UCRL- JC-105251
PREPRINT

A.2.11

Conservative Front-Tracking for Inviscid Compressible Flow

John B. Bell
Phillip Colella
Michael L. Welcome

This paper was prepared for presentation at
AIAA 10th Computational Fluid Dynamics
Conference, Honolulu, Hawaii, June 24-26, 1991

April 4, 1991

Lawrence Livermore National Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

CONSERVATIVE FRONT-TRACKING FOR INVISCID COMPRESSIBLE FLOW*

John B. Bell†
Lawrence Livermore National Laboratory
Livermore, CA 94550

Phillip Colella§
University of California Berkeley
Berkeley, CA 94720

Michael L. Welcome‡
Lawrence Livermore National Laboratory
Livermore, CA 94550

Abstract

In this paper we describe a front-tracking algorithm for modeling the propagation of discontinuous waves in two space dimensions. The algorithm uses a volume-of-fluid representation of the front in which the local frontal geometry is reconstructed from the state information on either side of the discontinuity and the Rankine-Hugoniot relations. The algorithm is coupled to an unsplit second-order Godunov algorithm and is fully conservative, maintaining conservation at the front. The combination of a volume-of-fluid representation of the front and a fully conservative algorithm leads to a robust high resolution method that easily acomodates changes in the topology of the front as well as kinks arising when a tracked front interacts with a captured discontinuity. The Godunov/tracking integration scheme is coupled to a local adaptive mesh refinement algorithm that selectively refines regions of the computational grid to achieve a desired level of accuracy. An example showing the combination of tracking and local refinement is presented.

Introduction

A broad range of phenomena in unsteady supersonic flow are dominated by the propagation and interaction of discontinuous nonlinear waves. There are two basic approaches to treating discontinuous waves associated with hyperbolic conservation laws. One approach is high-resolution finite difference methods that resolve discontinuities on a grid by addition of a suitable dissipation while preserving high accuracy in smooth regions. The other approach is front tracking where the discontinuity is treated as a free boundary whose dynamics are described in terms of Rankine-Hugoniot relations and appropriate entropy considerations. The literature associated with both of these approaches is vast and a comprehensive study is beyond the scope of this paper. The interested reader is referred to Chern and Colella [1], Chern et al [2], and Woodward and Colella [3] for a survey of the literature associated with these areas.

In this paper we present a new algorithm, which generalizes earlier work on Chern and Colella [1], that combines a conservative front-tracking scheme with an unsplit second-order Godunov difference method used in conjunction with local adaptive mesh refinement. The basic strategy is to explicitly model the propagation of some distinguished discontinuity through a fixed finite difference mesh and rely on the difference technique to capture other waves. The tracked discontinuity is represented using a volume of fluid description that is insensitive to complexity of the front, changes in front topology, and is readily extendable to three space dimensions.

The key idea of Chern and Colella is the use of an algebraic redistribution technique to alleviate CFL time step restrictions arising when a finite difference cell is split into two pieces, one of which can be arbitrarily small. The use of the redistribution technique provides a method that is globally conservative without encurring a penalty in time step size. The principal

* This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. Partial support under contract No. W-7405-Eng-48 was provided by the Applied Mathematical Sciences Program of the Office of Energy Research and by the Defense Nuclear Agency under IACRO 90-824. Prof. Colella was supported by the Army Research Office under grant DAALO3-88-K-0197; by DARPA and the National Science Foundation under grant DMS-8919074; and by a National Science Foundation Presidential Young Investigator award under grant ACS-895822.

† Group Leader, Applied Mathematics Group

§ Associate Professor, Dept. of Mechanical Engineering

‡ Staff Scientist, Applied Mathematics Group

innovations discussed in this paper are the use of a finite difference approximation to an effective advection equation for moving the front and the use of the redistribution technique to couple the tracking method to a local mesh refinement scheme where the tracked front can cross different levels of refinement.

In the next section we provide an overview of the basic tracking integration method. The details of the tracking algorithm of discuss in section 3. In section 4, we describe the coupling of tracking to the local adaptive mesh refinement algorithm of Berger and Colella [4]. The last section of the paper presents a numerical example.

Overview of the Integration Algorithm

In this section we describe the basic second-order Godunov algorithm with front tracking in a summary form. Details of the various components are described in the next section. We will describe the method in a fairly general form; however, the emphasis in this paper will be on tracking a gas-dynamic shock. In particular, that components of the algorithm that are specific to the type of discontinuity being tracked will be discussed only in the context of a gas-dynamic shock. Furthermore, special simplifications of the algorithm for this case will be indicated. Thus, we want to solve

$$\frac{\partial U}{\partial t} + \frac{\partial F^x}{\partial x} + \frac{\partial F^y}{\partial y} = 0 \qquad (2.1)$$

where, in the case of gas dynamics,

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} \quad F^x(U) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uE + up \end{bmatrix} \quad F^y(U) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vE + vp \end{bmatrix}.$$

The representation uses a distinct state U on each side of the tracked front. We will refer to one of the states as "inside" the front and the other as "outside" the front which we will denote as U^1 and U^2, respectively. Thus, for each cell, Δ_{ij}, through which the front passes, we define both states, U_{ij}^1 and U_{ij}^2. For these "mixed" cell through which the front passes we also define Λ_{ij}^l, l=1,2 to denote the volume fraction of the cell inside and outside the front respectively. ($\Lambda_{ij}^1 + \Lambda_{ij}^2 = 1$.) Cells fully inside or outside the front require only a single state vector. For the special case of a gas-dynamic shock, the post-shock state will be correspond to l=1 and the pre-shock state to l=2.

