, 38**]. Two changes in the level set algorithm were needed for successful validation, each improving upon the original algorithm [**30**]. Mass diffusion across the interface in the algorithm of [**30**] grows as $t^{1/2}$. This effect is undesirable, and was cured [**29**] with artificial compression. The strength of an artificial compression parameter controls the resulting narrowness of the diffused interfacial density discontinuity. Our point of view was only to control the long time growth of this diffused band, while not trying to limit it too sharply, say below about five mesh cells in width. When used in this mode, we found artificial compression to be satisfactory in performance. Other workers have reported growth of secondary instabilities along the interface resulting from the use of artificial compression. These secondary instabilities have been attributed to a larger artificial compression parameter, used in an attempt to gain a narrower diffused interface.

Inconsistency between the diffused mass and the sharp change in the equation of state (EOS) at the location of the level zero surface, $\phi = 0$, of the level set function ϕ is a second problem for [**30**]. Due to the gradual change in density and the sharp change in the equation of state, at most one additional thermodynamic quantity can have one sided continuity at the interface, and all the rest will generically have spikes, or standing waves located at the interface. As discovered and documented in [**12**], standing pressure waves at the interface result from differencing in conserved variables. The usual, if not totally satisfactory solution, to this problem is to use a mixed material EOS descriptive of atomically mixed fluids in varying proportions of mixture. In this case, the level set is totally decoupled from the flow and has the role of graphics post processing. We followed a special case of this approach, by restricting our studies to fluids with identical EOS, in which case the density discontinuity results from a jump in temperature.

A proposed cure to the level set EOS problem is to difference in nonconserved variables, such as pressure [**25**]. A more fundamental solution was recently proposed [**9**]. In cells that are near the front, in the sense of not having a full stencil of states on one side of it, ghost cell states are constructed to complete the stencil and to allow a standard finite difference operator update.

To begin the ghost cell state construction, entropy is extrapolated from the nearest state on the proper side of the front. From these extrapolated entropies and from the existing pressures at the grid locations, the equation of state reconstructs a density. The velocities are decomposed into components v_n and v_t, normal and tangential to the interface. Then v_n comes from the existing fluid grid value, while v_t is extrapolated. The extrapolated/real velocities, the pseudo densities, and the real pressures at the grid cell define a "ghost state" used to complete the stencil and to allow update of the near front states with insufficient data on their own side of the front.

This use of extrapolated values to complete missing stencil states for the update of near front states is similar to the treatment of near front states used for

many years in *FronTier* [**13**] and brings these two methods closer algorithmically. Here, the missing states are filled in completely by extrapolation, so that the stencil is composed of states taken from a single component. The states used for extrapolation are associated with the front position itself, rather than the cell centered states. Communication between the states on the two sides of the interface occurs through Riemann solvers based on these states located at the front.

4.2. Riemann Solvers for Front Propagation. Front tracking [**13, 4**] differs from ordinary finite differences through three features, that are supported through an algorithm applicable to a general system of conservation laws:

1. A data structure to support the definition of a sharp interface;
2. Special algorithms to compute updated finite differences for cells whose regular finite difference stencil crosses the front;
3. Riemann solvers to enforce correct propagation velocities and jump discontinuities at the front.

A fundamental difference between the front tracking method and the original formulation of the level set method [**30, 9**] is in the representation of the velocity of the moving front. Using a one dimensional formalism for simplicity, the Euler equations gives the motion of a front as a solution to the equation:

$$\frac{dx}{dt} = v_{\text{front}}, \tag{2}$$

where the front velocity v_{front} is determined by the flow state near the front and the Rankine-Hugoniot conditions across the front. Front tracking treats this equation literally and computes the discrete motion of the front in terms of the coupling between the front motion and the flow on either side of the front. One point of importance is that this method only requires that the velocity field be defined at the front and does not seek to extrapolate v_{front} into a space-time region adjacent to the front. A further property is that the discrete representation of v_{front} allows for the jump in the derivative of v_{front} across the front. In contrast, the level set method as formulated in [**30, 9**], first seeks an extrapolated front velocity field $v_{\text{grid}}(x,t)$ in a neighborhood of the front that agrees with v_{front} at the interface. It then seeks to update the interface position by integrating the equation:

$$\phi_t + v_{\text{grid}} \cdot \nabla\phi = 0 \tag{3}$$

where $\phi(x,t)$ is the level set function whose zero set $\phi = 0$ provides the space-time position of the front. Since this construction requires that v_{grid} be a smooth space-time field, a first order error is introduced at the front since the "true" interface velocity must in general have a jump in its derivative across the wave. A second source of truncation error occurs due to the practical requirement that v_{grid} be constructed at a single time level. This allows for the explicit integration of (3) but restricts the time accuracy of the discrete solution to first order in time.

This error is particularly significant when the flow field near the front is strongly nonlinear, as in the case of a shock refracting through a material interface. For a strong shock interacting with a contact discontinuity, the velocity gradient variation at the edge of the shock is $O(\Delta x^{-1})$. This truncation error in the level set propagation was identified by Adalsteinson and Sethian [**1**], who proposed to eliminate this error by using pure front velocities in the propagation of the level set. The level set function is constructed initially to be the distance to the interface. The given front velocity v_{front} is extended to a velocity v_{ext} defined in a band around the front,

so that the variation in the velocity v_{ext} occurs only on level sets of the level set function ϕ. Thus v_{ext} is constructed as a solution of the equation $\nabla v_{\text{ext}} \cdot \nabla \phi = 0$ at every time step. The front is advanced through the level set function ϕ using:

$$\phi_t + v_{\text{ext}} \cdot \nabla \phi = 0 \ . \tag{4}$$

Using (4), the entire set of level curves moves with rigid spacing under the extended front velocity v_{ext}, rather than with a spatially variable grid velocity v_{grid}. The velocity extension method makes no reference to v_{grid}, and it can thus be used in cases when only a front velocity is defined. This is the basis of Sethian's fast marching algorithm.

The truncation error of (4) appears to be identical to the front tracking propagation of (2). The method (4) of front propagation is algorithmically a close approximation to the front tracking use of a local velocity v_{front} defined at the front. These two algorithms utilize identical data, namely v_{front}. In front tracking, only the level set $\phi = 0$, *i.e.*, only the front itself is involved, and there is no need to construct an extended front velocity. From this point of view, front tracking can be thought of as an ultra narrow band level set method.

5. Numerical Experiments

5.1. One Dimensional Simulations. Numerical simulation shows that the L_∞ errors associated with the interaction of an captured strong shock and a tracked or level set contact are $O(1)$, and increase with shock strength. These errors are concentrated in a region of width $O(\Delta x)$ near the contact, so that the L_1 error is first order. These errors appear to have an origin similar to that of shock wall heating, which are common to many numerical methods.

For strong shocks, the L_1 density errors at the contact are an order of magnitude larger for the level set contacts than they are for tracked contacts. This difference appears to be due to the use of Riemann solvers in front tracking to describe nonlinear wave interactions as opposed to the finite differencing used in the level set method. In fact the level set density errors for strong shocks are comparable to those for the artificial compression method. See Table 1 and Figure 5. Point value density errors at the contact are given in Table 2.

For the numerical experiments, illustrated in Tables 1 and 2 and Figure 5, the fluid ahead of the incident shock was at rest, $v_a = 0$, and at an ambient pressure of $p_a = 1$. The ahead-shock densities to the left and right of the interface were $d_l = 1$ and $d_r = 5$ respectively. The computational domain is $[0, 6]$ and both fluids were taken as perfect gases with $\gamma = 1.4$. Simulations were conducted for various incident shock strengths as measured by the shock Mach number. Initially, the contact interface is located at $x = 3$. The shock travels to the left from the point $x = 3.5$. The computations used a total of 240 grid points for the spatial mesh. The numerical results are compared with the level set entropy extrapolation method of Fedkiw and Osher [**9**] and the artificial compression method [**29**].