The basic algorithm is, essentially, a three-step process. In the first two steps, which are actually independent, we compute fluxes in for U^1 and U^2 "ignoring" the presence of the tracked front and we advance the location of the front. To perform these first two steps, it is necessary to extrapolate states to the opposite side of the front; e.g., for cells within two zones of the front we must define U^1 in cells where $\Lambda^1 = 0$ and U^2 in cells where $\Lambda^2 = 0$ Using these extrapolated values, whose exact specification is given below, we define states $U_{ij}^{ext\,l}$, l=1,2 that extend the definitions of U^1 and U^2. The constructions used in the first two steps of the algorithm use these extended states.

In the first step of the algorithm, we use an unsplit second-order Godunov integration algorithm developed by Colella [5] to compute fluxes for the pre- and post-shock states. This scheme has the form

$$U_{ij}^{n+1} = U_{ij}^n + \frac{\Delta t}{\Delta x}(F_{i-1/2,j}^x - F_{i+1/2,j}^x) + \frac{\Delta t}{\Delta y}(F_{i,j-1/2}^y - F_{i,j+1/2}^y). \qquad (2.2)$$

Here, $F_{i+1/2,j}^x$, $F_{i,j+1/2}^y$ approximate time averaged fluxes at the cell edges, and are assumed to be explicit functions of U^n of the form

$$\begin{aligned} F_{i+1/2,j}^x &= F^x(U_{i,j-1}^n,\ldots,U_{i+1,j+1}^n;\ (D_x^- U)_{i-r,j-r}, \\ &\quad (D_y^- U)_{i-r,j-r},\ldots,(D_x^- U)_{i+r,j+r},(D_y^- U)_{i+r,j+r}) \\ F_{i,j+1/2}^y &= F^y(U_{i-1,j}^n,\ldots,U_{i+1,j+1}^n;\ (D_x^- U)_{i-r,j-r}, \\ &\quad (D_y^- U)_{i-r,j-r},\ldots,(D_x^- U)_{i+r,j+r},(D_y^- U)_{i+r,j+r}) \end{aligned}$$

where

$$(D_x^- U)_{i,j} = U_{i,j}^n - U_{i-1,j}^n, \qquad (D_y^- U)_{i,j-1} = U_{i,j}^n - U_{i,j-1}^n$$

In other words, $F_{i+1/2,j}^x, F_{i,j+1/2}^y$ on U^n depends on the 6 cells nearest the cell edge where the flux is defined, plus a possible dependence on values of U^n farther away which appear only as one sided differences in U^n. In addition, the scheme has the property that setting any of the $D_x^- U, D_y^- U$ to zero adds dissipation to the scheme and when they are all set to zero the resulting scheme is a first-order version of Godunov's method that has corner coupling so that it is stable for CFL's up to 1.0. Here, the fluxes for U^1 and U^2 are computed independently, ignoring the presence of the tracked front, using the extended values to fill the required stencil of the difference scheme. The only modification made to the integration module is to zero $D_x^- U$ and $D_y^- U$ when these differences involve cells that are fully on the other side of the front; i.e., difference contributions for computing U^l-fluxes that attempt to use U^l-values in cells where $\Lambda^L = 0$ are set to zero.

In the second step, we reconstruct the front shape from the volume fractions and solve an advection equation to advance the front. The reconstruction assumes that a normal to the front $\vec{n}_{ij}^f$ and a front speed s_{ij} can be computed from local information and that this information can be extended to a neighborhood around the mixed cells containing zones that can participate in the front movement, e.g., cells that are within one zone of a mixed cell. This phase of the algorithm depends on the specific type of discontinuity that is being tracked; its form for a shock is discussed in the next section. Using the normal and the volume fraction we can determine a

linear reconstruction of the front inside each mixed cell. An effective equation for the volume fraction is then used to advance the front by computing volume fractions at the new time level. In particular, we integrate

$$\Lambda_t + \nabla\cdot(\vec{s}\Lambda) = -\Lambda\,\nabla\cdot\vec{s} \tag{2.3}$$

where $\vec{s} = s\vec{n}$ using a piecewise linear reconstruction of the front inside each cell to compute the volume fraction flux.

The final step of the algorithm uses the fluxes computed in the second step and the frontal dynamics from the first step to update the cells near the tracked front. This procedure is based on ideas developed by Chern and Colella. The fluxes are combined with apertures that specify the area (in space-time along each edge) where that flux is applicable to compute a conservative update for both U^1 and U^2. for each cell through with the front passes during a time step. Since cells intersected by the front at the new time have a reduced volume, fully updating the cell would lead to a violation of CFL conditions and would potentially lead to an instability. Thus, each cell receives a fraction of its specified conservative update that is consistent with stability. To preserve conservation the remainder of the update is redistributed to neighboring cells in a volume-weighted manner that is consistent with the characteristic structure of the equations.

Integration Scheme with Tracking

In this section we describe the integration scheme with tracking in more detail. We denote the grid cells by Δ_{ij} where $i=i_{lo},i_{hi}$ and $j=j_{lo},j_{hi}$. Before beginning either the front advancement or the flux computation we must first define the extended states. The extension is done in a two step process. First, we identify a grid cells as potentially participating in the front motion if it is either a mixed cell, $0<\Lambda^l_{ij}<1$, or it is adjacent to a mixed cell. These are the cells through which a portion of the front may pass during the time step. For later reference, we define a marker, P_{ij} such that $P_{ij}=1$ for cells that can participate in front motion and 0 otherwise. In the first stage of the extension, we define an extended state

$$\overline{U}^l_{ij} = \left[\frac{\sum_{nbh(i,j)} \Lambda^{n,l} U^{n,l}}{\sum_{nbh(i,j)} \Lambda^{n,l}}\right] \tag{3.1}$$

for each cell where $\Lambda^l=0$, and $P_{ij}=1$, where $nbh(i,j)$ represents the 8 neighboring cells. We are guaranteed that at least one cell in $nbh(i,j)$ has nonzero Λ^l so that (3.1) is defined. We must now extend an additional cell beyond the cells defined by (3.1). For this purpose, for each cell that does not participate in front motion but borders a participating cell and satisfies $\Lambda^l_{ij}=0$ we define $\tilde{U}^l_{ij}$ to be the average of the $\overline{U}^l$ values defined by (3.1) lying in $nbh(i,j)$. (A volume weighted averaged does not make sense here because the barred states are associated with these cells have $\Lambda^l=0$.) These additional states are required to complete the stencil of the difference scheme for each cell that participates in front motion. We then define the extended states $U^{ext,n,l}_{ij}$ to be $U^{n,l}_{ij}$ if $\Lambda^l_{ij}>0$, $\overline{U}^l_{ij}$ for cells that participate in front motion but for which $\Lambda^l_{ij}=0$, and $\tilde{U}^l_{ij}$ for neighboring cells that are not otherwise defined.

In the first step of the algorithm, we use the extend states defined above to compute fluxes for the edges of cells using the second-order Godunov procedure described in the previous section. The use of extended states and the modification to the integration algorithm allows fluxes $F^{x,l}_{i+1/2,j}$ and $F^{y,l}_{i,j+1/2}$ to be defined for all edges of cell where $\Lambda^l_{ij}>0$ or where $\Lambda^l_{ij}=0$ and $P_{ij}=1$.

In the next step of the algorithm we reconstruct and move the front. To accomplish this step we need to define a normal to the front, $\vec{n}$ and a front speed s, for each cell that participates in front motion. In the case of a shock wave, the Rankine-Hugoniot relations determine the normal In particular,

$$\vec{n}_{ij} = \begin{bmatrix} u^{ext,1}_{ij} - u^{ext,2}_{ij} \\ v^{ext,1}_{ij} - v^{ext,2}_{ij} \end{bmatrix} \times \frac{1}{\sqrt{(u^{ext,1}_{ij} - u^{ext,2}_{ij})^2 + (v^{ext,1}_{ij} - v^{ext,2}_{ij})^2}} \tag{3.2}$$

which reflects the fact that the change in velocity across a shock occurs in a direction normal to the shock. Once the normal has been computed the shock speed s_{ij} is determined by projecting the pre- and post-shock states ($l=2$ and $l=1$, respectively) onto the normal direction and solving the associated Riemann problem to determine the resulting shock speed.

These normals and shock speeds can now be used to solve (2.3) to advance the front. We have used an operator split approach in which we solve

$$\Lambda^1_t + (s_x\Lambda^1)_x = -\Lambda^1 s_{x,x} \tag{3.3a}$$

and

$$\Lambda^1_t + (s_y\Lambda^1)_y = -\Lambda^1 s_{y,y} \tag{3.3b}$$

alternating which is performed first between successive time steps. Here, s_x and s_y are the x and y components of $s\vec{n}$. (We are using Λ^1 to advance the front; we will derive Λ^2 from the constraint that the Λ's sum to 1. To discretize the x-sweep, we first define edge-centered speeds using

$$s_{i+1/2,j} = \frac{P_{ij}\, s^x_{i,j} + P_{i+1,j}\, s^x_{i+1,j}}{P_{ij} + P_{j+1,j}}$$

for each edge bordering a cell where $P_{ij}=1$. These edge velocities are then used to compute volume-fraction fluxes across each cell edge. In computing these fluxes we need to utilize the local structure of the front. Using Λ^n_{ij} and $\vec{n}_{ij}$ we can uniquely construct the

front in cell Δ_{ij}. In particular, for each cell we can find a unique value of a such that

$$\overline{S}_{ij} \equiv \{\vec{x} \varepsilon\ \Delta_{ij}\ s.t. \vec{n}\cdot\vec{x} - a < 0\}$$

satisfies

$$\Lambda_{ij}^{n} = \frac{\text{Area}(\overline{S}_{ij})}{\Delta x\,\Delta y}$$

The volumetric flux of Λ though edge $i+\frac{1}{2}j$ is the area of the intersection of $\overline{S}$ in the upwind cell with the portion of that cell sweep out by the edge velocity. More precisely, if $s_{i+\frac{1}{2},j} \geq 0$ then

$$F_{i+\frac{1}{2},j}^{\Lambda} = \text{Area}(S_{ij} \cap [\Delta x - s_{i+\frac{1}{2},j}\Delta t, \Delta x] \times [0, \Delta y])$$

where the rectangle is defined with respect to a local coordinate system on Δ_{ij} where the lower-left corner of the cell is the origin. Similarly, if $s_{i+\frac{1}{2},j} < 0$ then

$$F_{i+\frac{1}{2},j}^{\Lambda} = -\text{Area}(\overline{S}_{i+1,j} \cap [0, s_{i+\frac{1}{2},j}\Delta t] \times [0, \Delta y])$$

where the rectangle is defined with respect to analogous cell $i+1,j$ coordinates. With this volumetric flux definition,

$$\Lambda_{ij}^{n+\frac{1}{2},1} = \frac{\left[\Lambda_{ij}^{n,1} - \dfrac{F_{i+\frac{1}{2},j}^{\Lambda} - F_{i-\frac{1}{2},j}^{\Lambda}}{\Delta x\ \Delta y}\right]}{\left[1 - \Delta t \dfrac{s_{i+\frac{1}{2},j} - s_{i-\frac{1}{2},j}}{\Delta x}\right]} \tag{3.4}$$

The y-sweep is defined analogously and defines Λ^{n+1} from $\Lambda^{n+\frac{1}{2}}$.