5.2. Other Comparisons. A five way comparison (theory, experiment, and three simulation codes) was conducted [**21, 22**] for the single mode Richtmyer Meshkov instability of a shock refracting through a sinusoidally perturbed interface separating two fluids. The agreement was impressive. The three codes agreed in their computations of the instability growth rate and their solutions were within

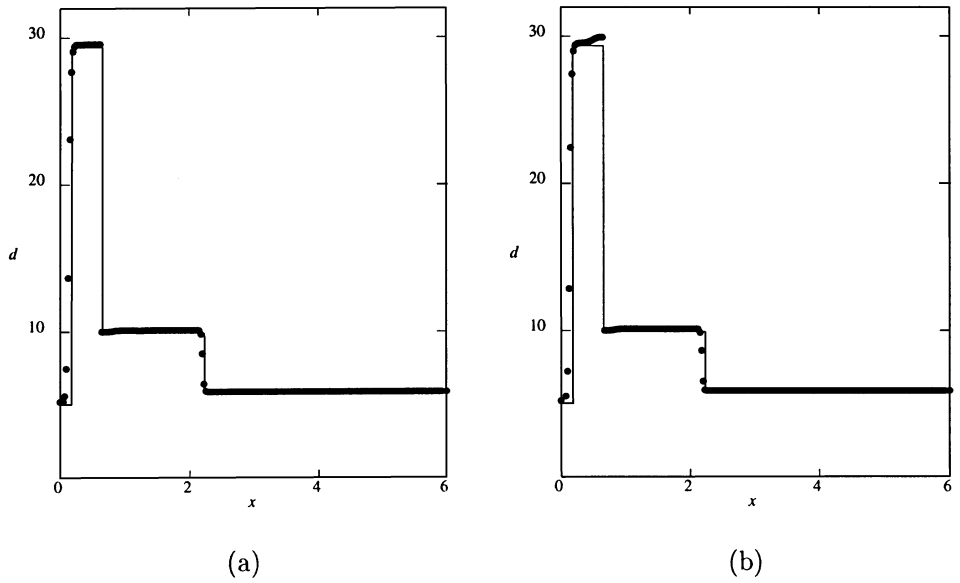

FIGURE 5. Density plot for the solution of an untracked Mach 10 shock interacting with a contact. The contact is tracked in the left frame and represented by a level set in the right. In each frame, the post interaction transmitted shock is on the left, the contact in the center, and the reflected shock wave is on the right. Also shown is the exact solution (solid line in each figure).

Mach Number	*FronTier*	Level Set	TVD/AC
2	9.0238×10^{-3}	1.7165×10^{-2}	7.3113×10^{-2}
5	1.3620×10^{-2}	1.7065×10^{-2}	3.1773×10^{-1}
10	1.8129×10^{-2}	4.9970×10^{-2}	3.7688×10^{-1}
50	1.9120×10^{-2}	2.9411×10^{-1}	4.1944×10^{-1}

TABLE 1. Comparison of L_1 Density Errors

Mach number	*FronTier* Error	Level Set Error
$M = 2$	-0.130781	-0.242735
$M = 5$	-0.063719	-0.027313
$M = 10$	-0.025553	0.430499
$M = 50$	0.071407	1.76907

TABLE 2. Comparison of Density Errors at Contact.

the experimental error bars as was the theory [**37, 38**], based on a low order perturbation expansion in powers of the interface amplitude, Padé resummation, and matched asymptotic expansions. *FronTier* and two higher order Godunov codes, the adaptive mesh refinement code, RAGE, and the PPM based code, PROMETHEUS, were compared. At the level of differences among the three computations, *FronTier*

and RAGE were quite similar while PROMETHEUS showed high frequency secondary instabilities at late time on the interface. The general opinion is that the high order interfacial oscillations shown by PROMETHEUS are numerical in origin and do not correspond to real surface instabilities. *FronTier* used a factor of two less resolution per linear dimension (a factor of eight in space-time mesh cells) than did RAGE, and a factor of three less than did PROMETHEUS (a factor of 27 in space-time mesh cells).