The final step in the integration algorithm merges the results of the first two steps. Although we will not make any distinction at this point in the discussion, for cells that are sufficiently far from the tracked front the algorithm discussed below collapses to the update formula (2.2) using appropriate fluxes. To incorporate the effects of the front into the update for U^l we must construct an approximation to the motion of the front during the time step. For cells that are mixed ($0<\Lambda<1$), given $\Lambda_{ij}^{n,1}$, $\Lambda_{ij}^{n+1,1}$, and the front speed s_{ij}, we can find a plane S_0 in $(\vec{x},t)$ space of the form $S_0 = \{(\vec{x},t)\colon \vec{x}\cdot\hat{n} - \overline{s}\,t + a = 0\}$ that locally represents the trajectory of the tracked front through the finite difference cell Δ_{ij}. If we define

$$S_1 = \{(\vec{x},t)\colon \vec{x}\cdot\hat{n} - \overline{s}\,t + a < 0\},$$

then $\overline{s}$ and a can be chosen so that

$$\Lambda_{ij}^{n} = \text{area of } S_1 \cap \Delta_{ij} \times \{t^n\}$$

$$\Lambda_{ij}^{n+1} = \text{area of } S_1 \cap \Delta_{ij} \times \{t^{n+1}\}\ .$$

In the case that $0<\Lambda_{ij}^{n,1}, \Lambda_{ij}^{n+1,1}<1$, S_0 will be uniquely determined from the Λ's and $\vec{n}$ with using the front speed s_{ij}. In the case when one of the Λ's is zero, then the speed s is used to define $\overline{s}$ in determining S_0.

We will use S_0 are S_1 to determine the areas of the surfaces in space-time on which fluxes of each of the components are imposed. We define apertures $A_{ij}^{i\pm\frac{1}{2},j,1}$ to be the area of $(S_1 \cap L_{i\pm\frac{1}{2},j} \times [t^n, t^n+\Delta t])$, and apertures $A_{ij}^{i,j\pm\frac{1}{2},1}$ to be the area of $(S_1 \cap L_{i,j\pm\frac{1}{2}} \times [t^n, t^n+\Delta t])$, where the $L_{i\pm\frac{1}{2},j}, L_{i,j\pm\frac{1}{2}}$ are the sides of Δ_{ij}. We also define

$$A_{ij}^{f} = \text{area of } S_0 \cap \Delta_{ij} \times [t^n, t^n+\Delta t].$$

The calculation of $A_{ij}^{i\pm\frac{1}{2},j,1}$, $A_{ij}^{i,j\pm\frac{1}{2},1}$, A^f is a straightforward exercise in trigonometry. Theses apertures represent the portion of the edge on which fluxes for U^1 are applied. The U^2 apertures are defined using by the compatibility condition

$$A^{i,j\pm\frac{1}{2},1} + A^{i,j\pm\frac{1}{2},2} = \Delta x\,\Delta t$$

and

$$A^{i\pm\frac{1}{2},j,1} + A^{i\pm\frac{1}{2},j,2} = \Delta y\,\Delta t$$

We mention that A^f is easily calculated from the $A^{i\pm\frac{1}{2},j}$'s, $A^{i,j\pm\frac{1}{2}}$'s and Λ's using the divergence theorem. Away from the front, the appropriate apertures are defined to be $\Delta t \times$ the length of the cell edge and $A^f = 0$.

We now apply the divergence theorem to each cell to obtain an initial approximation $U_{ij}^{*,l}$ to U^l at t^{n+1}

$$\Lambda_{ij}^{n+1} U_{ij}^{*,l} = \Lambda_{ij}^{n} U_{ij}^{n,l} - \frac{1}{\Delta x\,\Delta y}(A_{i+\frac{1}{2},j}^{l} F_{i+\frac{1}{2},j}^{x,l} - A_{i-\frac{1}{2},j}^{l} F_{i-\frac{1}{2},j}^{x,l} + A_{i,j+\frac{1}{2}}^{l} F_{i,j+\frac{1}{2}}^{y,l} - A_{i,j-\frac{1}{2}}^{l} F_{i,j-\frac{1}{2}}^{y,l} + (-1)^l A_{ij}^{f} F_{ij}^{f}) \tag{3.5}$$

Here $A_{i+\frac{1}{2},j}^{l}$, $A_{i,j+\frac{1}{2}}^{l}$ are given by

$$A_{i+\frac{1}{2},j}^{l} = \frac{1}{2}(A_{ij}^{i+\frac{1}{2},j,l} + A_{i+1,j}^{i+\frac{1}{2},j,l})$$

and

$$A_{i,j+\frac{1}{2}}^{l} = \frac{1}{2}(A_{i,j+1}^{i,j+\frac{1}{2},l} + A_{ij}^{i,j+\frac{1}{2},l})$$