The authors have conducted a series of comparisons of an artificial compression code with *FronTier*, for both Rayleigh Taylor and Richtmyer Meshkov problems, with good results. Comparisons to theory [**37, 38**] were also included.

References

[1] D. Adalsteinson and J. A. Sethian. The fast construction of extension velocities in level set methods. *J. Comp. Phys.*, 1998. submitted.

[2] G. Ben-Dor and I. Glass. Domains and boundaries of non-stationary oblique shock-wave reflexions: 2. monatomic gas. *J. Fluid Mech.*, 96(4):735–756, 1980.

[3] Y. Chen, Y. Deng, J. Glimm, G. Li, D. H. Sharp, and Q. Zhang. A renormalization group scaling analysis for compressible two-phase flow. *Phys. Fluids A*, 5(11):2929–2937, 1993.

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

[5] R. Courant and K. Friedrichs. *Supersonic Flow and Shock Waves*. Springer-Verlag, New York, 1976.

[6] A. Abd el Fattah and L. Henderson. Shock waves at a fast-slow gas interface. *J. Fluid Mech.*, 86:15–32, 1978.

[7] A. Abd el Fattah and L. Henderson. Shock waves at a slow-fast gas interface. *J. Fluid Mech.*, 89:79–95, 1978.

[8] A. Abd el Fattah, L. Henderson, and A. Lozzi. Precursor shock waves at a slow-fast gas interface. *J. Fluid Mech.*, 76:157–176, 1976.

[9] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comp. Phys.*, 1998. Submitted.

[10] H. Glaz, P. Colella, I. I. Glass, and R. L. Deschambault. A numerical and experimental study of pseudostationary oblique shock wave reflections with experimental comparisons. *Proc. Royal Soc. London A*, 398:117–140, 1985.

[11] J. Glimm, J. Grove, X. L. Li, and D. C. Tan. Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comp.*, (SUNYSB-AMS-98-03). Submitted.

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

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

[14] J. Glimm and O. McBryan. A computational model for interfaces. *Adv. Appl. Math.*, 6:422–435, 1985.

[15] J. Grove. The interaction of shock waves with fluid interfaces. *Adv. Appl. Math.*, 10:201–227, 1989.

[16] J. Grove. Irregular shock refractions at a material interface. In S. Schmidt, R. Dick, J. Forbes, and D. Tasker, editors, *Shock Compression of Condensed Matter 1991*, pages 241–244. North-Holland, 1992.

[17] J. Grove, R. Holmes, D. H. Sharp, Y. Yang, and Q. Zhang. Quantitative theory of Richtmyer-Meshkov instability. *Phys. Rev. Lett.*, 71(21):3473–3476, 1993.

[18] J. Grove and R. Menikoff. The anomalous reflection of a shock wave at a material interface. *J. Fluid Mech.*, 219:313–336, 1990.

[19] J. W. Grove. Applications of front tracking to the simulation of shock refractions and unstable mixing. *J. Appl. Num. Math.*, 14:213–237, 1994.

[20] J. F. Hawley and N. Zabusky. Vortex paradigm for shock accelerated density stratified interfaces. *Phys. Rev. L.*, 63:1241, 1989.

[21] R. L. Holmes, G. Dimonte, B. Fryxell, M. Gittings, J. W. Grove, M. Schneider, D. H. Sharp, A. Velikovich, R. P. Weaver, and Q. Zhang. Single mode Richtmyer-Meshkov instability growth: Experiment, simulation, and theory. In G. Jourdan and L. Houas, editors, *Proceedings of the 6th International Workshop on the Physics of Compressible Turbulent Mixing*, pages 197–202. Imprimerie Caractère, Marseille, France, 1997.