In the general case, F^f is the flux across the tracked front in the cell, given by

$$F_{ij}^{f} = F^{\vec{n}}(U_{ij}^{\vec{n}}) - s_{ij} U_{ij}^{\vec{n}},$$

where $U^{\vec{n}}$ is the value of the solution to the Riemann problem for the system (2.1) projected in the $\vec{n}_{ij}$ direction along the ray $\xi/\tau = s_{ij}$, with left and right states $U_{ij}^{ext,2}, U_{ij}^{ext,1}$. In the case of a shock wave a simpler approximation can be used, namely,

$$F_{ij}^{f} = F^{\vec{n}}(U_{ij}^{ext,n,2}) - s_{ij} U_{ij}^{ext,n,2}, \tag{3.6}$$

We want to now reexpress (3.5) the update of U in terms of the change in U caused by the tracked front. To accomplish this we first define update that ignore the front

$$U_{ij}^{ext,n+1,l} = U_{ij}^{ext,n,l} + \frac{\Delta t}{\Delta x}(F_{i-\frac{1}{2},j}^{x,l} - F_{i+\frac{1}{2},j}^{x,l}) + \frac{\Delta t}{\Delta y}(F_{i,j-\frac{1}{2}}^{y,l} - F_{i,j+\frac{1}{2}}^{y,l}). \tag{3.7a}$$

where the fluxes are computed from the extended states $U^{ext,n}$. Then we rewrite (3.5) as

$$\Lambda_{ij}^{n+1}U_{ij}^{*,l} = \Lambda_{ij}^{n+1}U_{ij}^{ext,n+1,l} + \delta M_{ij}^{l} \qquad (3.7b)$$

where

$$\delta M_{ij}^{l} = \Lambda_{ij}^{n}U_{ij}^{n} - \Lambda_{ij}^{n+1}U_{ij}^{ext,n+1,l} - \frac{1}{\Delta x\,\Delta y}(A_{i+1/2,j}^{l}F_{i+1/2,j}^{x,l} - A_{i-1/2,j}^{l}F_{i-1/2,j}^{x,l} + A_{i,j+1/2}^{l}F_{i,j+1/2}^{y,l} - A_{i,j-1/2}^{l}F_{i,j-1/2}^{y,l} + (-1)^{l}A_{ij}^{l}F_{ij}^{l} \qquad (3.7c)$$

We note that with this definition of δM, sufficiently far away from the tracked front $\delta M = 0$ to that (3.7) is simply a re-expression of (2.2). If we divide (3.7b) by $\Lambda_{ij}^{n+1,l}$ we would have a conservative update for U^{l}, namely,

$$U_{ij}^{n+1,l} = U_{ij}^{ext,n+1,l} + \frac{\delta M_{ij}{}^{l}}{\Lambda_{ij}^{n+1,l}} \quad .$$

However, because the Λ's can be arbitrarily small, this update can require an excessive time step restriction to remain stable. (This update is, of course, stable for cells where Λ=1.) To avoid this type of time-step limitation we will use the redistribution ideas of Chern and Colella [1]. Their basic idea is to define a preliminary update that adds a fraction of the update that will be stable at the time step determined CFL criteria on the regular grid. This gives a preliminary evolution in time, given by

$$U_{i,j}^{*,l} = U_{i,j}^{n,l} + \delta M_{i,j}^{l} \qquad (3.8)$$

This update does not preserve discrete conservation form. In order to have conservation, we must distribute $(1-\Lambda_{ij}^{n+1,l})\delta M_{ij}^{l}$ onto the grid. In the general case we do this by decomposing these increments into characteristic variables and distributing them to nearby cells in a volume-weighted fashion. If we expand

$$(1-\Lambda_{ij}^{n+1,l})\delta M_{ij}^{l} = \sum_{k}\alpha_{k}^{l}r_{k}^{l}$$

where $r_k(U)$, $k = 1,\ldots,N$ are the linearized right eigenvectors of the system (2.1) projected in the $\vec{n}_{ij}$ direction, with $r_k^l = r_k(\overline{U}_{ij}^l)$, then we can define

$$\begin{aligned}
\delta M^{+,l} &= \sum_{k<p}\alpha_k^l r_k^l \\
\delta M^{-,l} &= \sum_{k>p}\alpha_k^l r_k^l \\
\delta M^{0,l} &= \alpha_p^l r_p^l \\
\delta M^{red,1} &= \delta M^{+,1} + \delta M^{+,2} + \delta M^{0,1} \\
\delta M^{red,2} &= \delta M^{-,1} + \delta M^{-,2} + \delta M^{0,2} \\
\Lambda_{ij}^{red,l} &= \sum_{nbh(i,j)} \Lambda^{n+1,l}
\end{aligned} \qquad (3.9)$$

Then we define the final values of $U_{ij}^{n+1,l}$ to be

$$U_{ij}^{n+1,l} = \overline{U}_{ij}^{n+1,l} + \sum_{nbh(i,j)} \frac{\delta M^{red,l}}{\Lambda_{i,j}^{red,l}}. \qquad (3.10)$$

Again, in the special case of a gas dynamic shock, the redistribution step can be simplified by setting

$$\delta M^{red,1} = (1-\Lambda^{n+1,1})\delta M^{1} + \Lambda^{n+1,1}\delta M^{2}$$

and

$$\delta M^{red,2} = 0.$$

Finally, we note that in the shock wave case, several minor modifications were made to the methodology to deal with degenerate situations. First, when the jump in velocity in (3.2) across the front is small compared to the jump in pressure, the computation of the normal from the velocity jump is not reliable. This can occur, for example, when pieces of the front collide. When the jump is judged to be small we convert the pre-shock fluid in the zone to post-shock and adjust the volume fractions to reflect this change. These modifications are accumulated into the δM's in the redistribution step to that the method remains conservative. The other type of degeneracy that is encountered occurs when there are extremely sharp corners in the front. In this case it is possible for isolated fragments of pre-shock material to be entrained into the post-shock region. When this fragmentation occurs the fragments are absorbed into the post-shock states and the volume fractions are adjusted.

Coupling to AMR

In this section we describe how the basic conservative tracking methodology is coupled to a local adaptive mesh refinement algorithm based on the approach of Berger and Colella [2]. We will briefly review the basic adaptive refinement algorithm before describing the modifications needed to incorporate tracking. The algorithm is based on a hierarchical grid structure composed of grids of varying resolution. The grid hierarchy is constructed using an error estimation procedure to identify cells having unacceptable errors that are then clustered into logically-rectangular grids that are subdivided to form finer cells where more resolution is required. Integration of the differential equations on this hierarchical grid structure is a three-step procedure. First, the coarse grid is integrated to supply boundary data for finer grids. The fine grids are then integrated, subcycling in time, to catch up to the coarse grid. Finally, the coarse grids are corrected to reflect the improved resolution of the finer grids.

The bulk of the tracking methodology appears in the integration module of the code which is cleanly separated from the adaptive mesh shell. There are two modifications that are required to combine tracking with adaptive refinement. First, interpolation from coarse grids to finer grids is no longer a simple interpolation procedure; instead the front must be constructed on the coarse grid and the relation of the fine grid cells to the front must be used in the interpolation. This occurs in interpolating boundary conditions for the fine grids and

in initializing fine grids from coarse grids when the error estimation criteria call for finer grids in a particular region. Similarly, in averaging fine grid cells to define values on an underlying coarse grid cell, the averaging must be weighted by the volume fractions in the fine cells to ensure conservation.

The second modification represents a more substantial change. Without tracking the correction to the coarse grid to reflect the improved resolution of the fine grid is to replace underlying coarse grid values with the average of the covering fine grid values and to add a flux correction δF to coarse grid cells that border fine grids. More precisely, we set

$$U^c := U^c + \frac{\delta F}{A^c} \tag{4.1}$$

where

$$\delta F = \sum (L^f \Delta t^f F^f) - L^c \Delta t^c F^c$$

with the sum taken over the fine grid edges that cover the coarse edge and over the number of time steps the fine grid is subcycled for a coarse step. Here, A^c is the area of the coarse cell, $L^{c,f}$ are the lengths of the coarse cell edge and the fine cell edges that cover it. Note that δF is an extrinsic quantity, e.g. mass not density. The update (4.1) is equivalent to repeating the integration of the coarse cell using the sum of the fine grid fluxes to update the cell instead of the coarse grid flux.

When tracking is included, additional modification are needed to account for the effects of the redistribution step of the algorithm. These corrections arise because redistribution provides an additional mechanism for communication across a coarse-fine boundary. There are four basic coarse-fine redistribution terms:

δR_f^I: These are the values redistributed into the fine grid from the grid boundary cells; hence, there are artificial and their effect must be removed.

δR_c^I: These are the values redistributed from the coarse grid into the coarse grid cells underlying the fine grid that are subsequently lost when the coarse values is redefined by averaging the fine values.

δR_f^E: These are the values redistributed from the fine grid into its boundary cell. These values are then lost.

δR_c^E: These denote the redistribution values from the coarse grid underlying the fine grid to the coarse grid cells on the boundary of the fine grid. Their effect should be removed.

We now define

$$\delta R^I = \delta R_c^I - \sum \delta R^{f,I} \tag{4.2}$$

and

$$\delta R^E = \sum \delta R^{f,E} - \delta R_c^E \tag{4.3}$$

These terms, which are accumulated in extrinsic form, represent the values that should be added to the coarse interior cells on the boundary of the fine grid (and the fine cells that cover them) and the correction to be added to the coarse grid exterior to the find grid. We associate δR^I with the coarse grid cell from which the values came and δR^E with the coarse grid cell that received them. We note that the δR's and the δF's must, in general be accumulated by both states and that the fluxes appearing in the definition of δF are aperture weighted. These terms are then combined to form

$$\delta M_{c-f}^I = \frac{1}{A^c}(\delta F^I + \delta R^{E,I} + \delta R^{I,I}) \tag{4.4}$$

which is a generalized reflux correction to coarse grid cells that border fine grids. As we did in the main integration step, we include a stable portion of the update in each cell and redistribute the remainder to its neighbors. Thus,

$$U^{n+1,I} := U^{n+1,I} + \delta M_{c-f}^I$$

and

$$\delta M_{c-f}^{red,I} = (1-\Lambda^I)\delta M_{c-f}^L \tag{4.5}$$

The $\delta M_{c-f}^{red,I}$ are then redistributed to the neighboring coarse cells using the procedure defined in the previous section. Values redistributed during this procedure to coarse grids that are covered by fine grid cells are lifted to the fine grid, weighted by the fine grid Λ's. We note that for interfaces between coarse and fine grids that are not near a portion of the tracked front, the δR's in (4.4) vanish and the reflux correction reduces to (4.1). For a shock wave the redistribution can be simplified. First, $\delta R^{E,2}$ and $\delta R^{I,2}$ are both zero in (4.4). Furthermore, the pre-shock redistribution in (4.5) can call be placed in the post-shock cells as in the basic integration algorithm.