[22] R. L. Holmes, B. Fryxell, M. Gittings, J. W. Grove, G. Dimonte, M. Schneider, D. H. Sharp, A. Velikovich, R. P. Weaver, and Q. Zhang. Richtmyer-meshkov instability growth: Experiment, simulation, and theory. Technical Report LA-UR-97-2606, Los Alamos National Laboratory, Los Alamos, NM, 1997. too appear in J. Fluid Mech.

[23] R. L. Holmes, J. W. Grove, and D. H. Sharp. Numerical investigation of Richtmyer-Meshkov instability using front tracking. *J. Fluid Mech.*, 301:51–64, 1995.

[24] R. G. Jahn. The refraction of shock waves at a gaseous interface. *J. Fluid Mech.*, 1:457–489, 1956.

[25] S. Karni. Hybrid multifluid algorithms. *SIAM J. Sci. Comput.*, 17:1019–1039, 1996.

[26] X.-L. Li and J. Glimm. A numerical study of Richtmyer-Meshkov instability in three dimensions. In H. Kubota and S. Aso, editors, *Proceedings of the Second Asia CFD Conference, Tokyo*, volume 1, pages 87–92. Japan Society of Computational Fluid Dynamics, 1996.

[27] X.-L. Li, J. W. Grove, and Q. Zhang. Parallel computation of three dimensional Rayleigh-Taylor instability in compressible fluids through the front tracking method and level set methods. In *Proceedings of the 4th International Workshop on the Physics of Compressible Turbulent Mixing*. Cambridge University Press, Cambridge, 1993.

[28] X. L. Li and B. X. Jin. Parallel computation of Rayleigh-Taylor instability through high resolution scheme for contact discontinuity. In *Proceedings of the First Asia Conference on Computational Fluid Dynamics*, volume 2, pages 811–817. HKUST Press, 1995.

[29] X.-L. Li, B. X. Jin, and J. Glimm. Numerical study for the three dimensional Rayleigh-Taylor instability using the TVD/AC scheme and parallel computation. *J. Comp. Phys.*, 126:343–355, 1996.

[30] R. Mulder, S. Osher, and J. A. Sethian. Computing interface motion in compressible gas dynamics. *J. Comp. Phys.*, 100:209–228, 1992.

[31] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi equations. *Jour. Comp. Phys*, 79:12–49, 1988.

[32] R. Samtaney and N. J. Zabusky. Circulation deposition on shock-accelerated planar and curved density-stratified interfaces: Models and scaling laws. *J. Fluid. Mech.*, 269:45–78, 1994.

[33] J. A. Sethian. Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws. *J. Differential Geometry*, 31:131–161, 1990.

[34] J. A. Sethian. *Level Set Methods*. Cambridge University Press, 1996.

[35] X. L. Yang, I-L. Chern, N. J. Zabusky, R. Samtaney, and J. F. Hawley. Vorticity generation and evolution in shock-accelerated density-stratified interfaces. *Phys. Fluids A*, 4(7):1531–1540, 1992.

[36] Q. Zhang and M. J. Graham. A numerical study of Richtmyer-Meshkov instability driven by cylindrical shocks. *Phys. Fluids*, 10:974–992, 1998.

[37] Q. Zhang and S. Sohn. Nonlinear theory of unstable fluid mixing driven by shock waves. *Phys. Fluids*, 9:1106–1124, 1997.

[38] Q. Zhang and S. Sohn. Quantitative theory of Richtmyer-Meshkov instability in three dimensions. *ZAMP*, 1998. To appear.

Department of Applied Mathematics and Statistics, University at Stony Brook, Stony Brook, NY 11794-3600
E-mail address: glimm@ams.sunysb.edu

Hydrodynamics Methods Group, Applied Theoretical and Computational Physics Division, Los Alamos National Laboratory, Los Alamos, NM 87545
E-mail address: jgrove@lanl.gov

Department of Applied Mathematics and Statistics, University at Stony Brook, Stony Brook, NY 11794-3600
E-mail address: linli@ams.sunysb.edu

Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, P.R.China