Numerical Example

In this section we present a numerical example showing the combined tracking / adaptive mesh refinement algorithm. The example shows a tracked Mach 10 shock impinging on a circular cloud of gas. The cloud region is in thermodynamic equilibrium but it is 10 times as dense as the surrounding ambient gas. The shock is initially to the left of the cloud and moving to the right with the cloud initially at rest. The base grid is 60×30 with two levels of refinement allowed. Each refinement reduces the grid spacing by a factor of 4 in each direction, thus, a single grid at the finest level of resolution would be 960×480. The refinement criteria set to prevent the incident shock from being refined in the upper portion of the domain. Figures 1a through 1d show contours of the logarithm of density. In the contouring algorithm the pre- and post-shock states are averaged onto the grid for contouring. Thus, we observe an apparent thickening of the

tracked front as its level of refinement progresses from the finest level near the cloud to the coarsest level near the top of the domain. Figure 1a shows early time when the shock has just begun to enter the cloud. In the next frame, the shock has progressed most of the way through the cloud and the cloud is beginning to deform dramatically. Note that the bow shock is well-resolve using the capturing scheme. In 1c the tracked shock has left the cloud and the cloud has continued to deform. In the final frame, the tracked shock has transitioned to Mach reflection. The cloud is continuing to undergo increasingly complex distortions.

Figures 2a through 2d show a composite of the tracked front superimposed on boxes that represent where the refined grid patches are located. (Not at the same times as the frames in Figure 1.) The larger boxes show where the level 1 grids are located and the smaller boxes show the level 2 grids. During this sequence we note several changes in the structure of the front. Between frames 2a and 2b, the front pinches off to form two separate pieces. Between frames 2b and 2c the smaller piece of the front disappears. Finally, between 2c and 2d we see a sharp kink develop in the tracked front indicating the transition of Mach reflections. Because the algorithm reconstructs the front locally at each step from the local state variables and a volume fraction no special cases are needed to determine how fronts interact or change topology.

References

1. I.-L. Chern and P. Colella, "A Conservative Front Tracking Method for Hyperbolic Conservations Laws," UCRL-97200, LLNL, July 1987.
2. I-L. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv, "Front Tracking for Gas Dynamics," *J. Comp. Phys*, vol. 62, pp. 83-110, 1986.
3. P.R. Woodward and P. Colella, "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks," *J. of Comput. Phys.*, vol. 54, pp. 115-173, 1984.
4. M. J. Berger and P. Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *J. Comp. Phys.*, vol. 82, pp. 64-84, 1989.
5. P. Colella, "A Multidimensional Second Order Godunov Scheme for Conservation Laws," *J. Comp. Phys.*, vol. 87, pp. 171-200, 1990.

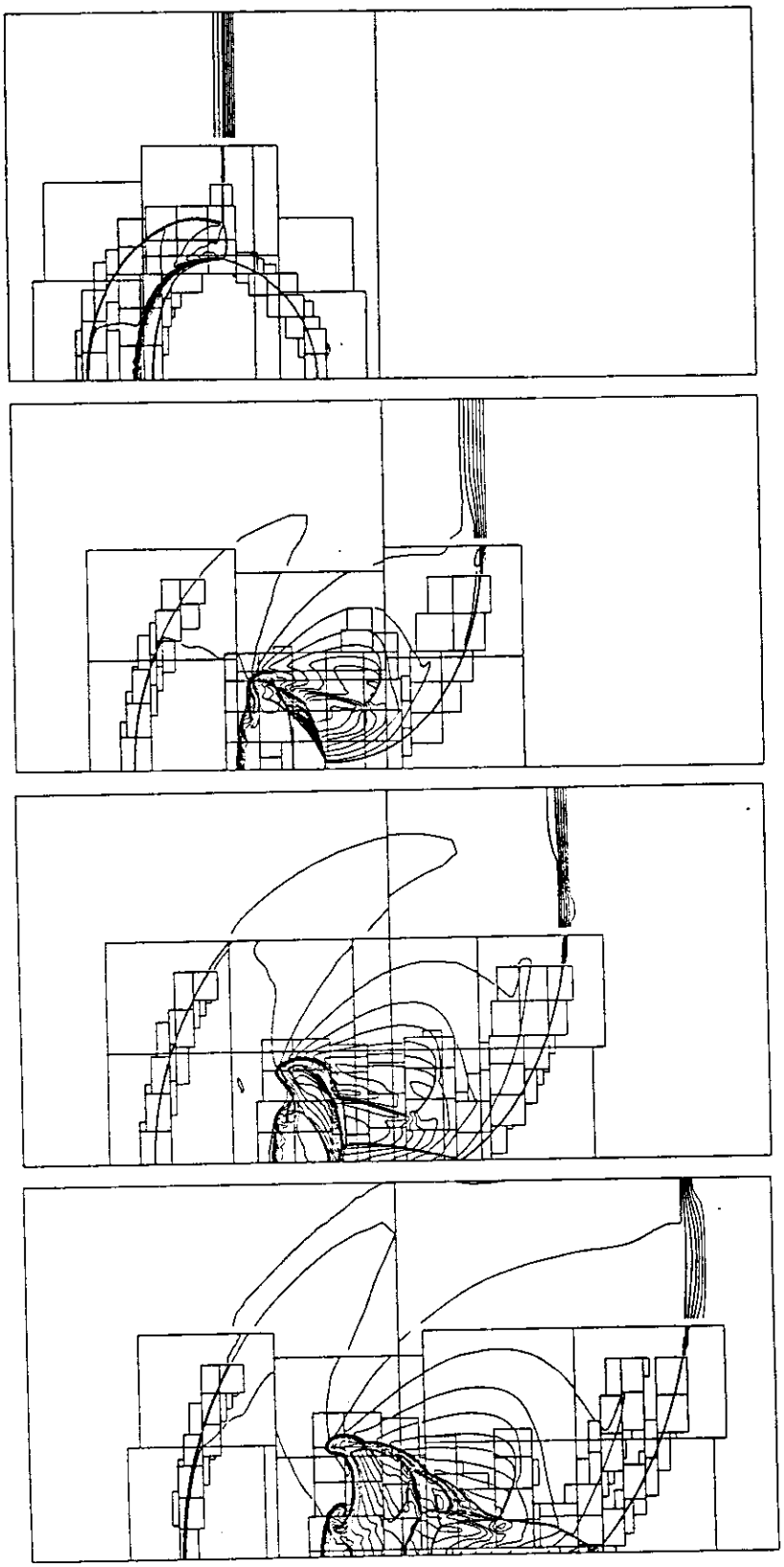

Figure 1. Times sequence of logarithmically spaced density contours from computation of a Mach 10 shock hitting a dense cloud.

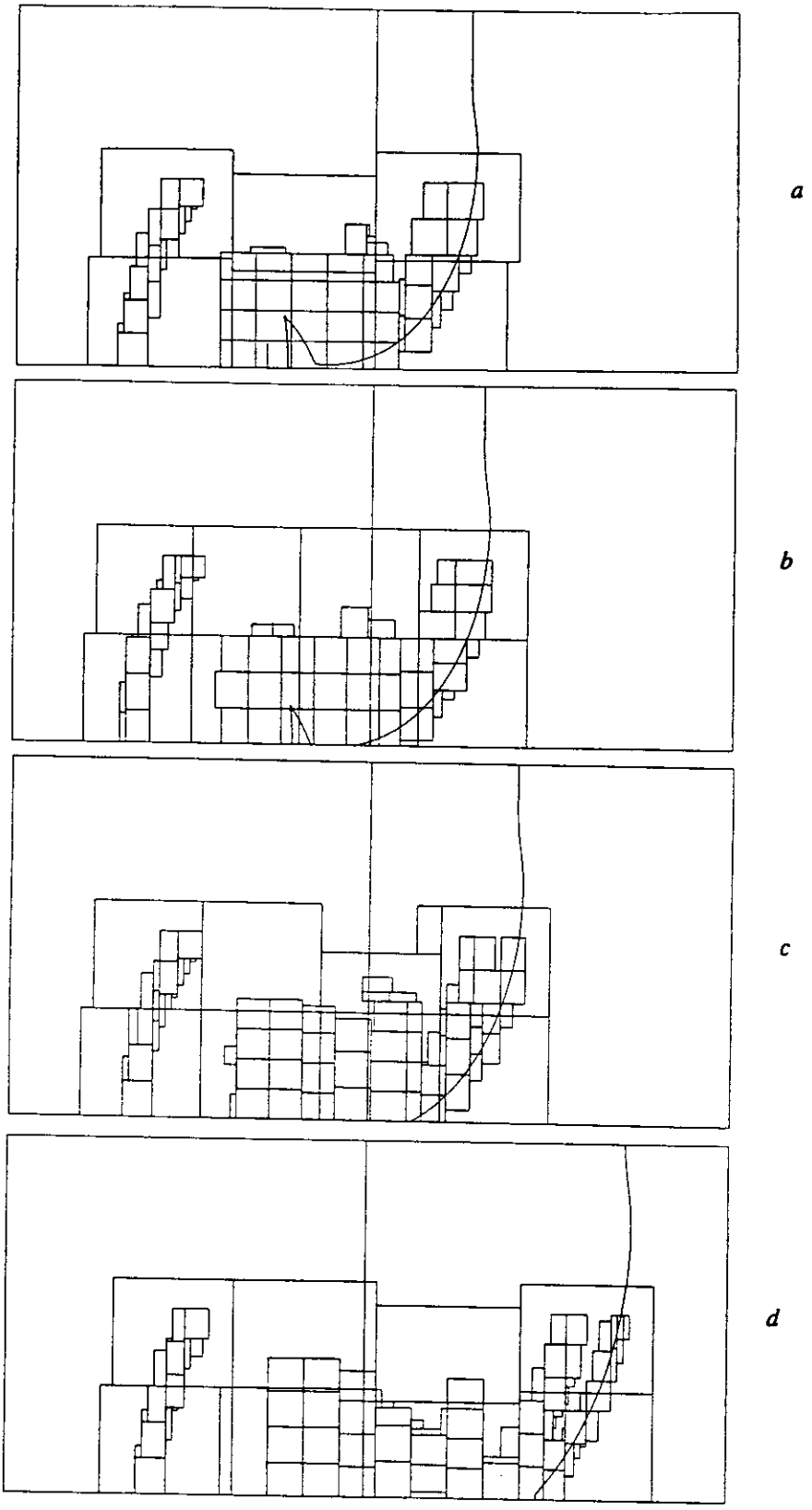

Figure 2. Times sequence of tracked front location superimposed with location of fine grids.